NSP Command Reference Manual

Abstract

The NSP Command Reference Manual defines the command and response syntax for standard cryptographic functions in the A10160-V, A9160-V and A8160-V Network Security Processors that support the Atalla Variant key management method. The command syntax for Network Security Processors that support the Atalla Key Block (AKB) key management method is documented in the

.

Network Security Processors are designed for use in Automated Teller Machine (ATM), Electronic Funds Transfer (EFT), and Point Of Sale (POS) networks. They can also be used for other types of applications that require Data Encryption Standard (DES) or triple DES (3DES) support.

Commands that support Public key cryptography are not supported in this product. Customer specific commands are not documented in this manual.

Network Security Processor version 2.0 requires the use of a Secure Configuration Assistant-3 and version 2.0 Security Administrator Smart cards.

Product Version

Ax160 NSP Variant Version 2.00

Part Number	Published
C8Z37-9000A	May 2013

Document History

Part Number	Product Version	Published
587397-002	version 1.00	November 2009
AJ556-9004A	version 1.10	April 2010
AJ556-9004B	version 1.11	June 2010
AJ556-9004C	version 1.13	October 2010
AJ556-9004D	version 1.17	May 2011
AJ556-9004E	version 1.30	July 2011

NSP Command Reference Manual

Glossary | Index | Figures | Tables

What's New in This Manual xix
New and changed information xix
About This Manual xxiii
Who Should Read This Manual xxiii
Your comments invited xxiv
Related documents xxiv
Type conventions xxiv

1. Introduction

Cryptographic functions 1-1
Operating overview 1-1
Command and response 1-1
Error responses 1-3
Detailed errors 1-4
Data formats 1-4
Programming guidelines 1-5
Sample program 1-6

2. Using DES keys

Master File Key 2-1
Key Exchange Key 2-1
Working keys 2-1
Key variants 2-2
Supported key types. 2-3
Key generation and translation 2-4
Non-volatile key table 2-4
Volatile table 2-5
Procedure to replace the current MFK with the pending MFK 2-5
Security precautions 2-6

3. DES key management

Quick reference 3-1

Generate Working Key, Any Type (Command 10) 3-4
Translate Working Key For Distribution (Command 11) 3-7
Translate Working Key For Local Storage (Command 13) 3-10
Load ATM Master Key – Diebold (Command 14) 3-13
Load ATM Master Key – IBM 3624 (Command 14) 3-16
Load ATM Master Key – IBM 4731 (Command 14) 3-19
Change ATM Communications Key – Diebold (Command 15) 3-23
Change ATM Communications Key – Docutel (Command 15) 3-26
Change ATM Communications Key – IBM 3624 (Command 15) 3-29
Change ATM Communications Key – IBM 4731 (Command 15) 3-32
Encrypt Financial Institution Table – Diebold (Command 16) 3-35
Encrypt Financial Institution Table – Docutel (Command 16) 3-38
Encrypt Financial Institution Table – IBM 3624 (Command 16) 3-41
Generate VISA Working Key (Command 18) 3-44
Translate Communications Key for Local Storage (Command 19) 3-46
Translate Working Key for Distribution to Non-Atalla Node (Command 1A) 3-49
Translate Communications Key for Local Storage Using a Specific Variant (Command 1D) 3-52
Generate New Initial Key for PIN Pad Using VISA DUKPT (Command 1E) 3-55
Generate Check Digits (Command 7E) 3-59
Translate Working Key for Local Storage Under the Current MFK to the Pending MFK (Command 9E) 3-63
Replace the Current MFK with the Pending MFK (Command 9F) 3-66
Translate an encrypted key between ECB and CBC modes (command 113) 3-69
Generate ATM MAC or Data Encryption Key (Command 11D) 3-72

4. Processing Personal Identification Numbers

About PIN Processing 4-1
PIN Processing Tasks 4-1
Encrypting PINs 4-2
Translating PIN Blocks 4-2
Verifying Incoming PIN Blocks 4-3
PIN Sanity Error 4-3
PIN Block Types 4-4
ANSI PIN Block 4-5
IBM 3624 PIN Block 4-7
PIN/Pad PIN Block 4-9
Docutel PIN Block 4-10
IBM Encrypting PIN Pad PIN Block 4-11
Burroughs PIN Block 4-12

ISO-3 PIN Block 4-13
IBM 4731 PIN Block 4-15
VISA Derived Unique Key Per Transaction PIN Block 4-17
PIN Processing Commands 4-21
Quick Reference 4-21
Encrypt PIN - ANSI Format 0 (Command 30) 4-23
Translate PIN (Command 31) 4-26
Translate PIN – VISA DUKPT (Command 31) 4-30
Verify PIN – Identikey (Command 32) 4-35
Verify PIN – IBM 3624 (Command 32) 4-41
Verify PIN – VISA (Command 32) 4-46
Verify PIN – Atalla DES BiLevel (Command 32) 4-51
Verify PIN – Diebold (Command 32) 4-56
Verify PIN – NCR (Command 32) 4-61
Verify PIN – Clear-PIN Comparison (Command 32) 4-67
Verify PIN – PIN-Block Comparison (Command 32) 4-70
Verify PIN – Burroughs (Command 32) 4-73
Verify PIN – Atalla 2x2 (Command 32) 4-78
Translate PIN – ANSI to PLUS and PLUS to ANSI (Command 33) 4-82
Translate PIN – ANSI to PIN/Pad (Command 33) 4-85
Translate PIN – ANSI to IBM 4731 (Command 33) 4-88
Translate PIN – IBM 3624 to IBM 3624 (Command 33) 4-92
Translate PIN – IBM 3624 to PIN/Pad (Command 33) 4-96
Translate PIN – PIN/Pad or Docutel to ANSI (Command 33) 4-100
Translate PIN – PIN/Pad or Docutel to PIN/Pad (Command 33) 4-103
Translate PIN – PIN/Pad or Docutel to IBM 4731 (Command 33) 4-106
Translate PIN – IBM 4731 to ANSI (Command 33) 4-110
Translate PIN – IBM 4731 to PIN/Pad (Command 33) 4-114
Translate PIN – IBM 4731 to IBM 4731 (Command 33) 4-118
Translate PIN – Double-Encrypted Input or Output (Command 35) 4-122
Verify Double-Encrypted PIN (Command 36) 4-126
PIN Change – Identikey (Command 37) 4-129
PIN Change – IBM 3624 (Command 37) 4-135
PIN Change – VISA (Command 37) 4-141
PIN Change – Atalla DES Bilevel (Command 37) 4-146
PIN Change – Diebold (Command 37) 4-152
PIN Change – NCR (Command 37) 4-157
Translate PIN And Generate MAC (Command 39) 4-163
Generate PVN and IBM Offset (Command 3D) 4-168

Decrypt PIN (Command 90) 4-172
PIN Translate (ANSI to PIN/Pad) and MAC Verification (Command BA) 4-175
Translate PIN (ANSI to PLUS) and Verify MAC (Command BB) 4-179
Translate PIN and Generate MAC (Command BD) 4-183
Verify Clear PIN (Command D0) 4-191
Generate Atalla 2x2 PVN (Command 11E) 4-194
Calculate PIN Offset (Command 30A) 4-197
Verify ePIN (Command 32C) 4-202
PIN and PIN-Block Translate (Command 335) 4-205
Generate ePIN Offset (Command 37B) 4-212

5. Processing Transaction Data

Data Processing Tasks 5-1
Encrypting and Decrypting Data 5-1
Supported Encryption/Decryption Methods 5-1
Using Initialization Vectors 5-2
Data Processing Commands 5-3
Quick Reference 5-3
Encrypt Or Decrypt Data Or Translate (Command 55) 5-5
Generate Random Number (Command 93) 5-9
Generate Initialization Vector (Command 94) 5-12
Reformat Initialization Vector (Command 95) 5-14
Verify Initialization Vector (Command 96) 5-16
Encrypt/Decrypt Data (Command 97) 5-18
3DES DUKPT Encrypt/Decrypt Data (Command 388) 5-26

6. Authenticating Transaction Data

About Data Authentication 6-1
Data Authentication Tasks 6-1
Authentication All at Once 6-2
Authentication in Batches 6-2
Verification in VISA UKPT Networks 6-3
Data Authentication Commands 6-3
Quick Reference 6-3
MAC Translate (Command 58) 6-5
Generate MAC and Encrypt or Translate Data (Command 59) 6-13
Verify and Generate MAC for VISA UKPT (Command 5C) 6-25
Verify MAC and Decrypt PIN (Command 5F) 6-29
Generate MAC (Command 98) 6-34
Verify MAC (Command 99) 6-40

Verify ACR (Atalla Challenge Response) Response MAC (Command 9B) 6-46
Verify DUKPT MAC (Command 348) 6-51
Generate DUKPT MAC (Command 386) 6-55

7. Authorizing VISA, MasterCard, American Express, and Discover Cards

About CVVs, CVCs, and CSCs 7-1
CVV, dCVV, CVC, CVC3, and CSC Commands 7-2
Quick Reference 7-2
Generate CVV/CVC (Command 5D) 7-3
Verify CVV/CVC (Command 5E) 7-6
Verify dCVV (Command 357) 7-9
Verify dynamic CVC3 (Command 359) 7-12
Verify AMEX CSC (Command 35A) 7-16
Generate AMEX CSC (Command 35B) 7-20
Verify Discover DCVV (Command 35F) 7-23
Verify AMEX Expresspay value - Magstrip Mode (Command 36A) 7-26

8. Processing EMV and Visa Stored Value Cards

EMV Master Key Derivation 8-1
VSVC Signatures 8-1
DES Key Management for VSVC 8-2
VSVC Data Elements 8-3
Quick Reference 8-4
Verify VSVC S1 Signature and Generate VSVC S2 Signature (Command BE) 8-5
Verify VSVC S3 Signature (Command BF) 8-10
Verify EMV ARQC (Command 350) 8-14
EMV PIN Change (Command 351) 8-23
Generate EMV MAC (Command 352) 8-31
Generate EMV ICC Master Key (Command 354) 8-38
Validate CAP Token (Command 356) 8-42

9. Storing Values in the Volatile Table

About the Volatile Table 9-1
Referencing a location 9-1
Volatile Table Tasks 9-1
Loading the Volatile Table 9-1
Verifying Values in the Volatile Table 9-2
Deleting Values from the Volatile Table 9-2
Volatile Table Commands 9-2

Quick Reference 9-2
Load Volatile Table Value (Command 70) 9-3
Delete Volatile Table Value (Command 71) 9-6
Verify Volatile Table Value (Command 72) 9-8
Clear Volatile Table (Command 73) 9-10
Load Diebold Number Table (Command 74) 9-12
Load Value to a Specific Volatile Table Location (Command 7F) 9-15

10. Printing Commands

Letter template file 10-1
Printing an encrypted PIN 10-3
Printing a key component 10-4
Printing a test page 10-4
HP Printers 10-5
Combine Key Components (Command 15E) 10-6
Generate PIN Printing Key (Command 160) 10-10
Print PIN Letter (Command 161) 10-12
PIN Issuance: IBM 3624 Method (Command 162) 10-19
PIN Issuance: Visa Method (Command 163) 10-26
Divide a Key into Components (Command 16E) 10-31
Print Component Letter (Command 16F) 10-35

11. Utility Commands

Quick Reference 11-1
Echo Test Message (Command 00) 11-4
Security Processor Clear Log (Command 9A) 11-6
Security Processor Configuration Status (Command 9A) 11-8
Security Processor Count Status (Command 9A) 11-12
Security Processor Crypto Test (Command 9A) 11-15
Security Processor Status ID (Command 9A) 11-17
Security Processor Status Key (Command 9A) 11-25
Configure Security Processor Option (Command 101) 11-29
Command Monitoring (Command 102) 11-32
Enable Premium Value Commands and Options (Command 105) 11-37
Define Temporary Serial Number (Command 106) 11-40
Confirm Temporary Serial Number (Command 107) 11-44
Define Security Policy (Command 108) 11-47
Confirm Security Policy (Command 109) 11-55
Get ID of Current Image (Command 1101) 11-58
Get Virtual NSP Information (Command 1102) 11-60

Get Temporary Serial Number Information (Command 1104) 11-62
License Premium Value Commands/Options in all Virtual NSPs (Command 1105) 11-64
Get System Configuration Information (Command 1110) 11-67
Get System Date and Time (Command 1111) 11-69
Get Average CPU Utilization (Command 1113) 11-71
Get System Information (Command 1120) 11-73
Get Log Signing Key Certificate (Command 1204) 11-75
Get Battery Life Remaining (Command 1216) 11-78
Return IP Address of NSP (Command 1221) 11-80
TCP/IP Socket Information (Command 1223) 11-82
Get Application Key Check Digits (Command 1226) 11-85
Reset to Factory State (Command 1227) 11-87
Confirm Reset to Factory State (Command 1228) 11-89
Select Virtual NSP (Command 1350) 11-91
Virtual NSP System Information (Command 1351) 11-93

12. Error Messages

Application Error Messages 12-1
Detailed Errors 12-2

A. Introduction to Cryptography

Data Encryption Standard (DES) A-1
Message Authentication A-1
Triple DES (3DES) A-1
Key Attributes A-4
Key Length A-4
Key Components A-5
Key Parity A-6
Weak and Semi-weak DES Keys A-7
Sample Clear-text Key Component Form A-8

B. Understanding Financial Interchange Networks

Overview B-1
Initializing the Financial Interchange Network B-2
Purpose B-2
Initialization Checklist B-3

C. Summary of Commands and Options

Network Security Processor Options C-18

Recommended settings for security options C-25

D. Contacting Atalla

24-hour Support D-1
On-site Support D-2

Glossary

Index

Figures

Figure 4-1. PIN Block 4-5
Figure 4-2. Account Number Block 4-6
Figure 4-3. IBM 3624 PIN Block 4-7
Figure 4-4. Encrypted IBM 3624 PIN Block 4-8
Figure 4-5. PIN/Pad Character PIN Block 4-9
Figure 4-6. Docutel PIN Block 4-10
Figure 4-7. IBM Encrypting PIN Pad 4-11
Figure 4-8. Burroughs PIN Block Type 4-12
Figure 4-9. ISO-3 PIN Block 4-13
Figure 4-10. ISO-3 Account Number Block 4-13
Figure 4-11. IBM 4731 PIN Block 4-15
Figure 4-12. IBM 4731 ICV 4-15
Figure 4-13. Encrypted IBM 4731 PIN Block 4-16
Figure A-1. TDEA Electronic Codebook A-2
Figure A-2. TDEA Cipher Block Chaining - Encryption A-3
Figure A-3. TDEA Cipher Block Chaining - Decryption A-4
Figure B-1. Simple Financial Interchange Network B-1
Figure B-2. Key Sharing B-2

Tables

Table 2-1. Supported key types 2-3
Table 3-1. Initialization commands 3-1
Table 3-2. Command 10: Generate Working Key, Any Type 3-5
Table 3-3. Response 20: Generate Working Key, Any Type 3-5
Table 3-4. Command 11: Translate Working Key for Distribution 3-8
Table 3-5. Response 21: Translate Working Key for Distribution 3-8
Table 3-6. Command 13: Translate Working Key for Local Storage Switch-to-Switch 3-11
Table 3-7. Response 23: Translate Working Key for Local Storage Switch-to-Switch 3-11

Table 3-8. Command 14: Load ATM Master Key – Diebold 3-14
Table 3-9. Response 24: Load ATM Master Key – Diebold 3-15
Table 3-10. Command 14: Load ATM Master Key – IBM 3624 3-17
Table 3-11. Response 24: Load ATM Master Key – IBM 3624 3-18
Table 3-12. Command 14: Load ATM Master Key – IBM 4731 3-20
Table 3-13. Response 24: Load ATM Master Key – IBM 4731 3-21
Table 3-14. Command 15: Change ATM Communications Key – Diebold 3-24
Table 3-15. Response 25: Change ATM Communications Key – Diebold 3-24
Table 3-16. Command 15: Change ATM Communications Key – Docutel 3-27
Table 3-17. Response 25: Change ATM Communications Key – Docutel 3-27
Table 3-18. Command 15: Change ATM Communications Key – IBM 3624 3-30
Table 3-19. Response 15: Change ATM Communications Key – IBM 3624 3-31
Table 3-20. Command 15: Change ATM Communications Key – IBM 4731 3-33
Table 3-21. Response 25: Change ATM Communications Key – IBM 4731 3-34
Table 3-22. Command 16: Encrypt Financial Institution Table – Diebold: 3-36
Table 3-23. Response 26: Encrypt Financial Institution Table – Diebold 3-36
Table 3-24. Command 16: Encrypt Financial Institution Table – Docutel 3-38
Table 3-25. Response 26: Encrypt Financial Institution Table – Docutel 3-39
Table 3-26. Command 16: Encrypt Financial Institution Table – IBM 3624 3-41
Table 3-27. Response 26: Encrypt Financial Institution Table – IBM 3624 3-42
Table 3-28. Command 18: Generate VISA Working Key 3-44
Table 3-29. Response 28: Generate VISA Working Key 3-45
Table 3-30. Command 19: Translate Communications Key for Local Storage 3-47
Table 3-31. Response 29: Translate Communications Key for Local Storage 3-47
Table 3-32. Command 1A: Translate Working Key for Distribution to Non-Atalla Node 3-50
Table 3-33. Response 2A: Translate Working Key for Distribution to Non-Atalla Node 3-50
Table 3-34. Command 1D: Translate Communications Key for Local Storage Using Specific Variant 3-53
Table 3-35. Response 2D: Translate Communications Key for Local Storage Using Specific Variant 3-53
Table 3-36. Command 1E: Generate New Initial Key for PIN Pad Using VISA DUKPT 3-56
Table 3-37. Response 2E: Generate New Initial Key for PIN Pad Using VISA DUKPT 3-57
Table 3-38. Command 7E: Generate Check Digits 3-60
Table 3-39. Command 9E: Translate Working Key for Local Storage Under Current MFK to Pending MFK 3-63

Table 3-40. Response AE: Translate Working Key for Local Storage Under Current MFK to Pending MFK 3-64
Table 3-41. Command 9F: Replace Current MFK with Pending MFK 3-66
Table 3-42. Response AF: Replace Current MFK with Pending MFK 3-67
Table 3-43. Translate an encrypted key between ECB and CBC modes 3-70
Table 3-44. Response 213: Translate an encrypted key between ECB and CBC modes 3-70
Table 3-45. Command 11D: Generate ATM MAC or Data Encryption Key 3-73
Table 3-46. Response 21D: Generate ATM MAC or Data Encryption Key 3-73
Table 4-1. ANSI - PIN Block Data 4-5
Table 4-2. IBM 3624 - PIN Block Data 4-7
Table 4-3. PIN/Pad - PIN Block Data 4-9
Table 4-4. Docutel - PIN Block Data 4-10
Table 4-5. IBM Encrypting PIN Pad - PIN Block Data 4-11
Table 4-6. Burroughs - PIN Block Data 4-12
Table 4-7. ISO-3 - PIN Block Data 4-13
Table 4-8. IBM 4731 - PIN Block Data 4-15
Table 4-9. VISA DUKPT - PIN Block Data 4-17
Table 4-10. PIN Processing Commands 4-21
Table 4-11. Command 30: Encrypt PIN 4-23
Table 4-12. Response 40: Encrypt PIN 4-24
Table 4-13. Command 31: Translate PIN 4-27
Table 4-14. Response 41: Translate PIN 4-28
Table 4-15. Command 31: Translate PIN – VISA DUKPT 4-31
Table 4-16. Response 41: Translate PIN – VISA DUKPT 4-32
Table 4-17. Command 32: Verify PIN – Identikey 4-37
Table 4-18. Response 42: Verify PIN – Identikey 4-38
Table 4-19. Command 32: Verify PIN – IBM 3624 4-43
Table 4-20. Response 42: Verify PIN – IBM 3624 4-44
Table 4-21. Command 32: Verify PIN – VISA 4-48
Table 4-22. Response 42: Verify PIN – VISA 4-49
Table 4-23. Command 32: Verify PIN – Atalla DES Bilevel 4-53
Table 4-24. Response 42: Verify PIN – Atalla DES Bilevel 4-54
Table 4-25. Command 32: Verify PIN – Diebold 4-57
Table 4-26. Response 42: Verify PIN – Diebold 4-59
Table 4-27. Command 32: Verify PIN – NCR 4-63
Table 4-28. Response 42: Verify PIN – NCR 4-65
Table 4-29. Command 32: Verify PIN – Clear-PIN Comparison 4-68
Table 4-30. Response 42: Verify PIN – Clear-PIN Comparison 4-69

Table 4-31. Command 32: Verify PIN – PIN-Block Comparison 4-71
Table 4-32. Response 42: Verify PIN – PIN-Block Comparison 4-71
Table 4-33. Command 32: Verify PIN – Burroughs 4-74
Table 4-34. Response 42: Verify PIN – Burroughs 4-76
Table 4-35. Command 32: Verify PIN –Atalla 2x2 4-79
Table 4-36. Response 42: Verify PIN – Atalla 2x2 4-80
Table 4-37. Command 33: Translate PIN – ANSI to PLUS, PLUS to ANSI 4-83
Table 4-38. Response 43: Translate PIN – ANSI to PLUS, PLUS to ANSI 4-84
Table 4-39. Command 33: Translate PIN – ANSI to PIN/Pad 4-86
Table 4-40. Response 43: Translate PIN – ANSI to PIN/Pad 4-87
Table 4-41. Command 33: Translate PIN – ANSI to IBM 4731 4-89
Table 4-42. Response 43: Translate PIN – ANSI to IBM 4731 4-90
Table 4-43. Command 33: Translate PIN – IBM 3624 to IBM 3624 4-93
Table 4-44. Response 43: Translate PIN – IBM 3624 to IBM 3624 4-94
Table 4-45. Command 33: Translate PIN – IBM 3624 to PIN/Pad 4-97
Table 4-46. Response 43: Translate PIN – IBM 3624 to PIN/Pad 4-98
Table 4-47. Command 33: Translate PIN – PIN/Pad or Docutel to ANSI 4-101
Table 4-48. Response 43: Translate PIN – PIN/Pad or Docutel to ANSI 4-102
Table 4-49. Command 33: Translate PIN – PIN/Pad or Docutel to PIN/Pad 4-104
Table 4-50. Response 43: Translate PIN – PIN/Pad or Docutel to PIN/Pad 4-105
Table 4-51. Command 33: Translate PIN – PIN/Pad or Docutel to IBM 4731 4-107
Table 4-52. Response 43: Translate PIN – PIN/Pad or Docutel To IBM 4731 4-108
Table 4-53. Command 33: IBM 4731 to ANSI 4-111
Table 4-54. Response 43: IBM 4731 to ANSI 4-112
Table 4-55. Command 33: IBM 4731 to PIN/Pad 4-115
Table 4-56. Response 43: IBM 4731 to PIN/Pad 4-116
Table 4-57. Command 33: IBM 4731 to IBM 4731 4-119
Table 4-58. Response 43: IBM 4731 to IBM 4731 4-120
Table 4-59. Command 35: Translate PIN – Double-Encrypted Input or Output 4-123
Table 4-60. Response 45: Translate PIN – Double-Encrypted Input or Output 4-124
Table 4-61. Command 36: Verify Double-Encrypted PIN 4-127
Table 4-62. Response 46: Verify Double-Encrypted PIN 4-127
Table 4-63. Command 37: PIN Change – Identikey 4-131
Table 4-64. Response 47: PIN Change – Identikey 4-132
Table 4-65. Command 37: PIN Change - IBM 3624 4-137
Table 4-66. Response 47: PIN Change - IBM 3624 4-138
Table 4-67. Command 37: PIN Change – VISA 4-143

Table 4-68. Response 47: PIN Change – VISA 4-144
Table 4-69. Command 37: PIN Change – Atalla DES BiLevel 4-148
Table 4-70. Response 47: PIN Change – Atalla DES BiLevel 4-149
Table 4-71. Command 37: PIN Change – Diebold 4-154
Table 4-72. Response 47: PIN Change – Diebold 4-155
Table 4-73. Command 37: PIN Change – NCR 4-160
Table 4-74. Response 47: PIN Change – NCR 4-161
Table 4-75. Command 39: Translate PIN and Generate MAC 4-164
Table 4-76. Response 49: Translate PIN and Generate MAC 4-165
Table 4-77. Command 3D: Generate PVN and IBM Offset 4-170
Table 4-78. Response 4D: Generate PVN and IBM Offset 4-171
Table 4-79. Command 90: Decrypt PIN 4-173
Table 4-80. Response A0: Decrypt PIN 4-173
Table 4-81. Command BA: PIN Translate (ANSI to PIN/Pad) and MAC Verification 4-176
Table 4-82. Response CA: PIN Translate (ANSI to PIN/Pad) and MAC Verification 4-177
Table 4-83. Command BB: Translate PIN (ANSI to PLUS) and Verify MAC 4-180
Table 4-84. Response CB: Translate PIN (ANSI to PLUS) and Verify MAC 4-181
Table 4-85. Command BD: Translate PIN and Generate ATM MAC 4-187
Table 4-86. Response CD: Translate PIN and Generate ATM MAC 4-189
Table 4-87. Command D0: Verify Clear PIN 4-192
Table 4-88. Response EO: Verify Clear PIN 4-192
Table 4-89. Command 11E: Generate Atalla 2x2 PVN 4-195
Table 4-90. Response 21E: Generate Atalla 2x2 PVN 4-196
Table 4-91. Command 30A: Calculate PIN Offset 4-199
Table 4-92. Response 40A: Calculate PIN Offset 4-199
Table 4-93. Command 32C: Verify ePIN Offset 4-203
Table 4-94. Response 42C: Verify ePIN Offset 4-203
Table 4-95. Command 335: PIN and PIN-Block Translate 4-207
Table 4-96. Response 435: PIN and PIN-Block Translate 4-209
Table 4-97. Command 37B: Generate ePIN Offset 4-213
Table 4-98. Response 47B: Generate ePIN Offset 4-213
Table 5-1. Data Processing Commands 5-3
Table 5-2. Command 55: Encrypt or Decrypt Data or Translate Link L to Link J 5-6
Table 5-3. Response 65: Encrypt or Decrypt Data or Translate 5-7
Table 5-4. Command 93: Generate Random Number 5-9
Table 5-5. Response A3: Generate Random Number 5-10
Table 5-6. Command 94: Generate Initialization Vector 5-12

Table 5-7. Response A4: Generate Initialization Vector 5-13
Table 5-8. Command 95: Reformat Initialization Vector 5-14
Table 5-9. Response A5: Reformat Initialization Vector 5-15
Table 5-10. Command 96: Verify Initialization Vector 5-16
Table 5-11. Response A6: Verify Initialization Vector 5-17
Table 5-12. Command 97: Encrypt/Decrypt Data 5-20
Table 5-13. Response A7: Encrypt/Decrypt Data 5-21
Table 5-14. Command 388: 3DES DUKPT Encrypt/Decrypt Data 5-27
Table 5-15. Response 488: 3DES DUKPT Encrypt/Decrypt Data 5-29
Table 6-1. Data Authentication Commands 6-3
Table 6-2. Command 58: MAC Translate 6-8
Table 6-3. Response 68: MAC Translate 6-10
Table 6-4. Command 59: ECB-Mode Encryption 6-15
Table 6-5. Command 59: CBC-Mode Encryption 6-16
Table 6-6. Command 59: ECB-Mode Translation 6-18
Table 6-7. Command 59: CBC-Mode Translation 6-19
Table 6-8. Response 69: ECB-Mode 6-21
Table 6-9. Response 69: CBC-Mode 6-22
Table 6-10. Command 5C: Verify and Generate MAC for VISA UKPT 6-26
Table 6-11. Response 6C: Verify and Generate MAC for VISA UKPT 6-26
Table 6-12. Command 5F: Verify MAC and Decrypt PIN 6-31
Table 6-13. Response 6F: Verify MAC and Decrypt PIN 6-32
Table 6-14. Command 98: Generate MAC 6-36
Table 6-15. Response A8: Generate MAC 6-37
Table 6-16. Command 99: Verify MAC 6-42
Table 6-17. Response A9: Verify MAC 6-43
Table 6-18. Command 9B: Verify Response MAC 6-47
Table 6-19. Response AB: Verify Response MAC 6-48
Table 6-20. Command 348: Verify DUKPT MAC 6-53
Table 6-21. Response 448: Verify DUKPT MAC 6-54
Table 6-22. Command 386: Generate DUKPT MAC 6-57
Table 6-23. Response 486: Generate DUKPT MAC 6-58
Table 7-1. CVV, dCVV, CVC, CVC3 and CSC Commands 7-2
Table 7-2. Command 5D: Generate CVV/CVC 7-4
Table 7-3. Response 6D: Generate CVV/CVC 7-4
Table 7-4. Command 5E: Verify CVV/CVC 7-7
Table 7-5. Response 6E: Verify CVV/CVC 7-8
Table 7-6. Command 357: Verify dCVV 7-10
Table 7-7. Response 457: Verify dCVV 7-10

Table 7-8. Command 359: Verify dynamic CVC3 7-13
Table 7-9. Response 459: Verify dynamic CVC3 7-14
Table 7-10. Command 35A: Verify AMEX CSC 7-17
Table 7-11. Response 45A: Verify AMEX CSC 7-18
Table 7-12. Command 35B: Generate AMEX CSC 7-21
Table 7-13. Response 45B: Generate AMEX CSC 7-21
Table 7-14. Command 35F: Verify Discover DCVV 7-24
Table 7-15. Response 45F: Verify Discover DCVV 7-25
Table 7-16. Command 36A: Verify AMEX Express pay value - Magstripe Mode 7-27
Table 7-17. Response 46A: Verify AMEX Express pay value - Magstripe Mode 7-28
Table 8-1. VSVC Data Elements 8-3
Table 8-2. VSVC Signature and EMV Commands 8-4
Table 8-3. Command BE: Verify VSVC S1 Signature 8-7
Table 8-4. Response CE: Verify VSVC S1 Signature 8-8
Table 8-5. Command BF: Verify VSVC S3 Signature 8-11
Table 8-6. Response CF: Verify VSVC S3 Signature 8-12
Table 8-7. Command 350: Verify EMV ARQC 8-18
Table 8-8. Response 450: Verify EMV ARQC 8-19
Table 8-9. Command 351: PIN Change – EMV 8-26
Table 8-10. Response 451: PIN Change – EMV 8-28
Table 8-11. Command 352: Generate EMV MAC 8-34
Table 8-12. Response 452: Generate EMV MAC 8-35
Table 8-13. Command 354: Generate ICC Master Key 8-39
Table 8-14. Response 454: Generate ICC Master Key 8-39
Table 8-15. Command 356: Validate CAP Token 8-45
Table 8-16. Response 456: Validate CAP Token 8-46
Table 9-1. Volatile Table Commands 9-2
Table 9-2. Command 70: Load Volatile Table Value 9-3
Table 9-3. Response 80: Load Volatile Table Value 9-4
Table 9-4. Command 71: Delete Volatile Table Value 9-6
Table 9-5. Response 81: Delete Volatile Table Value 9-7
Table 9-6. Command 72: Verify Volatile Table Value 9-8
Table 9-7. Response 82: Verify Volatile Table Value 9-9
Table 9-8. Command 73: Clear Volatile Table 9-10
Table 9-9. Response 83: Clear Volatile Table 9-10
Table 9-10. Command 74: Load Diebold Number Table 9-13
Table 9-11. Response 84: Load Diebold Number Table 9-13

Table 9-12. Command 7F: Load Value to a Specific Volatile Table Location 9-16
Table 9-13. Response 8F: Load Value to a Specific Volatile Table Location 9-16
Table 10-1. Command 15E: Combine Key Components 10-7
Table 10-2. Response 25E: Combine Key Components 10-8
Table 10-3. Command 160: Generate PIN Printing Key 10-10
Table 10-4. Response 260: Generate PIN Printing Key 10-11
Table 10-5. Command 161: Print PIN Letter 10-15
Table 10-6. Response 261: Print PIN Letter 10-16
Table 10-7. Command 162: PIN Issuance: IBM 3624 Method 10-21
Table 10-8. Response 262: PIN Issuance: IBM 3624 Method 10-22
Table 10-9. Command 163: PIN Issuance: Visa Method 10-28
Table 10-10. Response 263: PIN Issuance: Visa Method 10-29
Table 10-11. Command 16E: Divide a Key into Components 10-32
Table 10-12. Response 26E: Divide a Key into Components 10-33
Table 10-13. Command 16F: Print Component Letter 10-38
Table 10-14. Response 26F: Print Component Letter 10-40
Table 11-1. Utility Commands 11-1
Table 11-2. Command 00: Echo Test Message 11-4
Table 11-3. Response 00: Echo Test Message 11-5
Table 11-4. Command 9A: Security Processor CLEAR_LOG 11-6
Table 11-5. Response AA: Security Processor CLEAR LOG 11-7
Table 11-6. Command 9A: Security Processor Configuration Status 11-9
Table 11-7. Response AA: Security Processor Configuration Status 11-10
Table 11-8. Command 9A: Security Processor Count Status 11-12
Table 11-9. Response AA: Security Processor Count Status 11-14
Table 11-10. Command 9A: Security Processor Crypto Test 11-16
Table 11-11. Response AA: Security Processor Crypto Test 11-16
Table 11-12. Command 9A: Security Processor Status ID 11-18
Table 11-13. Response AA: Security Processor Status ID 11-22
Table 11-14. Command 9A: Security Processor Status Key 11-25
Table 11-15. Response AA: Security Processor Status Key 11-27
Table 11-16. Command 101: Configure Security Processor Option 11-30
Table 11-17. Response 201: Configure Security Processor Option 11-30
Table 11-18. Command 102: Command Monitoring 11-33
Table 11-19. Response 202: Command Monitoring 11-34
Table 11-20. Command 105:Enable Premium Value Commands and Options 11-38
Table 11-21. Response 205: Enable Premium Value Commands and Options 11-38
Table 11-22. Command 106: Define Temporary Serial Number 11-41
Table 11-23. Response 206: Define Temporary Serial Number 11-42

Table 11-24. Command 107: Implement Temporary Serial Number 11-44
Table 11-25. Response 207: Implement Temporary Serial Number 11-45
Table 11-26. Command 108: Define Security Policy 11-49
Table 11-27. Response 208: Define Security Policy 11-51
Table 11-28. Command 109: Confirm Security Policy 11-56
Table 11-29. Response 209: Confirm Security Policy 11-56
Table 11-30. Command 1101: Get ID of Current Image 11-58
Table 11-31. Response 2101: Get ID of Current Image 11-59
Table 11-32. Command 1102: Get Virtual NSP Information 11-60
Table 11-33. Response 2102: Get Virtual NSP Information 11-61
Table 11-34. Command 1104: Get Virtual NSP Information 11-62
Table 11-35. Response 2104: Get Temporary Serial Number Information 11-62
Table 11-36. Command 1105: License Premium Value Commands/Options in all Virtual NSPs 11-65
Table 11-37. Response 2105: License Premium Value Commands and Options in all Virtual NSPs 11-65
Table 11-38. Command 1110: Get System Configuration Information 11-67
Table 11-39. Response 2110: Get System Configuration Information 11-68
Table 11-40. Command 1111: Get System Date and Time 11-69
Table 11-41. Response 2111: Get System Date and Time 11-69
Table 11-42. Command 1113: Get Average CPU Utilization 11-71
Table 11-43. Response 2113 Get Average CPU Utilization 11-72
Table 11-44. Command 1120: Get System Information 11-73
Table 11-45. Response 2120: Get System Information 11-74
Table 11-46. Command 1204: Get Log Signing Key Certificate 11-75
Table 11-47. Response 2204: Get Log Signing Certificate 11-75
Table 11-48. Command 1216: Get Battery Life Remaining 11-78
Table 11-49. Response 2216: Get Battery Life Remaining 11-79
Table 11-50. Command 1221: Return IP Address of NSP 11-80
Table 11-51. Response 2221: Return IP Address of NSP 11-81
Table 11-52. Command 1223: TCP/IP Socket Information 11-83
Table 11-53. Response 2223: TCP/IP Socket Information 11-84
Table 11-54. Command 1226: Get Application Key Check Digits 11-85
Table 11-55. Response 2226: Get Application Key Check Digits 11-85
Table 11-56. Command 1227: Reset to Factory State 11-87
Table 11-57. Response 2227: Reset to Factory State 11-88
Table 11-58. Command 1228: Confirm Reset to Factory State 11-89
Table 11-59. Response 2228: Confirm Reset to Factory State 11-90
Table 11-60. Command 1350: Select Virtual NSP 11-91

Table 11-61. Response 2350: Select Virtual NSP 11-92
Table 11-62. Command 1351: Virtual NSP System Information 11-93
Table 11-63. Response 2351: Virtual NSP System Information 11-94
Table 12-1. Error Types 12-1
Table 12-2. Detailed Application Errors 12-2
Table A-1. Weak and Semi-weak Keys A-7
Table C-1. Command Locator C-2
Table C-2. Network Security Processor Options C-18

What's New in This Manual

New and changed information

Version 2.00 is based on version 1.35.

- The Network Security Processor can print cleartext PINs and key components. See Printing Commands for more information.
- The following new utility commands have been added: 102, 1102, 1104, 1105, 1113, 1204, 1350, and 1351.
- Option 87 (Enable/Disable NIC2) is not premium value.
- Command 1227 supports an optional field that when present instructs the Network Security Processor to erase its security audit log.
- On demand self- test can be performed using command <9A#DIAGTEST#....#>.

These new features are documented in the Installation and Operations Guide for the Atalla Ax160 NSP:

- Support for an encrypted communications channel between the host system and the Network Security Processor has been added. Refer to PROTOCOL_ASCII in section 4.
- The Network Security Processor performs startup, once-a-day, and conditional self tests. A record is added to the system log when each test is performed. Refer to DIAGTEST_TIME in section 4.
- The Network Security Processor can be configured to only allow connections from specified host IP addresses. Refer to ALLOWIP in section 4.
- A Deterministic Random Bit Generator (NIST Special Publication 800-90, March 2007) is used to generate random values.
- Support for multiple Master File Keys and security policies (virtual Network Security Processors (VNSPs)) has been added as an option available for purchase on the A10160 model. Refer to MULTI_VNSP in section 4.
- A system log record will be generated when a temporary serial number is defined or reset, and when there are 24 and 12 hours remaining before the temporary serial number expires.

These new features are documented in the SCA-3 User Guide:

- Initialization and configuration operations require the use of an SCA-3 and version 2.0 Security Administrator Smart cards. Refer to sections 1,3,4,5, 6, and 7.
- The SCA-3 supports the ability to adjust the Network Security Processor system time. Refer to NSP Time Adjustment in section 4.

- The SCA-3 supports the ability to halt and then restart the Network Security Processor. Refer to Remote NSP Restart in section 4.
- An SCA-3 that is connected to a personal computer that is running the Remote Management Utility program version 2.0 can send, receive, list and delete files from the Network Security Processor's USB flash memory device. Refer to NSP File in section 8.

Version 1.35 Changes

- The following commands which support the printing of PINs and key components have been added: 15E, 160, 161, 161, 162, 163, 16E, and 16F.
- To support the printing functionality four new parameters have been added to the config.prm file. For information on these parameters see section 4 of the .
- The Ax160 NSP will not start successfully if there are errors in the config.prm file.

Version 1.32 Changes

Version 1.32 is based on version 1.30

- Standard command 36A has been added.

Versions 1.12, 1.15, 1.16, 1.2x, and 1.31 were not released for the variant personality.

Version 1.30 Changes

- Option 4F has been added.
- Command 7E has been modified.
- Commands 35A and 35B have been modified to support CSC-2.
- In command 1216 the maximum value for the battery date counter has been reduced to 700 days.
- The performance of commands that return responses containing large amounts of unpacked ASCII data has been improved.

Version 1.17 Changes

- Customer specific premium value command 3A4 has been added.
- The accuracy of the NIC2 information in the system log has been improved.

Version 1.14 Changes

- Utility command 1223 has been modified to support option 023.

Version 1.13 Changes

- Customer specific premium value command 332 has been modified.
- Premium value option 87 has been added.
- Utility command 105 has been modified to allow lowercase characters in the serial number field.
- The operating system in the Atalla Cryptographic Engine has been updated.

Version 1.11 Changes

- Commands 348, 386 and 35F have been added.
- Two customer specific premium value commands 3A2 and 3A3 have been added.
- The ability to utilize the second Network Interface Connection (NIC2) on the Atalla Cryptographic Engine has been added. The
  contains detailed information; see section 2 for the location of NIC2 and section 4 for configuration parameters.
- Command 1223 has been updated to support NIC2.

About This Manual

Who Should Read This Manual

This manual is written for host application programmers who need to add hardware DES cryptographic support to their applications.

This manual is organized into the following sections:

- Section 1, Introduction, provides an overview of the command and response format, data formats, cryptographic functions supported, and provides information on communicating with the Network Security Processor.
- Section 2, Using DES keys, describes the different types of cryptographic DES keys used by the Network Security Processor. It also explains the differences between single and triple DES. Key parity, and the use of variants is also covered in this section.
- Section 3, DES key management, describes the commands and responses used to generate and or translate working keys for use in an ATM, POS, and EFT networks.
- Section 4, Processing Personal Identification Numbers, describes the commands and responses used to encrypt, generate, translate, and verify PINs.
- Section 5, Processing Transaction Data, describes the command and responses used to encrypt and decrypt data, and generate random numbers.
- Section 6, Authenticating Transaction Data, describes the commands and responses used to generate and verify message authentication codes.
- Section 7, Authorizing VISA, MasterCard, American Express, and Discover Cards, describes the commands and responses used to generate and verify Card Verification Values (CVV), Card Validation Codes (CVC), and Card Security Codes (CSC).
- Section 8, Processing EMV and Visa Stored Value Cards, describes the commands and responses used to generate and verify S1, S2, and S3 signatures.
- Section 9, Storing Values in the Volatile Table, describes the commands and responses used to store, and delete keys from the volatile table.
- Section 10, Printing Commands, describe the commands and responses that support printing PINs or key components.
- Section 11, Utility Commands, describes the utility commands, and provides their calling and responding parameters.
- Section 12, Error Messages, defines the error response format and lists the application error types.
- Appendix A, Introduction to Cryptography, describes cryptographic standards and terms.

- Appendix B, Understanding Financial Interchange Networks, explains how to initialize a network.
- Appendix C, Summary of Commands and Options, is a reference that lists commands and where they are located in the manual.
- Appendix D, Contacting Atalla, provides email and telephone contact information.
- Glossary, provides definitions of terms used in this manual.

The manual is provided in “electronic” form, as a.PDF file. PDF files can be viewed with Adobe Acrobat. Hypertext links are included to allow you to quickly locate specific information.

Your comments invited

After using this manual, please take a moment to send us your comments via an email message. Be sure to include your name, company name, address, and phone number in your message. If your comments are specific to a particular manual, also include the part number and title of the manual.

The email address is: Atalla.Support@HP.Com.

Many of the improvements you see in manuals are a result of suggestions from our customers. Please take this opportunity to help us improve future manuals.

Related documents

-

If you purchase a Secure Configuration Assistant-3, you will receive the following document:

-

Type conventions

Hypertext links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described.

For example:
See Data formats on page 1-4 for information on how to include these special characters in your command data.

Key presses

Keys you press are shown in boldface Helvetica type.

Example: Press the **clear** key to return to the Main Menu.

Emphasis

Words that are emphasized are shown in italic or bold.

Example: You create a Master File Key (MFK).

Key cryptogram notation

Key values are sent to the Network Security Processor in an encrypted form. The notation:

$E_{MFK.V}$(Working Key)

The first character (either E or D) indicates the DES operation (encryption or decryption). The subscripted value is the encrypting/decrypting key and any variant. The value in parenthesis is being operated on. The example above indicates that the Working Key has been encrypted under a specific variant of the Master File Key.

To aid readability, long strings of characters, such as key cryptograms, will be split into groups of four characters. Do not include these spaces when sending commands. For example:

The clear-text key value: 0123456789ABCDEF will be shown as:

```
0123 4567 89AB CDEF
```

Examples

Examples and explanations are shown in Courier type.

Example:

```
COMMAND=<101#023E#>
RESPONSE=<201#Y#>
```

Optional fields

Fields in the command and response syntax descriptions that are surrounded by square brackets are optional. The location of the closing square bracket is significant. If the field delimiter (#), precedes the closing square bracket the entire field is optional. If the field delimiter (#) follows the closing square bracket the field is required but can be empty. For example:

The key length field is optional:

<10#Variant#$E_{MFK.0}$(KEK)#[Key Length#]>

The key length field is required but can be empty:

<10#Variant#$E_{MFK.0}$(KEK)#[Key Length]#>

Even page numbering

Each section in this manual ends on an even page, even if the page is blank. This practice enables each section to start on an odd-numbered page, which helps give the manual a consistent appearance.

1 Introduction

This section describes the cryptographic functions supported by the Network Security Processor, the command and response message format, error reporting, and data formats.

Cryptographic functions

Network Security Processor support the following cryptographic functions:

- DES key management - key generation and key translation.
- Processing Personal Identification Numbers - encrypting, translating, and verifying PINs.
- Processing Transaction Data - encrypting and decrypting data.
- Authenticating Transaction Data - generating and verifying Message Authentication codes.
- Authorizing VISA, MasterCard, American Express, and Discover Cards - generating and verifying credit and debit authorization codes
- Processing EMV and Visa Stored Value Cards - generating and verifying Authorization Request Cryptograms, generating message authentication codes, and generating and verifying VCSC signatures.

Operating overview

The Network Security Processor must be initialized with a Master File Key before it can process a cryptographic command. Utility Commands do not require the Network Security Processor to be initialized with a Master File Key.

Network Security Processor operation occurs in three phases:

- Command. The host application writes the command to the Network Security Processor.
- Processing. The Network Security Processor performs the requested actions based on the specific commands received.
- Response. The Network Security Processor returns the response. The host reads the response.

Command and response

The application programming interface consists of a set of specific commands to which a response or error message is returned. The host application must send the command as a contiguous strings of characters. The TCP/IP message that contains

the command to be processed by the Network Security Processor must end with the "#>" end-of-command characters.

Note. The host application must not send any additional characters after the "#>" end-of-command characters.

To fit the page layout of this manual, command and response syntax descriptions, and examples, are when necessary, split into multiple lines usually at a field boundary. The response to the second example below is typical of a multi-line response.

Commands are identified by an ID and have the following format:

```
<CMDID#FIELD 1#FIELD 2#FIELD N#[^Context Tag#]>
```

where:

<	Starts the command
CMDID	Is the two, three, or four character Command ID
#	Is a delimiter after each command field (including the last field)
Field	Is the command data (fields vary in length and number)
^	Context Tag
>	Ends the command

Characters preceding the "<" start of command character are ignored. Characters following the ">" end of command character are also ignored. These four characters ("<", "#", "^", and ">") have special meaning to the Network Security Processor, and therefore cannot be included in the command data fields. In Ethernet TCP/IP communications, the carriage return (CR) character is also a special character that is interpreted as an end-of-command, it also cannot be included in the command data. See Data formats on page 1-4 for information on how to include these special characters in your command data.

Any cryptographic command can include an additional field after all required fields. This field is optional, and can be used to supply "context" information which is returned with the response message. The first character of this field must be ASCII hexadecimal 5E (^). The remaining data can be variable in length but it may not contain the #, >, <, or CR characters, or exceed the maximum command length of 5,000 characters.

The response format is identical to the command format with the exception that a carriage return CR (hexadecimal 0D) and a line feed LF (hexadecimal 0A) may follow the ">". The carriage return and line feed are denoted as CRLF. This capability is configurable, the default is CRLF is appended to the response; see option 023 to remove the CRLF from the response.

Input commands have odd-numbered first digits; the corresponding response commands are ten digits higher. For example, if 10 is the command ID, then 20 is the response ID; if 31 is the command ID, then 41 is the response ID. See Appendix C, Summary of Commands and Options for a listing of commands.

The following example shows the command and response for Command 10, notice that the CRLF is appended to the response:

Command

```
<10#1#F6F4D93F55860571#>
```

Response

```
<20#4110AD1F7EE6239A#F65C09AA7CD28F8A#82E1#>CRLF
```

The following example shows the command and response for Command 10 using a context tag:

Command

```
<10#1#F6F4D93F55860571#^Generate KPE for ATM 325#>
```

Response

```
<20#4110AD1F7EE6239A#F65C09AA7CD28F8A#82E1#
^Generate KPE for ATM 325#>CRLF
```

Error responses

If the Network Security Processor encounters an error, an error response message is returned. Use the information below to decode the error response. If you are contacting Atalla Technical support for assistance please be sure to provide the exact command and error response.

The format of the error response is:

`<00#XXYYZZ#>`

The response ID of 00 indicates an error is being returned.

`XX`-indicates the error number, Table 12-1, Error Types, on page 12-1 lists the error number and its description.

`YY` – the first field found to be in error. The command ID field is field zero. If this field returns the value 00, then any of the following may be true:

- The command specified an invalid command number.
- A necessary MFK or KEK is missing.
- In response to an echo (Command (ID = 00) command.

ZZ – the software version of the cryptographic command processor. This field returns a two digit software version number, use command <1101#> for more complete information on the software version.

Note. If a binary zero is present in an field that does not allow binary data the context tag will not be present in the error response.

Here is an example of an invalid command 10, (field 2 contains 15 characters instead of 16).

Command

```
<10#1#F6F4D93F5586057#>
```

Response

```
<00#010210#>CRLF
```

The error indicates length out of range in field 2, software version is 1.00.

Detailed errors

The detailed error is appended as a separate field after the error field (XXYYZZ). Detailed errors are included if option 021 is enabled. Table 12-2, Detailed Application Errors, on page 12-2 describes the detailed application error messages.

Here is an example of an invalid command that returns an error and a detailed error.

Command

```
<20#1#F6F4D93F55860571#>
```

Response

```
<00#030010#0201##>CRLF
```

The error indicates value out of range in field 0, software version is 1.00. The detailed error (201) indicates Invalid Command.

Data formats

Commands and responses usually contain only hexadecimal characters. Each character is one byte. The Network Security Processor requires ASCII characters.

There are a limited number of commands that do not have fields that specify data type and length, the command Generate MAC and Encrypt or Translate Data (Command 59) is an example. In some instances the data to be processed may be

unprintable, or contain control characters for some protocols, or include the special characters (“<“, “#”, “^”, “>”, carriage return and line feed). For example, to encrypt or authenticate this message:

```
Sell stock when price is > $50.
```

The “>” character causes the Network Security Processor to incorrectly terminate the message. There are two ways to resolve this issue:

- The host application converts each data character to ASCII hexadecimal. This allows all possible characters to be processed. The example data “Sell stock when price is > $50.” would be converted to:
  `53656C6C2073746F636B207768656E20707269636520973203E20 2435302E`
- Use a command that supports binary data and includes data type and data length fields such as Generate MAC (Command 98).

Programming guidelines

If your host application is running on a NonStop Himalaya system, you can use Boxcar or the Atalla Resource Manager (ARM) to manage the host to Network Security Processor TCP/IP Ethernet interface. See the for information on configuring Boxcar. See the for more information on ARM.

The remainder of this section applies to a Unix host environment. Communicating with the Network Security Processor using Ethernet TCP/IP involves the following basic steps:

- Setting up the application
- Opening the socket interface
- Connecting the socket to the Network Security Processor
- Sending the command
- Receiving the response
- Closing the socket

The following subsections explain each of the above programming steps.

Setting Up the application

The host application should interact with the Network Security Processor in a single-threaded manner. This means you must send a command and wait for a response. Also, if you are using the C programming language in a UNIX environment, be sure to include the following files:

- sys\types.h

- sys\socket.h
- netinet\in.h
- stdio.h

Opening the socket interface

The **socket** system call requires the domain, socket type, and protocol. The Network Security Processor socket should be coded to use Internet domain (AF_INET), a stream socket (SOCK_STREAM), and the default protocol (0) for SOCK_STREAM.

Connecting a socket to the NSP

The **connect** system call is used to establish the connection. When connecting the socket to the Network Security Processor, you need to specify the IP address of the Network Security Processor and the TCP port number. The Network Security Processor supports a maximum of 64 sockets. Requests to open additional sockets will immediately receive a close socket from the Network Security Processor.

Sending commands to the NSP

Use the **send** system call to send the command to the Network Security Processor.

Receiving responses

Use the **recv** system call to received the Network Security Processor response. Since a response from the Network Security Processor can exceed the length of a single Ethernet frame, it is important to code your application in a way that ensures that you have received the entire response from the Network Security Processor. All commands end with these two characters “#>”. If option 23 is disabled, a carriage return and line feed 0x0D and 0x0A are appended after the “#>” characters. A response to a command that contains binary data may have the” #>” characters as part of the response data, therefore if you use Network Security Processor commands with binary data you will need to code your application to check the response data length field to be sure you have read all the data from the Network Security Processor response message.

Closing a socket

The **close** system call is used to terminate the session with the Network Security Processor. The application should close the sockets before the application terminates. Sockets that are not closed remain open, thereby reducing the number of sockets available. Opening and closing a socket for each command is not recommended.

Sample program

Here is a sample program that can be used to communicate with an Ax160 NSP over TCP/IP.

```
/ * Example code for Ax160 TCP/IP communication */

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define MAX_MSG 8192
/*
 * Assume the Ax160 NSP response will be returned in 10 byte chucks. This will
 * demonstrate how to look for the end of a command across multiple packets.
 */

#define PKT_READ_SIZE 10
int
main(void)
{
  char ipaddr[40];              /* IP address */
  int portnumber = 0;           /* IP Port number */
  char message[MAX_MSG];        /* Buffer of message being sent */
  char msgrsp[PKT_READ_SIZE];   /* Buffer of message being read back */
  char retmsg[MAX_MSG];         /* Buffer that contains response */
  int msglen = 0;
  int rcvlen = 0;
  int msg_done = 0;
  int rsp_start = 0;
  int socketnum = 0;
  struct sockaddr_in aname;
  int status = 0;
  int rsp_ptr = 0;
  /*
   * Load IP address, port number, and message
   */
  sprintf(ipaddr, "%s", "192.168.1.100");
  portnumber = 7000;
  sprintf(message, "%s", "<1101#>");
  msglen = strlen(message);
  /*
   * Create a socket
   */
  socketnum = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
  if (socketnum < 0)
  {
    printf("Unable to obtain socket number\n");
    exit(2);
  }
  /*
   * Set socket infomation in the socket structure
   */
  aname.sin_family = AF_INET;
  aname.sin_port = htons(portnumber);
  aname.sin_addr.s_addr = inet_addr(ipaddr);
  /*
   * Connect to the target Ax160
   */
  if (connect(socketnum, &aname, sizeof(aname)) < 0 )
  {
```

```
  printf("Connection error");
  close(socketnum);
  exit(2);
}
 /*
 * Send the message
 */
status = send(socketnum, message, msglen, 0);
if (status < 0)
{
  printf("Unable to send to socket\n");
  close(socketnum);
  exit(2);
}
/*
 * Fetch the response
 */
rsp_start = 0;
do
{
  int i = 0;
  rcvlen = recv(socketnum, msgrsp, (size_t)(PKT_READ_SIZE), 0);
  if (rcvlen < 0)
  {
    printf("Unable to receive from socket\n");
    close(socketnum);
    exit(2);
  }
  if (rcvlen == 0)
  {
    printf("Received 0 length message\n");
    close(socketnum);
    exit(2);
  }
  i = 0;
  if (rsp_start == 0)
  {
     /*
     * Search for the start of the response
     */
    for (; i<rcvlen; i++)
    {
     if (msgrsp[i] == '<')
     {
      /*
        * Found the start of the response
      */
       rsp_start = 1;
       break;
    }
    }
  }
  if (rsp_start != 0)
  {
    /*
```

```
        * We are processing a response, copy characters into the output buffer
        */
       for (; i<rcvlen; i++)
       {
        /*
         * Error if response get too big for the buffer we allocated
         */
        if (rsp_ptr >= MAX_MSG - 1)
        {
          printf("Error: response would overflow buffer\n");
          exit(2);
        }
        retmsg[rsp_ptr++] = msgrsp[i];
        if (msgrsp[i] == '>')
        {
         msg_done = 1;
         break;
        }
       }
     }
       /*
      * Continue to perform socket reads until we get the whole response
      */
   } while (msg_done == 0);
   /*
    * Null terminate the response string for printf
    */
   retmsg[rsp_ptr] = 0;
   /*
    * Output the response
    */
   printf("Message:  %s\n", message);
   printf("Response: %s\n", retmsg);
   close(socketnum);
}
```

2 Using DES keys

A secure financial system network has several types of DES keys, each of which are used for a specific purpose. The majority of these keys are generated by the Network Security Processor, and returned to the host application in two forms. One form is for use by the Network Security Processor. In this form the DES key is encrypted under the Master File Key and stored in the host application's key database. The second form is for use by the remote system. In this form the DES key is encrypted under the Key Exchange Key. The most common uses of these DES keys are to encrypt, translate, and verify PINs, encrypt and decrypt data, generate and verify message authentication codes, and generate and verify card verification values.

The Network Security Processor will not accept or return clear-text DES key values. All DES keys must be supplied encrypted under the Master File Key which resides within the secure boundary of the Network Security Processor. When importing a DES key that was generated on a remote system, the DES must be encrypted under a Key Exchange Key.

DES keys contain 64 bits; they are called single-length keys. Triple DES keys contain 128 bits; they are called double-length or 2key-3DES keys. In the variant personality only the Master File key can triple-length (192 bits), all other keys must be either single or double length. For more information on DES keys, see Key Attributes on page A-4.

Master File Key

The Master File Key (MFK) encrypts Key Exchange Keys and working keys. The MFK is never used to encrypt PINs or data and is never shared with another node. The length of the MFK must be equal to or greater than the length of the Key Exchange Keys and working keys it protects. Security Officers use the Secure Configuration Assistant-3 (SCA-3) to create components of the Master File Key, and then send them to the Network Security Processor. These components are combined to form a secret key which is stored in the Network Security Processor's Non-volatile key table. To minimize downtime, a Pending MFK (PMFK) can be loaded into the Network Security Processor using the same procedure. For more information, see Procedure to replace the current MFK with the pending MFK on page 2-5.

Key Exchange Key

To maintain secrecy, working keys are encrypted under a key called a Key Exchange Key (KEK) before they are sent from one node to another. Key Exchange Keys are unique for each network node. The length of the Key Exchange Key must be equal to or greater than the length of the working keys it protects.

Working keys

Working keys are types of keys used to perform specific cryptographic operations, PIN Encryption Keys and Message Authentication Keys are two examples of working keys

used for a specific purpose. Working keys are stored in the non-volatile key table, they are stored on a host database encrypted using a **specific variant** of the MFK. This encrypted form of the key is called a key cryptogram. When a particular working key is needed to process a transaction, the host sends the cryptogram of the working key to the Network Security Processor where it is decrypted by the MFK and then used to process the transaction data. Most commands accept either 1key-3DES (single-length) or 2key-3DES (double-length) keys, however several commands support only 1key-3DES (single-length) working keys. See the specific command documentation to confirm the key lengths supported. Working keys can also be stored in the Volatile table.

Key variants

Secure cryptography requires that keys be separated according to their intended use. For example, a key may be used as a PIN Encryption Key (KPE) or a Data Key (KD), but not both. This strict categorization is intended to prevent system compromise by substitution or misuse. Each **type of key** is encrypted by a specific variant of either the MFK or a KEK. Variants are produced by performing an exclusive-OR with a fixed value and the first – that is, most significant – byte of each half of the MFK or KEK. Each type of working key is encrypted by a unique version of the MFK or a KEK. The command syntax sections of this manual contain notations similar to this:

$$E_{MFK.V}(\text{Working Key})$$

This represents the working key encrypted under a specific variant of the MFK. For example, a PIN Encryption Key (KPE) is encrypted under variant 1 of the MFK. The notation would be:

$$E_{MFK.1}(\text{KPE})$$

Some commands require the variant to be specified. See Table 2-1 on page 2-3 for a complete list of supported key types.

Variants are unique to Atalla Network Security Processors. When importing or exporting working keys from a node that does not use the Atalla variant method, ensure that the appropriate working key cryptograms are created **without variants**. See commands:

- Generate VISA Working Key (Command 18) on page 3-44
- Translate Communications Key for Local Storage (Command 19) on page 3-46
- Translate Working Key for Distribution to Non-Atalla Node (Command 1A) on page 3-49
- Translate Communications Key for Local Storage Using a Specific Variant (Command 1D) on page 3-52.

Supported key types.

Table 2-1. Supported key types (page 1 of 2)

Variant	Working Key	Abbrev.
0	Key Exchange Key	KEK, KEK-IN
1	PIN Encryption Key	KPE
2	Data or Communication Key	KC
3	Message Authentication Code key	KMAC
3	VISA Card Verification Value Mastercard Card Validation Code	KCVV KCVC
4	PIN Verification Key	KPV
5	ATM A key	AATM
5	ATM B key	BATM
5	ATM master key	KMATM
5	Object Key	KOP
6	Initialization Vector	IV
6	Decimalization/Conversion Table	DECTAB
7	Challenge Response Authentication Key	KMACR
8	Derivation Key	DK
9	Visa VSVC Master Key / EMV Master Key	VSVCMK / MK
10	PIN Encryption Key - Encrypt Only	KPE-EO
11	Custom	MK-DL
12	Custom	PMK
13	Master Message Authentication Key	KMAC-MK
14	Custom	none
15	none	none
16	Data Encrypt Only	ENC
17	Data Decrypt Only	DEC
18	Generate Message Authentication Code only	GMAC
19	Verify Message Authentication Code only	VMAC
20	PIN Encryption Key - Decrypt Only	KPE - DO
21	Custom	none
22	Custom	none
23	Custom	none
24	Custom	none
25	Custom	none

Table 2-1. Supported key types (page 2 of 2)

Variant	Working Key	Abbrev.
26	Custom	none
27	Custom	none
28	Custom	none
29	Custom	none
30	Challenge Data	none
31	Key Exchange Key - Outgoing	KEK-OUT

Key generation and translation

A common use of the Network Security Processor is to protect sensitive information as it travels through an insecure network. DES encryption is typically used for this purpose. A random DES key is used to encrypt the sensitive information at the origin, and the same DES key must be used at the receiving node to successfully decrypt the information. This means that both the origin and destination must share the same DES key. When establishing working keys, a special purpose key called a Key Exchange Key (KEK) is created, and exchanged out-of-band; that is, it is not transmitted over the network. Once both nodes have the same KEK, they can use it to encrypt working keys for transmission between the two nodes.

Ideally, working keys such as PIN Encryption Keys (KPEs), Data Encryption and Decryption (KDs) and Message Authentication Keys (KMACs) are system generated, this insures no one individual knows the key. The Network Security Processor supports a generic key generation command Generate Working Key, Any Type (Command 10) on page 3-4 that can be used to generate any type of key. The generated key is encrypted in two forms one for local storage and use (encrypted under the MFK) and one for export to the remote node (encrypted under the KEK).

To receive an encrypted working key from a remote node it must be translated from encryption under the KEK to encryption under the MFK. See Section 3, DES key management for more information on key generation and key translation commands.

Non-volatile key table

The Network Security Processor has a non-volatile key table that stores the Master File Key and Pending Master File Key. Keys stored in the non-volatile key table are maintained without external power for up to five years. Once loaded into the non-volatile key table, they cannot be extracted, transmitted, or downloaded in clear-text form. Securing the Network Security Processor involves using the Secure Configuration Assistant-3 (SCA-3) to either add an Network Security Processor to an existing security association or create a new security association for the Network Security Processor, and defining and sending a Master File Key to the Network Security Processor. See Security Processor Status Key (Command 9A) on page 11-25 for the command syntax to determine what keys are stored in the non-volatile key

table. See the key table. for the procedures to load keys into the non-volatile

Volatile table

Early model Network Security Processors supported only asynchronous or bisynchronous communications at a maximum baud rate of 19,200 bits per second. Performance was limited by the communications interface. To minimize the number of characters sent in a command, a volatile table was created. The host application was able to preload keys into the table. When a specific key was needed, the index into the table was provided in the command, instead of the 16 character key cryptogram, reducing the length of the command. The benefit was better performance, however the host application became more complex as it now had to manage the table. The Ethernet TCP/IP interface is fast enough such that there is no performance benefit in using the table. The Verify PIN – Diebold (Command 32) on page 4-56, is the only command that requires the use of the volatile table.

See Section 9, Storing Values in the Volatile Table if you decide to use the volatile table. The volatile table can store up to 9,999 1key-3DES (single-length) keys and Diebold Number Tables or 4,998 2key-3DES (double-length) keys.

Procedure to replace the current MFK with the pending MFK

All working key cryptograms encrypted under the current MFK must be translated to encryption to the new MFK. This task can be accomplished manually with the SCA-3, or a more efficient process is to follow the procedure below. The new MFK must be loaded as a pending MFK. This procedure assumes working keys exist encrypted under the current MFK which resides in the Network Security Processor.

1. Using the SCA-3 define and load the pending MFK into the Network Security Processor.
2. Translate all working keys from encryption under the current MFK to the Pending MFK. See Translate Working Key for Local Storage Under the Current MFK to the Pending MFK (Command 9E) on page 3-63 for the command syntax required to perform this task.
3. Replace the current MFK with the pending MFK. See Replace the Current MFK with the Pending MFK (Command 9F) on page 3-66 for the command syntax required to perform this task.
4. Configure the host application to use the new key cryptograms generated in step 2 above.

Security precautions

The Network Security Processor is only as secure as you and your procedures make it. Many attempts to obtain confidential information are performed by employees or other individuals with access to, or knowledge of, security related equipment. Here are some recommendations on keeping your Network Security Processor secure:

- Always keep production cryptographic keys secret. Key components should recorded and stored in a secure location.
- Always define production keys with multiple key components. Never let one individual have access to an entire production key.
- Make sure that key component holders are restricted to their one key component. They should never be allowed to assume the role of another key component holder which would give them access to the entire secret key value.
- Never allow a test key to be used in the same system that has production keys.
- Before migrating a test unit into production, always insure that all test keys have been deleted.
- Whenever possible choose 2key-3DES (double-length) keys.
- When not in use, keep the SCA-3 locked in a secure location.
- The passwords for the SCA-3 Security Administrator and Shareholder smart cards must be kept secret. Each Security Administrator should possess only one smart card.
- Keep the front bezel and access door locked. Store the keys in a secure location. Do not keep the keys in the locks.
- Never use the SCA-3 calculate cryptogram feature to encrypt a key that is known by a single individual. Always validate the source of the key and the business requirement, before you allow it to be entered into your system.
- Configure and secure your system such that only authorized individuals and host applications have access to the Network Security Processor.
- Only enable commands that you have confirmed are required by your host application, all other commands should be disabled by the Network Security Processor’s security policy. Do not enable commands and options listed as a high security exposure until you have confirmed that there is a legitimate business requirement to do so.

3 DES key management

This section contains the information on commands used to support the initialization and management of cryptographic keys in a financial interchange network, see Initializing the Financial Interchange Network on page B-2 for an overview.

Quick reference

Table 3-1 identifies each command by number, name, and purpose. While the table organizes the initialization commands by category, the commands themselves are presented in numerical order.

Table 3-1. Initialization commands (page 1 of 3)

Command Number	Name	Purpose
10	Generate Working Key, Any Type	Generates a variety of working keys. The command returns the generated key in two forms: one for storing locally, encrypted under the specified variant of the MFK, and one for transmitting to another Atalla node, encrypted under the specified variant of the KEK.
18	Generate VISA Working Key	Generates a PIN Encryption Key for use with VISA security processors. The command returns PIN Encryption Key key in two forms: one for storing locally, encrypted under variant 1 of the MFK, and one for transmitting to a non-Atalla node, encrypted under a KEK without a variant.
1E	Generate New Initial Key for PIN Pad Using VISA DUKPT	Reinitializes PIN pads that perform VISA Derived Unique Key Per Transaction (DUKPT) key management.
11D	Generate ATM MAC or Data Encryption Key	Generates either a MAC or Data Encryption Key.
11	Translate Working Key for Distribution	Exports a working key. This command translates a working key from encryption using the specified variant of the MFK, to encryption using the specified variant of any KEK.
13	Translate Working Key for Local Storage, Switch-to-Switch	Imports a working key. This command translates a working key from encryption using the specified variant of any KEK, to encryption using the specified variant of the MFK.

Table 3-1. Initialization commands (page 2 of 3)

Command Number	Name	Purpose
19	Translate Working Key from a Non-Atalla Node for Local Storage	Imports a working key from a non-Atalla node. This command translates a working key from encryption using a base key without a variant, to encryption using a specified variant of the MFK.
1A	Translate Working Key for Distribution to Non-Atalla Node	Exports a working key to a non-Atalla Node. This command translates a working key from encryption using the specified variant of the MFK, to encryption using any KEK without a variant.
1D	Translate Communications Key for Local Storage Using a Specific Variant	Imports a working key from a non-Atalla node. This command translates a communications key from encryption using a base key without a variant, to encryption using the specified variant of the MFK.
9E	Translate Key using Pending MFK	Translates a working key from encryption under the MFK to encryption under the PMFK.
113	Translate Key between modes of DES	Translates a KEK encrypted working key from ECB to CBC, or CBC to ECB mode of DES.
14	Load ATM Master Key – Diebold	Encrypts the ATM Master Key for downloading to Diebold ATMS.
14	Load ATM Master Key – IBM 3624	Encrypts the ATM Master Key for downloading to IBM 3624 ATMs.
14	Load ATM Master Key – IBM 4731	Encrypts the ATM Master Key for downloading to IBM 4731 ATMs.
15	Change ATM Communications Key – Diebold	Encrypts a Communications Key for downloading to Diebold ATMs.
15	Change ATM Communications Key – Docutel	Encrypts a Communications Key for downloading to Docutel ATMs.
15	Change ATM Communications Key – IBM 3624	Encrypts a Communications Key for downloading to an IBM 3624 ATM.
15	Change ATM Communications Key – IBM 4731	Encrypts a Communications key for downloading to an IBM 4731 ATM.
16	Encrypt Financial Institution Table – Diebold	Encrypts keys for downloading to Diebold ATMs financial institution tables.

Table 3-1. Initialization commands (page 3 of 3)

Command Number	Name	Purpose
16	Encrypt Financial Institution Table – Docutel	Encrypts keys for downloading to Docutel ATMs financial institution's tables.
16	Encrypt Financial Institution Table – IBM 3624	Encrypts keys for downloading to IBM 3624 ATMs financial institution tables.
7E	Generate Check Digits	Generates check digits for a key encrypted under the MFK.
9F	Make Pending MFK Current	Replaces the current MFK with the Pending MFK.

Generate Working Key, Any Type (Command 10)

Command 10 generates a variety of working keys that are either 1key-3DES (single-length) or 2key-3DES (double-length). The odd-parity key is generated in two forms: one for storing locally, encrypted under the appropriate variant of the MFK, and one for transmitting to another network node encrypted under the appropriate variant of the KEK.

This command is enabled in the Network Security Processor's default security policy.

Command

<10#Variant#[$E_{MFK.0}$(KEK)]#[Key Length#]>

Response

<20#$E_{MFK.V}$(Working Key)#$E_{KEK.V}$(Working Key)#
Working Key Check Digits#>[CRLF]

Calling Parameters

10

Field 0, the command identifier.

Variant

Field 1, the MFK and KEK variant (v) used to encrypt the generated working key, thus establishing its function. This field can be one or two bytes long and can contain the numbers 0 - 31. See Key variants on page 2-2 for a list of supported variants.

[$E_{MFK.0}$(KEK)]

Field 2, the KEK encrypted under variant zero of the MFK. This key is used to encrypt the randomly generated working key. This field can contain a 16 or 32 byte hexadecimal value, a volatile table location, or can be empty. If this field is empty field 2 of the response will also be empty.

[Key Length#]

Field 3, an optional field used to specify the length of the generated working key. A value of "S" indicates a 1key-3DES (single-length) working key. A value of "D" indicates a 2key-3DES (double-length) working key. If this field contains a "D", then field 2 must be 32 bytes, or reference a volatile table location that contains a 2key-3DES (double-length) key. If this field is not included, a 1key-3DES (single-length) working key will be generated.

Table 3-2. Command 10: Generate Working Key, Any Type

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	10
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.0}$(KEK)*	0, 16, 32	0 - 9, A - F
3	Key Length	0,1	S,D

*Can be a volatile table location.

Responding Parameters

`20`

Field 0, the response identifier.

$E_{MFK.V}$`(Working Key)`

Field 1, the working key encrypted using the variant of the MFK specified when you issued the command. The host application stores this cryptogram on its local database for subsequent use. This field contains a 16 or 32 byte hexadecimal value.

$E_{KEK.V}$`(Working Key)`

Field 2, the working key encrypted using the variant of the KEK specified in field one of the command. The host application transmits this cryptogram to the network node that uses this KEK. This field contains either 16 or 32 byte hexadecimal value, or if field 2 of the command is empty this field will be empty.

`Working Key Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the working key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-3. Response 20: Generate Working Key, Any Type

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	20
1	$E_{MFK.V}$(Working Key)	16, 32	0 - 9, A - F
2	$E_{KEK.V}$(Working Key)	0, 16, 32	0 - 9, A - F
3	Working Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Generate the KEK and obtain the cryptogram of it encrypted under variant zero of the MFK. Store this cryptogram in your host application database.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

This command generates a random working key therefore your test results will not match these examples.

Generate a 1key-3DES (single-length) PIN Encryption Key.

- Variant: 1.
- Clear-text KEK: 0123 4567 89AB CDEF.
  The KEK encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E.

The command looks like this:

```
<10#1#9007B8751BB7AB4E#>
```

The Network Security Processor returns the following response:

```
<20#35F25A7EBD9F789A#80AD2AE8BCA3D9B6#255E#>
```

Generate a 2key-3DES (double-length) PIN Encryption Key.

- Variant: 1.
- Clear-text KEK: 0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The KEK encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E 0B17 6C3E BEED 18AF.

The command looks like this:

```
<10#1#9007B8751BB7AB4E0B176C3EBEED18AF#D#>
```

The Network Security Processor returns the following response:

```
<20#10A0EA9CFCA1A165BF18BB2A3528DFD9#
335A6BA90E2D4B400C61C650F4699ED6#6F93#>
```

Generate a 2key-3DES (double-length) PIN Verification Key that is not encrypted under the KEK.

- Variant: 4.

The command looks like this:

```
<10#4##D#>
```

The Network Security Processor returns the following response:

```
<20#2A5133BC5DC0297BBEA70E1E2CF8DDEE##6F93#>
```

Translate Working Key For Distribution (Command 11)

Command 11 translates a working key of any type, from encryption using the specified variant of the MFK to encryption using the specified variant of the KEK. Use this command to export a working key to another node that uses Atalla Variant Network Security Processors. Your node and the remote node must have the same KEK. This command translates both 1key-3DES (single-length) and 2key-3DES (double-length) working keys.

This command is enabled in the Network Security Processor's default security policy.

Command

<11#Variant#$E_{MFK.0}$(KEK)#$E_{MFK.V}$(Working Key)#>

Response

<21#$E_{KEK.V}$(Working Key)#Working Key Check Digits#>[CRLF]

Calling Parameters

`11`

Field 0, the command identifier.

`Variant`

Field 1, the variant (V) of the MFK under which the working key has been encrypted, and the variant to be applied when generating the output cryptogram (Field 1 of the response). This field can be one or two bytes long and can contain the numbers 0-31. See Key variants on page 2-2 for a list of supported variants.

$E_{MFK.0}$(KEK)

Field 2, the cryptogram of the KEK on the network node to which this working key will be transmitted. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.V}$(Working Key)

Field 3, the cryptogram of the working key encrypted using the variant of the MFK specified in Field 1. If this field contains a 2key-3DES (double-length) key, then Field 2 must also contain a 2key-3DES (double-length) key, or a reference to a volatile table location that contains a 2key-3DES (double-length) key. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

Table 3-4. Command 11: Translate Working Key for Distribution

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	11
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.0}$(KEK)*	16, 32	0 - 9, A - F
3	$E_{MFK.V}$(Working Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`21`

Response identifier.

`E`$_{KEK.V}$`(Working Key)`

Field 1, the working key encrypted using the variant of the KEK specified in field one of the command. This field contains a 16 or 32 byte hexadecimal value.

`Working Key Check Digits`

Field 2, check digits; the first four digits that result from encrypting zeros using the working key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-5. Response 21: Translate Working Key for Distribution

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	21
1	$E_{KEK.V}$(Working Key)	16, 32	0 - 9, A - F
2	Working Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- This command is used for transmitting a working key from one network node to another. Both nodes must use Atalla Variant Network Security Processors and have the same KEK.
- Generate the working key to be transmitted.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a 1key-3DES (single-length) PIN Encryption Key (KPE).

- Variant (V): 1.
- Clear-text KEK: 1111 1111 1111 1111.
  The KEK encrypted under variant 0 of the MFK: 4791 B313 B61D AC09.
- Clear-text KPE: 0123 4567 89AB CDEF.
  The KPE encrypted under variant 1 of the MFK: AE86 D417 E64E 07E0.

The command looks like this:

```
<11#1#4791B313B61DAC09#AE86D417E64E07E0#>
```

The Network Security Processor returns the following response:

```
<21#C1691433AA138864#D5D4#>
```

Translating a 2key-3DES (double-length) PIN Encryption Key (KPE).

- Variant (V): 1.
- Clear-text KEK: 1111 1111 1111 1111 2222 2222 2222 2222.
  The KEK encrypted under variant 0 of the MFK: 4791 B313 B61D AC09 370B E7D9 20BF 774C.
- Clear-text KPE: 0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The KPE encrypted under variant 1 of the MFK: AE86 D417 E64E 07E0 BC62 A2AD 7251 6EA1.

The command looks like this:

```
<11#1#4791B313B61DAC09370BE7D920BF774C#
AE86D417E64E07E0BC62A2AD72516EA1#>
```

The Network Security Processor returns the following response:

```
<21#A7CD84EEB2AA0737EFD23931DC36DEFF#08D7#>
```

Translate Working Key For Local Storage (Command 13)

Command 13 translates a working key from encryption using any KEK to encryption using the MFK. Use this command to import a working key from another node that uses Atalla Variant Network Security Processors. Your node and the remote node must have the same KEK. This command translates both1key-3DES (single-length) and 2key-3DES (double-length) working keys.

This command is enabled in the Network Security Processor's default security policy.

Command

<13#Variant#$E_{MFK.0}$(KEK)#$E_{KEK.V}$(Working Key)#>

Response

<23#$E_{MFK.V}$(Working Key)#Working Key Check Digits#>[CRLF]

Calling Parameters

13

Field 0, the command identifier.

Variant

Field 1, the variant (V) of the KEK under which the working key has been encrypted, and also the variant to be applied to the MFK when generating the output cryptogram (Field 1 of the response). This field can by one or two bytes long and can contain the numbers 0 to 31. See Key variants on page 2-2 for a list of supported variants.

Variant 0 is supported if option 65 is enabled.

$E_{MFK.0}$(KEK)

Field 2, the KEK encrypted under variant 0 of the MFK. This key is used to protect the working key during a key exchange with the transmitting node. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. If field 3, contains a 2key-3DES (double-length) key, this field must also be a 2key-3DES (double-length) key, or a reference to a volatile table location that contains a 2key-3DES (double-length) key.

$E_{KEK.V}$(Working Key)

Field 3, the cryptogram of the working key sent from the remote node. It is encrypted using the variant of the KEK specified in Field 1. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

Table 3-6. Command 13: Translate Working Key for Local Storage Switch-to-Switch

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	13
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.0}$(KEK)*	16, 32	0 - 9, A - F
3	$E_{KEK.V}$(Working Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`23`

Field 0, the response identifier.

$E_{MFK.V}$`(Working Key)`

Field 1, the working key, decrypted using the variant specified when you issued the command and re-encrypted using the same variant of the MFK. This field contains a 16 or 32 byte hexadecimal value

`Working Key Check Digits`

Field 2, check digits; the first four digits that result from encrypting zeros using the working key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-7. Response 23: Translate Working Key for Local Storage Switch-to-Switch

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	23
1	$E_{MFK.V}$(Working Key)	16, 32	0 - 9, A - F
2	Working Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- This command is used to receive a working key that has been transmitted from another node that uses Atalla Variant Network Security Processors.
- Generate the Key Exchange Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a 1key-3DES (single-length) PIN Encryption Key.

- Variant (V): 1.
- Clear-text KEK: 1111 1111 1111 1111.
  The KEK encrypted under variant 0 of the MFK: 4791 B313 B61D AC09.
- Clear-text KPE: 0123456789ABCDEF.
  The KPE encrypted under variant 1 of the KEK: C169 1433 AA13 8864.

The command looks like this:

```
<13#1#4791B313B61DAC09#C1691433AA138864#>
```

The Network Security Processor returns the following response:

```
<23#AE86D417E64E07E0#D5D4#>
```

Translating a 2key-3DES (double-length) PIN Encryption Key.

- Variant (V): 1.
- Clear-text KEK: 1111 1111 1111 1111 2222 2222 2222 2222.
  The KEK encrypted under variant 0 of the MFK: 4791 B313 B61D AC09 370B E7D9 20BF 774C.
- Clear-text KPE: 0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The KPE encrypted under variant 1 of the KEK: A7CD 84EE B2AA 0737 EFD2 3931 DC36 DEFF.

The command looks like this:

```
<13#1#4791B313B61DAC09370BE7D920BF774C#
A7CD84EEB2AA0737EFD23931DC36DEFF#>
```

The Network Security Processor returns the following response:

```
<23#AE86D417E64E07E0BC62A2AD72516EA1#08D7#>
```

Load ATM Master Key – Diebold (Command 14)

Command 14 – Diebold, encrypts the ATM master key using an Encryption Key for downloading to Diebold ATMs. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<14#1#Variants#$E_{MFK.I}$(Master Key)#$E_{MFK.V}$(Encryption Key)#>

Response

<24#1#$E_{Encryption\ Key}$(Master Key)#Master Key Check Digits#>
[CRLF]

Calling Parameters

`14`

Field 0, the command identifier.

`1`

Field 1, the ATM identifier; in this command, Diebold.

`Variants`

Field 2, the MFK variants, I and V, appropriate for the input keys. The I value pertains to the master key to be downloaded; the V value pertains to the encryption key already established in the ATM.

The following types of keys and their corresponding variants can be downloaded.

Types of ATM Master Key to be Downloaded:

Key Type	Variant
MAC Master Key	1
PIN Master Key	5
VISA Master Key	5

The following keys and their corresponding variants can exist in the ATMs:

Key Type	Variant
Communications Key, KC	1
ATM A Key	5
PIN Master Key-1	5
VISA Master Key-1	5

$E_{MFK.I}$(Master Key)

Field 3, the ATM master key encrypted using the proper variant of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. This key can not be a replicated 1key-3DES (single-length) key.

$E_{MFK.V}$(Encryption Key)

Field 4, the cryptogram of the key under which the ATM master key is to be encrypted. This key is encrypted using the proper variant of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. The length of this key has to be equal or greater than the length of the Master Key and can not be a replicated 1key-3DES (single-length) key.

Table 3-8. Command 14: Load ATM Master Key – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	14
1	ATM identifier (Diebold)	1	1
2	Variants (I, V)	2	1, 5
3	$E_{MFK.I}$(Master Key)*	16, 32	0 - 9, A - F
4	$E_{MFK.V}$(Encryption Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

24

Field 0, the response identifier.

1

Field 1, the ATM identifier; in this command, Diebold.

$E_{Encryption\ Key}$(Master Key)

Field 2, the master key value encrypted using the encryption key. No variant is associated with this encryption because the ATM does not support key variants. This field contains a 16 byte hexadecimal value.

`Master Key Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the master key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-9. Response 24: Load ATM Master Key – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	24
1	ATM identifier (Diebold)	1	1
2	$E_{\text{Encryption Key}}$(Master Key)	16, 32	0 - 9, A - F
3	Master Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Perform the following tasks before using Command 14:

- Manually load the ATM with its initial keys.
- Generate the appropriate key for encrypting the ATM master key and encrypt it using the proper variant of the MFK.
- Generate the ATM master key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Loading an ATM master key.

- Variant (I): 1.
- Variant (V): 5.
- Clear-text Master Key: 3333 3333 3333 3333 5555 6666 7777 8888.
  The Master Key encrypted under variant 1 of the MFK:3219 92E9 44B0 F423 1DE1 CF68 9E96 99D6.
- Clear-text Encryption Key: 1111 1111 1111 1111 0123 4567 89AB CDEF.
  The Encryption Key encrypted under variant 5 of the MFK:118A 17BA 953B D16C 608FC3DD BDDA 3E56.

The command looks like this:

```
<14#1#15#321992E944B0F4231DE1CF689E9699D6#118A17BA953BD16C608
FC3DDBDDA3E56#>
```

The Network Security Processor returns the following response:

```
<24#1#CA652727D7ECC3FF29D072B935BEC86E#B15B#>
```

Load ATM Master Key – IBM 3624 (Command 14)

Command 14 – IBM 3624, encrypts the ATM master key for downloading to an IBM 3624 ATM. This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<14#3#$E_{MFK.5}$(KM)#$E_{MFK.1}$(K1)#$E_{MFK.2}$(K2)#Message#>

Response

<24#3#IBM 3624 Message#>[CRLF]

Calling Parameters

14

Field 0, the command identifier.

3

Field 1, the ATM identifier; in this command, IBM 3624.

$E_{MFK.5}$(KM)

Field 2, the ATM master key encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(K1)

Field 3, the ATM A key (K1) encrypted under variant 1 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.2}$(K2)

Field 4, the ATM communications key (K2) encrypted under variant 2 of the MFK. This field contains a 16-byte hexadecimal value, or a volatile table location.

Message

Field 5, bytes five to eight in the IBM 3624 request message, represented as eight hexadecimal characters.

Table 3-10. Command 14: Load ATM Master Key – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Command Identifier	2	14
1	ATM identifier (IBM 3624)	1	3
2	$E_{MFK.5}$(KM)*	16	0-9, A-F
3	$E_{MFK.1}$(K1)*	16	0 - 9, A - F
4	$E_{MFK.2}$(K2)*	16	0 - 9, A - F
5	Message	8	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`24`

Field 0, the response identifier.

`3`

Field 1, the ATM identifier; in this command, IBM 3624.

`IBM 3624 Message`

Field 2, the result of the partial double-encryption process defined in IBM key management. This result is formed using the following steps.

1. First, the ATM master key is encrypted using the ATM A key. This is denoted as E_{K1}(KM), where K1 is the ATM A key and KM is the ATM master key to be downloaded.
2. Let L4 represent the leftmost 4 bytes of E_{K1}(KM) and let R4 represent the rightmost 4 bytes of E_{K1}(KM). Each value – L4 and R4 – is 8 hexadecimal characters long.

   The four variable bytes in Field 5 of the command are concatenated to the left of L4, forming an eight byte (that is, 16 hexadecimal character) field, denoted as follows.

   `[(4 Var Bytes) || L4]`

   This field is then encrypted using the ATM communication key, K2. The result of this encryption is denoted as follows.

   E_{K2}`[(4 Var Bytes) || L4]`

3. R4 is concatenated to the right of the encrypted result of step two, denoted as follows.

   E_{K2}[(4 Var Bytes) || L4] || R4

   This result is the 12 byte (that is, 24 hexadecimal character) field that is sent to the IBM 3624 ATM.

Table 3-11. Response 24: Load ATM Master Key – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	24
1	ATM identifier (IBM 3624)	1	3
2	IBM 3624 Message	24	0 - 9, A - F

Usage Notes

Before using Command 14, generate the ATM Master Key (MK), ATM A key (K1) and the ATM communications key (K2).

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Loading an ATM master key.

- Clear-text Master Key: 1111 1111 1111 1111.
  The Master Key encrypted under variant 5 of the MFK: 118A 17BA 953B D16C.
- Clear-text ATM A Key (K1): 2222 2222 2222 2222.
  The ATM A Key (K1) encrypted under variant 1 of the MFK: C880 88CB 8FE8 46FE.
- Clear-text ATM Communications Key (K2): 3333 3333 3333 3333.
  The ATM Communications Key (K2) encrypted under variant 2 of the MFK: C22F 5A1F 22D1 ABF1.
- Message: 56789ABC.

The command looks like this:

```
<14#3#118A17BA953BD16C#C88088CB8FE846FE#C22F5A1F22D1ABF1#
56789ABC#>
```

The Network Security Processor returns the following response:

```
<24#3#2B41AE49E5C8E28F811DA672#>
```

Load ATM Master Key – IBM 4731 (Command 14)

Command 14 – IBM 4731, generates an IBM 4731 ATM master key (KM). This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<14#4#$E_{MFK.0}$(Exchange Key)#$E_{MFK.0}$(Initial Master Key)#
Message#[Key Length#]>

Response

<24#4#$E_{MFK.1}$(ATM Master Key)#
$E_{MFK.2}$($E_{Initial\ Master\ Key}$(ATM Master Key))#
$E_{Exchange\ Key}$($E_{Initial\ Master\ Key}$(ATM Master Key))#
E($E_{Initial\ Master\ Key}$(ATM Master Key))(message type/date)#
Exchange Key Check Digits#Initial Master Key Check Digits#
ATM Master Key Check Digits#
ATM Master Key encrypted under Initial Master Key Check
Digits#>

Calling Parameters

14

Field 0, the command identifier.

4

Field 1, the ATM identifier; in this command, IBM 4731.

$E_{MFK.0}$(Exchange Key)

Field 2, the Exchange Key encrypted under variant 0 of the MFK. This key is used to encrypt the cryptogram of the generated ATM master key encrypted under the Initial Master Key. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. The length of this key has to be equal or greater than the length of the Initial Master Key. If the Key Length field contains a value of 2 (generate 2key-3DES working key), the KEK has to be 2key-3DES (double-length) and can not be a replicated 1key-3DES (single-length) key.

$E_{MFK.0}$(Initial Master Key)

Field 3, the Initial Master Key encrypted under variant 0 of the MFK. This key is used to encrypt the generated ATM Master Key. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. The length of this key has to be equal or greater than the length of the ATM Master Key. If the Key Length field contains a value of 2 (generate 2key-3DES working key), the Master Key has to be 2key-3DES (double-length) and can not be a replicated 1key-3DES (single-length) key.

Message

Field 4, the message type/date in binary form. This field contains an 8 byte binary value, where each character represents one byte.

[Key Length#]

Field 5, length of the generated IBM 4731 ATM master key. This is an optional field. If used, it can be one byte long and can be empty, or contain the numbers 1 (to generate 1key-3DES key) or 2 (to generate 2key-3DES key). If this field is not present in the command, the default 1key-3DES key will be generated.

Table 3-12. Command 14: Load ATM Master Key – IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	14
1	ATM identifier (IBM 4731)	1	4
2	$E_{MFK.0}$(Exchange Key)*	16, 32	0 - 9, A - F
3	$E_{MFK.0}$(Initial Master Key)*	16, 32	0 - 9, A - F
4	Message	8	Binary
5	[Key Length]	0,1	empty, 1-2

*Can be a volatile table location.

Responding Parameters

24

Field 0, the response identifier.

4

Field 1, the ATM identifier; in this command, IBM 4731.

$E_{MFK.1}$(ATM Master Key)

Field 2, the generated ATM Master Key encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value.

$E_{MFK.2}(E_{Initial\ Master\ Key}(ATM\ Master\ Key))$

Field 3, the generated ATM Master Key encrypted under the Initial Master Key. This cryptogram is then encrypted under variant 2 of the MFK. This field contains a 16 or 32 byte hexadecimal value.

$E_{Exchange\ Key}(E_{Initial\ Master\ Key}(ATM\ Master\ Key))$

Field 4, the generated ATM Master Key encrypted under the Initial Master Key.This cryptogram is then encrypted under the Exchange Key. This field contains a 16 or 32 byte hexadecimal value.

$E(E_{Initial\ Master\ Key}(ATM\ Master\ Key))(message\ type/date)$

Field 5, the generated ATM Master Key encrypted under the Initial Master Key. This cryptogram is then used to encrypt the message type/date. This field contains a 16 byte hexadecimal value.

Exchange Key Check Digits

Field 6, check digits; the first four digits that result from encrypting zeros using the Exchange Key. If option 88 is enabled, this field will contain the first six digits of the result.

Initial Master Key Check Digits

Field 7, check digits; the first four digits that result from encrypting zeros using the Initial Master Key. If option 88 is enabled, this field will contain the first six digits of the result.

Master Key Check Digits

Field 8, check digits; the first four digits that result from encrypting zeros using ATM Master Key. If option 88 is enabled, this field will contain the first six digits of the result.

Cryptogram Check Digits

Field 9, check digits; the first four digits that result from encrypting zeros using the cryptogram of the ATM Master Key encrypted under the Initial Master Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-13. Response 24: Load ATM Master Key – IBM 4731 (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	24
1	ATM identifier (IBM 4731)	1	4
2	$E_{MFK.1}$(Master Key)	16, 32	0 - 9, A - F
3	$E_{MFK.2}(E_{Initial\ Master\ Key}$(Master Key))	16, 32	0 - 9, A - F

Table 3-13. Response 24: Load ATM Master Key – IBM 4731 (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	$E_{Exchange\ Key}(E_{Initial\ Master\ Key}$(ATM Master Key))	16, 32	0 - 9, A - F
5	E($E_{Initial\ Master\ Key}$(ATM Master Key)) (message type/date)	16	0 - 9, A - F
6	Exchange Key Check Digits	4 or 6	0 - 9, A - F
7	Initial Master Key Check Digits	4 or 6	0 - 9, A - F
8	ATM Master Key Check Digits	4 or 6	0 - 9, A - F
9	Cryptogram Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 14, generate the Initial Master Key (KI) and the KEK (KX).

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

This example generates a random ATM Master Key your result will be different.

Generating an IBM 4731 ATM Master Key.

- Clear-text Exchange Key (KX): 0123 4567 89AB CDEF FEDC BA98 7654 3210. The Exchange Key encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E 0B176C3EBEED18AF.
- Clear-text Initial Master key (KI): 1111 2222 3333 4444 0123 4567 89AB CDEF. The Initial Master Key encrypted under variant 0 of the MFK: 45ED 2536 2B16 0750 9007 B875 1BB7 AB4E.
- Message: 01234567

The command looks like this:

```
<14#4#9007B8751BB7AB4E0B176C3EBEED18AF#45ED25362B1607509007B8
751BB7AB4E#01234567#1#>
```

The Network Security Processor returns the following response:

```
<24#4#C74B7C95BACD75BC#E2DE3E4599DE60F3#0466F2C849DDA497#
06E639A3F8267CBD#08D7#0389#9024#D8F0#>
```

Change ATM Communications Key – Diebold (Command 15)

Command 15 – Diebold, encrypts a communications key for downloading to a Diebold ATM. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<15#1#Variant#$E_{MFK.1}$(Communications Key)#$E_{MFK.V}$(P)#>

Response

<25#1#E_P(KC)#Communications Key Check Digits#>[CRLF]

Calling Parameters

15

Field 0, the command identifier.

1

Field 1, the ATM identifier; in this command, Diebold.

Variant

Field 2, the MFK variant, V, (1 or 5) under which the encrypting key was encrypted. This field contains a 1 byte decimal value which can be either 1 or 5.

$E_{MFK.1}$(KC)

Field 3, the new Communications Key encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location and can not be a replicated 1key-3DES (single-length) key.

$E_{MFK.V}$(P)

Field 4, the encryption key encrypted under variant 1 or 5 of the MFK. This key is used to encrypt the new communications key. If the variant specified in Field 2 is 1, then P represents the old communications key. If the variant specified in Field 2 is 5, then P represents the PIN master key. The length of this key has to be equal or greater than the length of the Communication Key, and can not be a replicated 1key-3DES (single-length) key.

Table 3-14. Command 15: Change ATM Communications Key – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	15
1	ATM identifier	1	1
2	Variant (V)	1	1, 5
3	$E_{MFK.V}(KC)$*	16, 32	0 - 9, A - F
4	$E_{MFK.V}(P)$*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`25`

Field 0, the response identifier.

`1`

Field 1, the ATM identifier; in this command, Diebold.

$E_P(KC)$

Field 2, the new Communications Key encrypted using the encryption key for downloading.

`Communications Key Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the communications key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-15. Response 25: Change ATM Communications Key – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	25
1	ATM identifier (Diebold)	1	1
2	$E_P(KC)$	16, 32	0 - 9, A - F
3	Communications Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- The communications key in a Diebold ATM is used to encrypt PINs; therefore, although the term communications key is used for this key, it is supported in the Atalla key management scheme as a PIN encryption key.
- Generate the encryption key which will be used to encrypt the new Communications Key. That is, either the previous communications key or the PIN master key.
- Generate the new Communications Key (KC). The communications key can be randomly generated using Command 10.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Changing an ATM Communications Key.

- Variant: 1.
- Clear-text new Communications Key: 0123456789ABCDEF 1111222233334444. The new Communications Key encrypted under variant 1 of the MFK: AE86D417E64E07E0 D538A881DE91EAF1.
- Clear-text Encryption Key (P):3333 3333 3333 3333 5555 6666 7777 8888. The Encryption Key encrypted under variant 1 of the MFK: 321992E944B0F423 1DE1CF689E9699D6.

The command looks like this:

```
<15#1#1#AE86D417E64E07E0D538A881DE91EAF1#321992E944B0F4231DE1
CF689E9699D6#>
```

The Network Security Processor returns the following response:

```
<25#1#4DEB22EE1652AA8A216FF8BA794E8AFD#4E15#>
```

Change ATM Communications Key – Docutel (Command 15)

Command 15 – Docutel, encrypts a Communications Key for downloading to a Docutel ATM. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<15#2#$E_{MFK.1}$(Communications Key)#$E_{MFK.5}$(ATM Master Key)#>

Response

<25#2#$E_{Master\ Key}$(Communications Key)#
Communications Key Check Digits#>[CRLF]

Calling Parameters

15

Field 0, the command identifier.

2

Field 1, the ATM identifier; in this command, Docutel.

$E_{MFK.1}$(KC)

Field 2, the Communications Key encrypted under variant 1 of the MFK. This is the key to be downloaded to the ATM. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location and can not be a replicated 1key-3DES (single-length) key.

$E_{MFK.5}$(KM)

Field 3, the ATM Master Key encrypted under variant 5 of the MFK. This key is used to encrypt the new Communications Key. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location and can not be a replicated 1key-3DES (single-length) key. If field 2 contains a 2key-3DES (double-length) key this field must also contain a 2key-3DES key.

Table 3-16. Command 15: Change ATM Communications Key – Docutel

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	15
1	ATM identifier (Docutel)	1	2
2	$E_{MFK.1}$(Communications Key)*	16 or 32	0 - 9, A - F
3	$E_{MFK.5}$(ATM Master Key)*	16 or 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`25`

Field 0, the response identifier.

`2`

Field 1, the ATM identifier; in this command, Docutel.

$E_{\texttt{Master Key}}$`(Communications Key)`

Field 2, the Communications Key encrypted under the ATM master key. This field contains a 16 byte hexadecimal value.

`Communications Key Check Digits`

Field 3, the result of encrypting 0123 4567 89AB CDEF using the Communications Key. This field contains a 16 byte hexadecimal value.

Table 3-17. Response 25: Change ATM Communications Key – Docutel

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	25
1	ATM identifier (Docutel)	1	2
2	E_{KM}(KC)	16 or 32	0 - 9, A - F
3	E_{KC}(01234...EF)	16	0 - 9, A - F

Usage Notes

- The communications key on a Docutel ATM is used to encrypt PINs; therefore, although the term communications key is used for this key, it is supported in the Atalla key management scheme as a PIN encryption key (KPE).
- Generate the ATM master key.
- Generate the new communications key (KC). The communications key can be randomly generated using Command 10.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Changing the primary node's ATM communications key.

- Clear-text Communications Key: 1111 1111 1111 1111.
  The Communications Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.
- Clear-text ATM Master Key: 2222 2222 2222 2222.
  The ATM Master Key encrypted under variant 5 of the MFK: EA45 F59C 6242 F687.

The command looks like this:

```
<15#2#C6283830AE9E875A#EA45F59C6242F687#>
```

The Network Security Processor returns the following response:

```
<25#2#08024FCF811DA672#8A5AE1F81AB8F2DD#>
```

Change ATM Communications Key – IBM 3624 (Command 15)

Command 15 – IBM 3624, encrypts a communications key for downloading to an IBM 3624 ATM. This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<15#3#$E_{MFK.2}$(Communications Key)#$E_{MFK.I}$(P)#
$E_{MFK.2}$(Communications Key-1)#Message#Variant#>

Response

<25#3#IBM 3624 Message#>[CRLF]

Calling Parameters

15

Field 0, the command identifier.

3

Field 1, the ATM identifier; in this command, IBM 3624.

$E_{MFK.2}$(Communications Key)

Field 2, the Communications Key encrypted under variant 2 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.V}$(P)

Field 3, either the ATM Master Key (KM) encrypted under variant 1 of the MFK, or the old communications key (KC-1) encrypted under variant 2 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location. The contents of this field depend on the value of Field 6.

$E_{MFK.2}$(Communications Key-1)

Field 4, the old Communications Key (KC-1) encrypted under variant 2 of the MFK. This field contains a 16-byte hexadecimal value, or a volatile table location.

Message

Field 5, bytes five to eight of the IBM 3624 request message, represented as eight hexadecimal characters.

`Variant`

Field 6, the variant that applies to Field 3. This field contains a 1 byte decimal value which can be either 1 or 2.

Table 3-18. Command 15: Change ATM Communications Key – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	15
1	ATM identifier (IBM 3624)	1	3
2	$E_{MFK.2}$(Communications Key)*	16	0 - 9, A - F
3	$E_{MFK.V}$(P)*	16	0 - 9, A - F
4	$E_{MFK.2}$(old Communications Key)*	16	0 - 9, A - F
5	Message	8	0 - 9, A - F
6	Variant (V)	1	1, 2

*Can be a volatile table location.

Responding Parameters

`25`

Field 0, the response identifier.

`3`

Field 1, the ATM identifier; in this command, IBM 3624.

`IBM 3624 Message`

Field 2, the result of the partial double encryption process defined in IBM key management. This is formed using the following steps.

1. First, the communications key, KC, is encrypted using the appropriate key, P. The result, E_P(KC), is divided into two parts, L4 and R4.
2. The four variable bytes of Field 5 in the command are concatenated to the left of L4, denoted as follows.

   `(4 Var Bytes) || L4`

   The result is then encrypted using the old communications key, KC-1, denoted as follows.

   E_{KC-1}`[(4 Var Bytes) || L4]`
3. R4 is then concatenated to the right of this encrypted result to obtain KC-1, denoted as follows.

$E_{KC-1}[(4\ Var\ Bytes)\ ||\ L4]\ ||\ R4$

This result is the 12 byte (that is, 24 hexadecimal character) field that is sent to the 3624 ATM.

Table 3-19. Response 15: Change ATM Communications Key – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	25
1	ATM identifier (IBM 3624)	1	3
2	IBM 3624 message	24	0 - 9, A - F

Usage Notes

- The communications key on an IBM 3624 ATM is used to encrypt data; therefore it is encrypted under variant 2 of the MFK.
- Generate the encryption key to be downloaded.
- Generate both the old and new communications keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Changing the primary node's ATM communications key.

- Clear-text Communications Key (KC): 0123 4567 89AB CDEF.
  The Communications Key encrypted under variant 2 of the MFK: 80BC DEAC 5703 BC84.
- Clear-text ATM Master Key: 3333 3333 3333 3333.
  The ATM Master Key encrypted under variant 1 of the MFK:3219 92E9 44B0 F423.
- Clear-text old Communications Key (KC-1): 0123456789ABCDEF.
  The old Communications Key (KC-1) encrypted under variant 2 of the MFK: 80BC DEAC 5703 BC84.
- IBM 3624 request message: 12345678.
- Variant: 1.

The command looks like this:

```
<15#3#80BCDEAC5703BC84#321992E944B0F423#80BCDEAC5703BC84#
12345678#1#>
```

The Network Security Processor returns the following response:

```
<25#3#11581BCF707F368E06463E6C#>
```

Change ATM Communications Key – IBM 4731 (Command 15)

Command 15 – IBM 4731, generates a random communication key (KC) for downloading to an IBM 4731 ATM. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<15#4#$E_{MFK.0}$(Exchange Key)#Message#[Key Length]#>

Response

<25#4#$E_{MFK.3}$(Communications Key)#
$E_{Exchange\ Key}$(Communications Key)#$E_{Communications\ Key}$(message)#
Communications Key Check Digits#>

Calling Parameters

15

Field 0, the command identifier.

4

Field 1, the ATM identifier; in this command, IBM 4731.

$E_{MFK.0}$(Exchange Key)

Field 2, the Exchange Key encrypted under variant 0 of the MFK. This key is used to encrypt the generated Communications Key. This field contains a 16 or 32 byte hexadecimal value. The length of this Exchange Key has to be equal or greater than the length of the Communications Key. If the Key Length field contains a value of 2 (generate 2key-3DES working key), the Exchange Key has to be 2key-3DES (double-length) and can not be a replicated 1key-3DES (single-length) key.

Message

Field 3, the message type/date in binary form. This field contains an 8 byte binary value.

[Key Length]

Field 4, length of the generated IBM 4731 ATM master key. This is an optional field. If used, it can be one byte long and can be empty, or contain the number 1 (to

generate 1key-3DES key) or 2 (to generate 2key-3DES key). If this field is not present in the command, the default 1key-3DES key will be generated.

Table 3-20. Command 15: Change ATM Communications Key – IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	15
1	ATM identifier (IBM 4731)	1	4
2	$E_{MFK.0}$(Exchange Key)*	16, 32	0 - 9, A - F
3	Message	8	any (binary)
4	[Key Length]	0, 1	empty, 1-2

*Can be a volatile table location.

Responding Parameters

`25`

Field 0, the response identifier.

`4`

Field 1, the ATM identifier; in this command, IBM 4731.

$E_{MFK.3}$`(KC)`

Field 2, the generated Communications Key (KC) encrypted under variant 3 of the MFK. This field contains a 16 or 32 byte hexadecimal value.

E_{KX}`(KC)`

Field 3, the generated Communications Key (KC) encrypted under the Exchange Key. This field contains a 16 or 32 byte hexadecimal value.

E_{KC}`(message)`

Field 4, the message type/date encrypted under the generated Communications Key (KC). This field contains a 16 byte hexadecimal value.

`Communications Key Check Digits`

Field 5, check digits; the first four digits that result from encrypting zeros using the Communications Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-21. Response 25: Change ATM Communications Key – IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	25
1	ATM identifier (IBM 4731)	1	4
2	$E_{MFK.3}$(KC)	16, 32	0 - 9, A - F
3	$E_{Exchange\ Key}$(Communications Key)	16, 32	0 - 9, A - F
4	$E_{Communications\ Key}$(Message)	16	0 - 9, A - F
5	Communications Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using the command, generate the Exchange Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

This command generates a random communications key, your test results will be different.

Changing the primary node's ATM communications key.

- Clear-text Exchange Key (KX): 0123 4567 89AB CDEF 1111 2222 3333 4444. The Exchange Key (KX) encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E 45ED 2536 2B16 0750.
- Message: 01234567

The command looks like this:

```
<15#4#9007B8751BB7AB4E45ED25362B160750#01234567#1#>
```

The Network Security Processor returns the following response:

```
<25#4#9986C1DB8ABE561D#CE7A35D9A2787D86#60EF4DE29208C532#
E6C0#>
```

Encrypt Financial Institution Table – Diebold (Command 16)

Command 16 – Diebold, encrypts keys to be downloaded to Diebold ATMs' financial institution tables (FITs). This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<16#1#Variant#$E_{MFK.V}$(P)#$E_{MFK.5}$(Q)#>

Response

<26#1#E_{Q}(P)#Check Digits of P#>[CRLF]

Calling Parameters

16

Field 0, the command identifier.

1

Field 1, the ATM identifier; in this command, Diebold.

Variant

Field 2, the variant (V) to be applied to the MFK when encrypting the key to be downloaded. This field contains a 1 byte decimal value which can be either 2 or 5.

$E_{MFK.V}$(P)

Field 3, the key to be downloaded to the ATM, encrypted using the variant of the MFK specified in Field 2. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.5}$(Q)

Field 4, the key used to encrypt the Financial Institution Table entry. This key is encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 3-22. Command 16: Encrypt Financial Institution Table – Diebold:

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	16
1	ATM identifier (Diebold)	1	1
2	Variant (V)	1	2, 5
3	$E_{MFK.V}(P)$*	16	0 - 9, A - F
4	$E_{MFK.5}(Q)$*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`26`

Field 0, the response identifier.

`1`

Field 1, the ATM identifier; in this case, Diebold.

$E_Q(P)$

Field 2, the FIT key to be downloaded, encrypted using either the PIN master key (PMK) or the VISA master key (VMK). This field contains a 16 byte hexadecimal value.

`Check Digits of P`

Field 3, check digits; the first four digits that result from encrypting zeros using the encrypting key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-23. Response 26: Encrypt Financial Institution Table – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	26
1	ATM identifier (Diebold)	1	1
2	$E_Q(P)$	16	0 - 9, A - F
3	Check Digits of P	4 or 6	0 - 9, A - F

Usage Notes

- Each execution of this command encrypts a single key for a FIT entry.
- Generate the encrypting key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Changing the primary node's ATM communications key.

- Clear-text FIT Key (P): 3333 3333 3333 3333.
  The FIT Key encrypted under variant 2 of the MFK: C22F 5A1F 22D1 ABF1.
- Clear-text Encrypting Key (Q): 1111 1111 1111 1111.
  The Encrypting Key encrypted under variant 5 of the MFK:118A 17BA 953B D16C.

The command looks like this:

```
<16#1#2#C22F5A1F22D1ABF1#118A17BA953BD16C#>
```

The Network Security Processor returns the following response:

```
<26#1#F679786E2411E3DE#ADC6#>
```

Encrypt Financial Institution Table – Docutel (Command 16)

Command 16 – Docutel, encrypts keys to be downloaded to Docutel ATM's financial institution tables (FITs). This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<16#2#$E_{MFK.5}$(PIN Verification Key)#$E_{MFK.5}$(ATM Master Key)#>

Response

<26#2#$E_{ATM\ Master\ Key}$(PIN Verification Key)#
PIN Verification Key Check Digits#>[CRLF]

Calling Parameters

16

Field 0, the command identifier.

2

Field 1, the ATM identifier; in this command, Docutel.

$E_{MFK.5}$(PIN Verification Key)

Field 2, the PIN Verification Key (KPV) encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.5}$(ATM Master Key)

Field 3, the ATM Master Key encrypted under variant 5 of the MFK. This key is used to encrypt the PIN Verification Key. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 3-24. Command 16: Encrypt Financial Institution Table – Docutel

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	16
1	ATM identifier (Docutel)	1	2
2	$E_{MFK.5}$(PIN Verification Key)*	16	0 - 9, A - F
3	$E_{MFK.5}$(ATM Master Key)*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`26`

Field 0, the response identifier.

`2`

Field 1, the ATM identifier; in this command, Docutel.

E_{KM}`(KP)`

Field 2, the PIN Verification Key encrypted under the ATM Master Key. This field contains a 16 byte hexadecimal value.

`PIN Verification Key Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the PIN Verification Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-25. Response 26: Encrypt Financial Institution Table – Docutel

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	26
1	ATM identifier (Docutel)	1	2
2	$E_{\text{ATM Master Key}}$(PIN Verification Key)	16	0 - 9, A - F
3	PIN Verification Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 16, generate the PIN Verification Key (KPV) and the ATM Master Key (KM).

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Changing the primary node's ATM communications key.

- Clear-text PIN Verification Key: 2222 2222 2222 2222.
  The PIN Verification Key encrypted under variant 5 of the MFK: EA45 F59C 6242 F687.
- Clear-text ATM Master Key: 1111 1111 1111 1111.
  The ATM Master Key encrypted under variant 5 of the MFK: 118A 17BA 953B D16C.

The command looks like this:

```
<16#2#EA45F59C6242F687#118A17BA953BD16C#>
```

The Network Security Processor returns the following response:

```
<26#2#950973182317F80B#0096#>
```

Encrypt Financial Institution Table – IBM 3624 (Command 16)

Command 16 – IBM 3624 encrypts keys to be downloaded to IBM 3624 ATMs' financial institution tables (FITs). This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<16#3#$E_{MFK.5}$(PIN Verification Key)#$E_{MFK.1}$(ATM Master Key)#>

Response

<26#3#$E_{ATM\ Master\ Key}$(PIN Verificatin Key)#
PIN Verification Key Check Digits#>[CRLF]

Calling Parameters

16

Field 0, the command identifier.

3

Field 1, the ATM identifier; in this command, IBM 3624.

$E_{MFK.5}$(KPV)

Field 2, the PIN Verification Key encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(ATM Master Key)

Field 3, the ATM Master Key encrypted under variant 1 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 3-26. Command 16: Encrypt Financial Institution Table – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	16
1	ATM identifier (IBM 3624)	1	3
2	$E_{MFK.5}$(PIN Verification Key)*	16	0 - 9, A - F
3	$E_{MFK.1}$(ATM Master Key)*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`26`

Field 0, the response identifier.

`1`

Field 1, the ATM identifier; in this command, IBM 3624.

E_{KM}`(KP)`

Field 2, the PIN Verification Key encrypted under the ATM Master Key. This field contains a 16 byte hexadecimal value.

`PIN Verification Key Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the PIN Verification Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-27. Response 26: Encrypt Financial Institution Table – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	26
1	ATM identifier (IBM 3624)	1	3
2	$E_{\text{ATM Master Key}}$(PIN Verification Key)	16	0 - 9, A - F
3	PIN Verification Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 16, generate the PIN Verification Key (KPV) and the ATM Master Key (KM).

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Changing the primary node's ATM communications key.

- Clear-text PIN Verification Key: 2222 2222 2222 2222.
  The PIN Verification Key encrypted under variant 5 of the MFK: EA45 F59C 6242 F687.
- Clear-text ATM Master Key: 1111 1111 1111 1111.
  The ATM Master Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.

The command looks like this:

```
<16#3#EA45F59C6242F687#C6283830AE9E875A#>
```

The Network Security Processor returns the following response:

```
<26#3#950973182317F80B#0096#>
```

Generate VISA Working Key (Command 18)

Command 18 generates an odd parity 1key-3DES (**single-length**) acquirer or issuer PIN Encryption Key – for use with VISA security processors. The Network Security Processor generates two cryptograms: one for local storage and one to transmit to another network node.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<18#$E_{MFK.0}$(Key Exchange Key)#>

Response

<28#$E_{KEK.0}$(VISA Working Key)#$E_{MFK.1}$(VISA Working Key)#
VISA Working Key Check Digits#>[CRLF]

Calling Parameters

18

Field 0, the command identifier.

$E_{MFK.0}$(Key Exchange Key)

Field 1, the Key Exchange Key encrypted under variant 0 of the MFK. Visa refers to this key as a Zone Control Master Key (ZCMK). This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

Table 3-28. Command 18: Generate VISA Working Key

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	18
1	$E_{MFK.0}$(Key Exchange Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

28

Field 0, the response identifier.

E_{KEK}(VISA Working Key)

Field 1, the VISA working key encrypted under the KEK. The host application transmits this value to the VISA network switch. This field contains a 16 byte hexadecimal value.

$E_{MFK.1}$(VISA Working Key)

Field 2, the VISA working key encrypted using variant 1 of the MFK. The host application stores this cryptogram on its local data base for subsequent use. This field contains a 16 byte hexadecimal value.

VISA Working Key Check Digits

Field 3, check digits; that is, the first six digits that result from encrypting zeros using the VISA working key. This field contains a six byte hexadecimal value.

Table 3-29. Response 28: Generate VISA Working Key

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	28
1	$E_{KEK.0}$(VISA Working Key)	16	0 - 9, A - F
2	$E_{MFK.1}$(VISA Working Key)	16	0 - 9, A - F
3	VISA Working Key Check Digits	6	0 - 9, A - F

Usage Notes

- Generate a KEK (VISA refers to this as a Zone Control Master Key or ZCMK).

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

This command generates a random value therefore your results will be different.

Generating a VISA working key.

- Clear-text Key Exchange Key: 0123 4567 89AB CDEF.
  The Key Exchange Key encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E.

The command looks like this:

```
<18#9007B8751BB7AB4E#>
```

The Network Security Processor returns the following response:

```
<28#EA3310FF19DB4F4C#6CE476EF7B6E4776#7DE170#>
```

Translate Communications Key for Local Storage (Command 19)

Command 19 translates a working key from base key encryption (without a variant) to MFK encryption for local storage and subsequent use. This command is used to import a working key from a network that does not use Atalla variants. This command supports both 1key-3DES (single-length) and 2key-3DES (double-length) working keys.

This command is not enabled in the Network Security Processor's default factory security policy.

This command has a high security exposure if support for variant zero is enabled. The Network Security Processor's security policy must enable option 65 to allow variant zero to be used in this command. Option 65 must be purchased and enabled with a command 105, then added to the Network Security Processor's security policy.

Command

<19#Variant#$E_{MFK.V}$(Base Key)#$E_{Base\ Key}$(Working Key)#>

Response

<29#$E_{MFK.V}$(Working Key)#Working Key Check Digits#>[CRLF]

Calling Parameters

19

Field 0, the command identifier.

Variant

Field 1, the MFK variant under which the working key will be encrypted. This field contains a 1 or 2 byte decimal value in the range of 0 to 31. See Key variants on page 2-2 for a list of supported variants.

$E_{MFK.V}$(Base Key)

Field 2, the base key encrypted under the variant of the MFK specified in field 1. The base key is used to encrypt the working key. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{Base\ Key}$(Working Key)

Field 3, the working key encrypted under the base key. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

Table 3-30. Command 19: Translate Communications Key for Local Storage

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	19
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.V}$(Base Key)*	16, 32	0 - 9, A - F
3	$E_{Base\ Key}$(Working Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`29`

Field 0, the response identifier.

$E_{MFK.V}$`(Working Key)`

Field 1, the cryptogram of the translated key.

`Working Key Check Digits`

Field 2, check digits; the first four digits that result from encrypting zeros using the working key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-31. Response 29: Translate Communications Key for Local Storage

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	29
1	$E_{MFK.V}$(Working Key)	16, 32	0 - 9, A - F
2	Working Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Option 65 must be enabled in the Network Security Processor's security policy to use variant 0.
- This command is typically used to receive a working key transmitted from a non-Atalla node.
- Encrypt the Base Key under the appropriate variant of the MFK.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN Encryption Key for local storage.

- Variant: 1.
- Clear-text Base Key: 0123 4567 89AB CDEF.
  The Base Key encrypted under variant 1 of the MFK: AE86 D417 E64E 07E0.
- Clear-text PIN Encryption Key: FEDC BA98 7654 3210.
  The PIN encryption key encrypted under the Base Key: 12C6 26AF 058B 433B.

The command looks like this:

```
<19#1#AE86D417E64E07E0#12C626AF058B433B#>
```

The Network Security Processor returns the following response:

```
<29#BC62A2AD72516EA1#A68C#>
```

Translate Working Key for Distribution to Non-Atalla Node (Command 1A)

Command 1A translates a working key from encryption under a specified variant of the MFK, to KEK encryption without a variant for distributing to a non-Atalla network. This command supports both 1key-3DES (single-length) and 2key-3DES (double-length) keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<1A#Variant#$E_{MFK.0}$(Key Exchange Key)#$E_{MFK.V}$(Working Key)#>

Response

<2A#$E_{Key\ Exchange\ Key.0}$(Working Key)#Working Key Check Digits#>
[CRLF]

Calling Parameters

1A

Field 0, the command identifier.

Variant

Field 1, the MFK variant under which the working key has been encrypted. This field contains a 1 or 2 byte decimal value which can be in the range of 0 to 31. See Key variants on page 2-2 for a list of supported variants.

$E_{MFK.0}$(KEK)

Field 2, the Key Exchange Key (KEK) encrypted under variant 0 of the MFK. This key will be used to encrypt the working key for transmission to the non-Atalla network. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.V}$(Working Key)

Field 3, the working key encrypted using the variant of the MFK specified in Field 1. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

Table 3-32. Command 1A: Translate Working Key for Distribution to Non-Atalla Node

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	1A
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.0}$(Key Exchange Key)*	16, 32	0 - 9, A - F
3	$E_{MFK.V}$(Working Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`2A`

Field 0, the response identifier.

$E_{\texttt{Key Exchange Key}}$`(Working Key)`

Field 1, the working key encrypted under the KEK. No variant is applied to the KEK. This field contains a 16 or 32 byte hexadecimal value.

`Working Key Check Digits`

Field 2, check digits; that is, the first six digits that result from encrypting zeros using the working key. This field contains a six byte hexadecimal value.

Table 3-33. Response 2A: Translate Working Key for Distribution to Non-Atalla Node

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	2A
1	$E_{KEK.0}$(Working Key)	16, 32	0 - 9, A - F
2	Working Key Check Digits	6	0 - 9, A - F

Usage Notes

- This command is used for distributing a working key to a non-Atalla network.
- Generate the working key cryptograms.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN Encryption Key for distributing to a non-Atalla node.

- Variant: 1.
- Clear-text Key Exchange Key (KEK): 0123 4567 89AB CDEF.
  The Key Exchange Key encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E.
- Clear-text PIN Encryption Key (KPE): 0123 4567 89AB CDEF.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: AE86 D417 E64E 07E0.

The command looks like this:

```
<1A#1#9007B8751BB7AB4E#AE86D417E64E07E0#>
```

The Network Security Processor returns the following response:

```
<2A#56CC09E7CFDC4CEF#D5D44F#>
```

Translate Communications Key for Local Storage Using a Specific Variant (Command 1D)

Command 1D translates a working key from encryption using a base key without a variant to encryption using the MFK. This command is restricted to importing working keys that use variants 1, 2 or 3. This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<1D#Variant#$E_{MFK.2}$(Base Key)#$E_{Base\ Key}$(Working Key)#>

Response

<2D#$E_{MFK.V}$(Working Key)#Working Key Check Digits#>[CRLF]

Calling Parameters

`1D`

Field 0, the command identifier.

`Variant`

Field 1, the variant (V) of the MFK under which the working key will be encrypted. This field contains a 1 byte decimal value with a range of 1 - 3. See Key variants on page 2-2 for a list of supported variants.

$E_{MFK.2}$(Base Key)

Field 2, the Base Key encrypted under variant 2 of the MFK. This key is used by the transmitting node to encrypt the working key. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{Base\ Key}$(Working Key)

Field 3, the Working Key encrypted under the Base Key. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 3-34. Command 1D: Translate Communications Key for Local Storage Using Specific Variant

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	1D
1	Variant (V)	1	1, 2 or 3
2	$E_{MFK.2(}$Base Key)*	16	0 - 9, A - F
3	$E_{Base\ Key}$(Working Key)*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`2D`

Field 0, the response identifier.

$E_{MFK.V}$`(Working Key)`

Field 1, the Working Key encrypted under the MFK using the variant specified in field 1 of the command. This field contains a 16 byte hexadecimal value.

`Working Key Check Digits`

Field 2, the check digits; the first four digits that result from encrypting zeros using the working key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-35. Response 2D: Translate Communications Key for Local Storage Using Specific Variant

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	2D
1	$E_{MFK.V}$(Working Key)	16	0 - 9, A - F
2	Working Key Check Digits	4 or 6	0 - 9, A - F

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN Encryption Key

- Variant: 1.
- Clear-text Base Key: 1010 2020 4040 8080.
  The Base Key encrypted under variant 2 of the MFK: 1693 C76D 5493 D733.
- Clear-text PIN Encryption Key: E0E0 D0D0 B0B0 7070.
  The PIN Encryption Key encrypted under the Base Key: 3758 EB8D B208 C875.

The command looks like this:

```
<1D#1#1693C76D5493D733#3758EB8DB208C875#>
```

The Network Security Processor returns the following response:

```
<2D#F0224DB34CB2E9B9#19D7#>
```

Generate New Initial Key for PIN Pad Using VISA DUKPT (Command 1E)

Command 1E re-initializes PIN pads that perform VISA Derived Unique Key Per Transaction (DUKPT) key management.

This command by default will generate a 1key-3DES (single-length) session key. Use option A2 to control the length of the generated session key. A new optional field, session key length, has been added as the last field of the command. When option A2 is set to “B”, the host application must include the New Base Derivation key field and the session key length field. If there is no new Base Derivation Key, include the field, but leave it empty.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor’s security policy.

Command

```
<1E#E(MFK.8)(Derivation Key)#Current Key Serial Number#
New Key Serial Number#[E(MFK.8)(New Derivation Key)#]>
[Session Key Length#]>
```

Response

```
<2E#E(Current Key)(New Initial PIN Encryption Key)#
Check Value#>[CRLF]
```

Calling Parameters

`1E`

Field 0, the command identifier.

$E_{MFK.8}$`(Derivation Key)`

Field 1, the Derivation Key encrypted under variant 8 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”.

`Current Key Serial Number`

Field 2, this value is used with the Derivation Key specified in Field 1 to derive the current PIN pad key. This field contains a 10 to 20 byte hexadecimal value. Leading Fs will be suppressed.

`New Key Serial Number`

Field 3, the new key serial number for the PIN pad, left-padded with Fs. If a new Derivation Key is defined in Field 4, then this field generates the new initial key serial number; otherwise, the Derivation Key in Field 1 is used. This field contains a 16 byte hexadecimal value.

`[E`$_{\text{MFK.8}}$`(New Derivation Key)#]`

Field 4, the new Derivation Key encrypted under variant 8 of the MFK. This field is required only if option A2 is set to “B”, for all other cases this field is optional. If it exists, this field contains a 16 or 32 byte hexadecimal value, or a volatile table location. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”. This field is required, but can be empty, if option A2 is set to “B”35 TD0 Tc0 Tw(i Tm Tc 00[SessDerivatiLinglen Key)#])T

`Check Value`

Field 2, the new initial key's check value. The length of this field depends on the length of the session key. It will contain 8 hexadecimal digits if the session key is a 1key-3DES (single-length) key. It will contain 16 hexadecimal digits if the session key is a 2key-3DES (double-length) key.

Table 3-37. Response 2E: Generate New Initial Key for PIN Pad Using VISA DUKPT

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	2E
1	$E_{Current\ KSN}$(New Initial Key)*	16	0 - 9, A - F
2	Check Value	8,16	0 - 9, A - F

*E refers to special encryption defined by VISA.

Usage Notes

- This command is typically used to load a new initial key serial number and a new initial key into a PIN pad without taking the PIN pad out of service. You will use this command in a number of circumstances, including when the PIN pad exceeds its million-transaction limit, when the PIN pad's initial key serial number has been changed, or when the acquirer's Derivation Key has been changed. This command can be used only with PIN pads that support it.
- Before using Command 1E, generate the Derivation Key and set option A2 appropriately.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Generating a new initial key.

- Option A2 is set to “S”.
- Clear-text Derivation Key: 1334 1334 1334 1334.
  The Derivation Key encrypted under variant 8 of the MFK: 4A79 F2A0 E61F EECF.
- The current Key Serial Number: FFFF 9876 5432 10E0 0001.
- The new Initial Key Serial Number: 0123 4567 89.

The command looks like this:

```
<1E#4A79F2A0E61FEECF#9876543210E00001#FFFFFF0123456789#>
```

The Network Security Processor returns the following response:

```
<2E#F90FB12DC2CD138D#1567922B#>
```

This example shows the syntax when the option A2 is set to “B” or “S”, and a new Base Derivation Key is included in field 4. The clear text value of the new Base Derivation Key is 0123456789ABCDEF.

```
<1E#4A79F2A0E61FEECF#9876543210E00001#FFFFFF0123456789#AAA57E
4E99AE9B03#S#>
```

The Network Security Processor returns the following response:

```
<2E#1C96A87EDC8672CF#3AE4C948#>
```

Generating a new initial key using a 2key-3DES (double-length) session key.

- Option A2 is set to “D”.
- Clear-text Base Derivation Key: 1334 1334 1334 1334 5678 5678 5678 5678
  The Base Derivation Key encrypted under variant 8 of the MFK:
  4A79F2A0E61FEECF24103C06FD668967
- The current Key Serial Number: FFFF 9876 5432 10E0 0001.
- The new Initial Key Serial Number: 0123 4567 89.

The command looks like this:

```
<1E#4A79F2A0E61FEECF24103C06FD668967#9876543210E00001#FFFFFF0
123456789#>
```

The Network Security Processor returns the following response:

```
<2E#0C92829F9CDE4DA3#1FBB8F5B87EF5FA0#>
```

This example shows the syntax when the option A2 is set to “B” or “D”.

```
<1E#4A79F2A0E61FEECF24103C06FD668967#9876543210E00001#FFFFFF0
123456789##D#>
```

The Network Security Processor returns the following response:

```
<2E#0C92829F9CDE4DA3#1FBB8F5B87EF5FA0#>
```

This example shows the syntax when the option A2 is set to “B” or “D”, and a new Base Derivation Key is included in field 4.

- The clear text value of the new Base Derivation Key is 0123456789ABCDEF FEDCBA9876543210.
  The New Base Derivation Key encrypted under variant 8 of the MFK:
  AAA57E4E99AE9B0328F6BA950E1664FA

```
<1E#4A79F2A0E61FEECF24103C06FD668967#9876543210E00001#FFFFFF0
123456789#AAA57E4E99AE9B0328F6BA950E1664FA#D#>
```

The Network Security Processor returns the following response:

```
<2E#1B80BEC57C9C0286#FF3C341951FEE2CF#>
```

Generate Check Digits (Command 7E)

This command generates check digits in order to confirm that two parties hold the same key value. Each party calculates the check digits from the key using the same algorithm and then compares results. This command supports both 1key-3DES (single-length) and 2key-3DES (double-length) working keys.

In version 1.30 and above option 4F controls methods I and R.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<7E#Check Digit Method#Variant#E_MFK.V(Working Key)#
[Adjusted Variant#]>
```

Response

```
<8E#Check Digit Method#Generated Check Digits#>
```

Calling Parameters

`7E`

Field 0, the command identifier.

`Check Digit Method`

Field 1, the check digit method. This field contains 1 byte. The possible values are listed in the following table:

Method	Description	Value
A	$E_{adjustedkey}$(0000000000000000)	leftmost 6
F	E_{key}(0123456789ABCDEF)	leftmost 4
I*	E_{key}(key)	rightmost 4
R*	(E_{KEY}(KEY)) XOR KEY	rightmost 4
S	E_{key}(000000000000000)	leftmost 4
V	E_{key}(000000000000000)	leftmost 6

* Method R is allowed only when option 4F is enabled. When option 4F is enabled, method I is not allowed.

`Variant`

Field 2, the variant used to encrypt the working key. This field can be one or two bytes long and can contain the numbers 0 to 31. See Key variants on page 2-2 for a list of supported variants.

$E_{MFK.V}$(Working Key)

Field 3, the working key encrypted under the specified variant of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

[Adjusted Variant#]

Field 4, this field is only present if field 1 contains the letter A. The Adjusted Variant is exclusive-OR with the decrypted working key. This field can contain either a 1 or 2 byte decimal value in the range of 0 to 31 inclusive.

Table 3-38. Command 7E: Generate Check Digits

Field #	Contents	Length (bytes)	Legal Characters
0	Command Identifier	2	7E
1	Check Digit method	1	A, F, I, R, S, V
2	Variant V	1, 2	0 - 31
3	$E_{MFK.V}$(Working Key)*	16, 32	0 - 9, A - F
4	Adjusted Variant**	1, 2	0 - 31

*Can be a volatile table location

**This field is present only if field 1 contains the letter A.

Responding Parameters

8E

Field 0, the response identifier.

Check Digit Method

Field 1, the check digit method supplied in field 1 of the command.

Generated Check Digits

Field 2, the calculated check digits. This field contains a four or six byte hexadecimal value.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Variant 4
- Clear-text PIN Verification Key: 0000 0000 5555 6666.
  The PIN Verification Key encrypted under variant 4 of the MFK: BEFB 77D6 B00C DC17.

Check digit method: A.

The key is exclusive-OR'd with the adjusted variant then encrypts zeros. The leftmost 6 digits of the result are the check digits.

The command looks like this:

```
<7E#A#4#BEFB77D6B00CDC17#4#>
```

The Network Security Processor returns the following response:

```
<8E#A#E7B8A6#>
```

Check digit method: F.

The key encrypts 0123456789ABCDEFF. The leftmost 4 digits of the result are the check digits.

The command looks like this:

```
<7E#F#4#BEFB77D6B00CDC17#>
```

The Network Security Processor returns the following response:

```
<8E#F#E1E3#>
```

Check digit method: IBM method (I).

The key encrypts itself. The rightmost 4 digits of the result are the check digits.

The command looks like this:

```
<7E#I#4#BEFB77D6B00CDC17#>
```

The Network Security Processor returns the following response:

```
<8E#I#46A5#>
```

Check digit method: R.

The key encrypts itself, this cryptogram is XOR'd with the key. The rightmost 4 digits of the result are the check digits.

The command looks like this:

```
<7E#R#4#BEFB77D6B00CDC17#>
```

The Network Security Processor returns the following response:

```
<8E#R#20C3#>
```

Check digit method: Standard Atalla method (S).

The key encrypts zeros. The leftmost 4 digits of the result are the check digits.

The command looks like this:

```
<7E#S#4#BEFB77D6B00CDC17#>
```

The Network Security Processor returns the following response:

```
<8E#S#3BAF#>
```

Check digit method: VISA method (V).

The key encrypts zeros. The leftmost 6 digits of the result are the check digits.

The command looks like this:

```
<7E#V#4#BEFB77D6B00CDC17#>
```

The Network Security Processor returns the following response:

```
<8E#V#3BAFC4#>
```

Translate Working Key for Local Storage Under the Current MFK to the Pending MFK (Command 9E)

Command 9E translates a working key from encryption under the current MFK to encryption under the pending MFK. This command supports both 1key-3DES (single-length) and 2key-3DES (double-length) keys.

This command is enabled in the Network Security Processor's default security policy.

Command

<9E#Variant#$E_{MFK.V}$(Working Key)#>

Response

<AE#$E_{Pending\ MFK.V}$(Working Key)#Working Key Check Digits#>
[CRLF]

Calling Parameters

9E

Field 0, the command identifier.

Variant

Field 1, the variant of the current MFK under which the working key has been encrypted. This field can be one or two bytes long and can contain the numbers 0 to 31.

$E_{MFK.V}$(Working Key)

Field 2, the working key encrypted using the variant of the MFK specified in field one. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

Table 3-39. Command 9E: Translate Working Key for Local Storage Under Current MFK to Pending MFK

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	9E
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.V}$(Working Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`AE`

Field 0, the response identifier.

$E_{Pending\ MFK.V}$(Working Key)

Field 1, the working key decrypted using the variant of the current MFK specified in field one of the command and re-encrypted using the same variant of the pending MFK. This field contains a 16 or 32 byte hexadecimal value.

`Working Key Check Digits`

Field 2, check digits; that is the first four digits that result from encrypting zeros using the working key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-40. Response AE: Translate Working Key for Local Storage Under Current MFK to Pending MFK

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AE
1	$E_{Pending\ MFK.V}$(Working Key)	16, 32	0 - 9, A - F
2	Working Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Load the pending MFK (PMFK1) into the Network Security Processor non-volatile key table.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values. The pending MFK is 98107645FED3BCA2 2ABC3DEF45670189.

Translating a data encryption key (KD).

- Variant: 2
- Clear-text Data Encryption Key (KD): 0123 4567 89AB CEDF.
  The Data Encryption Key encrypted under variant 2 of the MFK: 80BC DEAC 5703 BC84.

The command looks like this:

```
<9E#2#80BCDEAC5703BC84#>
```

The Network Security Processor returns the following response:

```
<AE#7B8CA7B9B6E17408#D5D4#>
```

Replace the Current MFK with the Pending MFK (Command 9F)

Command 9F replaces the current MFK with the pending MFK. When the pending MFK is promoted to the MFK, the name of the new MFK increments.

This command is enabled in the Network Security Processor's default security policy.

Note. Upon successful execution of this command, all keys in the volatile table are erased.

Command

```
<9F#MFK Name#MFK Check Digits#Pending MFK Name#
Pending MFK Check Digits#>
```

Response

```
<AF#OK#>[CRLF]
```

Calling Parameters

`9F`

Field 0, the command identifier.

`MFK Name`

Field 1, the current MFK's name.

`MFK Check Digits`

Field 2, the current MFK's check digits; that is the result of encrypting zeros using the MFK. This field contains a four byte hexadecimal number.

`Pending MFK Name`

Field 3, the pending MFK's name, PMFK1.

`Pending MFK Check Digits`

Field 4, the pending MFK's check digits; that is the result of encrypting zeros using the pending MFK. This field contains a four byte hexadecimal value.

Table 3-41. Command 9F: Replace Current MFK with Pending MFK (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	9F
1	MFK name	0, 4	0 -9, A- Z

Table 3-41. Command 9F: Replace Current MFK with Pending MFK (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
2	MFK Check Digits	4	0 - 9, A - F
3	Pending MFK name	5	PMFK1
4	Pending MFK Check Digits	4	0 - 9, A - F

Responding Parameters

`AF`

Field zero, the response identifier.

`OK`

Field one, an indicator that the current MFK has been replaced and the volatile table has been erased.

Table 3-42. Response AF: Replace Current MFK with Pending MFK

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AF
1	Verification indicator	2	OK

Usage Notes

Load the pending MFK and translate all working keys using the pending MFK, see Translate Working Key for Local Storage Under the Current MFK to the Pending MFK (Command 9E) on page 3-63.

Command 9F increments the MFK name to the next value in this list: “MFK2”, “MKF3”, “MFK4”, “MFK5”, “MFK6”, “MFK7”, “MFK8”, “MFK9”, “MFKA”, “MFKB”, “MFKC”…. “MFKZ”, “MFK2”, “MFK3” …

For Example:
If the current MFK name is “MFK1” after command 9F it will be “MFK2”.
If the current MFK name is “MFK2” after command 9F it will be “MFK3”.
If the current MFK name is “MFKZ” after command 9F it will be “MFK2”.

NOTE: The MFK name will not increment to “MFK1”. It will not be used by command 9F. This name is reserved for use with the SCA-3.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values. The pending MFK clear-text value is 98107645FED3BCA2 2ABC3DEF45670189.

Replacing the current MFK with a pending MFK.

- Current MFK's name: MFK1.
- Current MFK's check digits: 057A.
- Pending MFK's name: PMFK1.
- Pending MFK's check digits: 6270.

The command looks like this:

```
<9F#MFK1#057A#PMFK1#6270#>
```

The Network Security Processor returns the following response:

```
<AF#OK#>
```

Translate an encrypted key between ECB and CBC modes (command 113)

Command 113 changes the encryption mode used to encrypt a working key. This command supports Electronic Code Book (ECB) and Cipher Block Chaining (CBC) modes. The CBC initialization vector is binary zeros. It is not supplied in the command.

The working key should be a 2key-3DES (double-length) key. The working key can be a 1key-3DES (single-length) key if option 6A is enabled in the Network Security Processor's security policy.

This command is enabled in the Network Security Processor's default security policy.

Command

<113#Variant V#$E_{MFK.0}$(KEK)#$E_{KEK.V}$(Working Key)#[Mode#]>

Response

<213#$E_{KEK.V}$(Working Key)#Working Key Check Digits#>[CRLF]

Calling Parameters

113

Field 0, the command identifier.

Variant V

Field 1, the variant applied to the KEK prior to encrypting the working key. This field contains a 1 or 2 digit value in the range of 0 through 31.

$E_{MFK.0}$(KEK)

Field 2, the Key Exchange Key encrypted using ECB under variant zero of the MFK. This field should contain a 32 character hexadecimal value or a volatile table location that contains a 2key-3DES key (double-length). It can contain a 16 character hexadecimal value or a volatile table location of a 1key-3DES (single-length) key, only if option 6A is enabled in the Network Security Processor's security policy **and** if field 3 contains a 1key-3DES key. If field 3 contains a 2key-3DES key, this field must contain a 2key-3DES key, or a volatile table location that contains a 2key-3DES key.

$E_{KEK.V}$(Working Key)

Field 3, the working key encrypted, using the mode specified in field 4, under the variant specified in field 1, of the Key Exchange Key. This field should contain a 32

character hexadecimal value. It can contain a 16 character hexadecimal value only if option 6A is enabled in the Network Security Processor's security policy.

`[Mode#]`

Field 4, the mode of DES used in the translation of the working key. This field is optional. If not present the working key will be translated from ECB to CBC mode of DES. If present this field consists of two characters:

1# - indicates translate the working from ECB to CBC mode of DES.

2# - indicates translate the working key from CBC to ECB mode of DES.

Table 3-43. Translate an encrypted key between ECB and CBC modes

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	113
1	Variant V	1, 2	0 - 31
2	$E_{MFK.0}$(KEK)*	16, 32	0 - 9, A - F
3	$E_{KEK.V}$(Working Key)	5	0 - 9, A - F
4	[Mode#]	empty, 2	1# or 2#

* Can be a volatile table location

Responding Parameters

`213`

Field zero, the response identifier.

$E_{KEK.V}$`(Working Key)`

Field 1, the working key encrypted, using the mode specified in command field 4, under the variant specified in command field 1, of the Key Exchange Key. The length of this field is the same length as field 3 in the command.

`Working Key Check Digits`

Field 2, the first four digits that result from encrypting zeros using the working key. This field contains a four byte hexadecimal value. If Option 88 is enabled then a 6-digit check digit will be returned.

Table 3-44. Response 213: Translate an encrypted key between ECB and CBC modes

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	213
1	$E_{KEK.V}$(Working Key)	16, 32	0 - 9, A - F
2	Working Key Check Digits	4,6	0 - 9, A - F

Usage Notes

- Encrypt the KEK under variant 0 of the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

The KEK clear-text value: 0123 4567 89AB CDEF FEDC BA98 7654 3210, check digits 08D7.

The KEK encrypted under variant 0 of the MFK:
9007B8751BB7AB4E0B176C3EBEED18AF

The working key clear-text value: 1234123412341234 5678567856785678, check digits DB82

ECB to CBC Translation

The working key, ECB encrypted, under variant 1 of the KEK:
65F36EFD9E5518DDEEAB6E607C3E6EA7

The command looks like this:

```
<113#1#9007B8751BB7AB4E0B176C3EBEED18AF#65F36EFD9E5518DDEEAB6
E607C3E6EA7#>
```

The Network Security Processor's response is:

```
<213#65F36EFD9E5518DDF479C816D90734E8#DB82#>
```

CBC to EBC Translation

The working key, CBC encrypted, under variant 1 of the KEK:
65F36EFD9E5518DDF479C816D90734E8

The command looks like this:

```
<113#1#9007B8751BB7AB4E0B176C3EBEED18AF#65F36EFD9E5518DDF479C
816D90734E8#2#>
```

The Network Security Processor's response is:

```
<213#65F36EFD9E5518DDEEAB6E607C3E6EA7#DB82#>
```

Generate ATM MAC or Data Encryption Key (Command 11D)

Command 11D allows a PIN Encryption Key, Data Encryption Key, or MAC key to be generated, and in addition to being encrypted under a specified MFK variant (1, 2 or 3), it will be encrypted under variant 0 of a Key Exchange Key (KEK). The KEK is provided encrypted under a specified variant (0 or 5) of the MFK. This command generates a 1key-3DES (single-length) or 2key-3DES (double-length) working key.

This command is not enabled in the Network Security Processor's default factory security policy. You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

Command

<11D#Variant V#Variant K#$E_{MFK.K}$(Key Exchange Key)#
[Key Length]#>

Response

<21D#$E_{MFK.V}$(Working Key)#$E_{Key\ Exchange\ Key}$(Working Key)#
Working Key Check Digits#>

Calling Parameters

11D

Field 0, the command identifier.

Variant V

Field 1, the variant (V) of the MFK under which the generated working key will be encrypted. This field contains a 1 byte decimal value which can be either 1, 2, or 3.

Variant K

Field 2, the variant (K) of the MFK under which the KEK has been stored. This field contains a 1 byte decimal value which can be either 0 or 5.

$E_{MFK.K}$(KEK)

Field 3, the Key Exchange Key encrypted using the variant of the MFK specified in Field 2. This field contains a 16 byte or 32 byte hexadecimal value, or a volatile table location. If the Key Length field contains a value of 2 (generate 2key-3DES working key), the KEK has to be 2key-3DES (double-length). This KEK can not be a replicated 1key-3DES (single-length) key.

`[Key Length]`

Field 4, length of the generated Working Key. This is an optional field. If used, it is one byte long and can be empty, or contain the number 1 (to generate 1key-3DES key) or 2 (to generate 2key-3DES key). If this field is not present in the command, a 1key-3DES key will be generated.

Table 3-45. Command 11D: Generate ATM MAC or Data Encryption Key

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	11D
1	Variant V	1	1,2,3
2	Variant K	1	0,5
3	$E_{MFK.K}$(Key Exchange Key)*	16, 32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`21D`

Field 0, the response identifier.

$E_{MFK.V}$`(Working Key)`

Field 1, the working key encrypted using the variant of the MFK specified in Field 1 of the command. This field contains a 16 or 32 byte hexadecimal value.

$E_{\text{Key Exchange Key}}$`(Working Key)`

Field 2, the working key encrypted under the KEK. This field contains a 16 or 32 byte hexadecimal value.

`Working Key Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the Working Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 3-46. Response 21D: Generate ATM MAC or Data Encryption Key

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	21D
1	$E_{MFK.V}$ (Working Key)*	16, 32	0 - 9, A - F
2	$E_{\text{Key Exchange Key}}$(Working Key)	16, 32	0 - 9, A - F
3	Working Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Encrypt the Key Exchange Key under variant zero of the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

This command generates a random key therefore your results will be different.

Generating an ATM MAC Key.

- Variant V: 3
- Variant K: 5
- Key Length: 1
- Clear-text Key Encryption Key (KEK): 0000 1111 2222 3333.
  The Key Exchange Key encrypted under variant 5 of the MFK: 784D DF5F 89FB 9EBF.

The command looks like this:

```
<11D#3#5#784DDF5F89FB9EBF#>
```

The Network Security Processor returns the following response:

```
<21D#8FC5F6000E039870#94F5064C96FE9841#2D3D#>
```

Example 2

Generating an ATM MAC Key.

- Variant V: 3
- Variant K: 5
- Key Length: 2
- Clear-text Key Encryption Key (KEK): 0000 1111 2222 3333 4444 5555 6666 7777.
  The Key Exchange Key encrypted under variant 5 of the MFK: 784D DF5F 89FB 9EBF CDCB 224A E777 56B2.

The command looks like this:

```
<11D#3#5#784DDF5F89FB9EBFCDCB224AE77756B2#2#>
```

The Network Security Processor returns the following response:

```
<21D#25166617EC743AB125166617EC743AB1#F8C6AB5B46CFD570F8C6AB5
B46CFD570#D5D4#>
```

4 Processing Personal Identification Numbers

This section outlines the tasks involved in processing PINs and describes the PIN processing commands supported in the Network Security Processor.

To skip this introduction go to Table 4-10 for a list of commands.

About PIN Processing

The personal identification number – or PIN – is the secret, unique number that identifies a consumer who is transacting business on an automated teller machine (ATM) or point of sale (POS) network.

The following list outlines the processes that a PIN typically undergoes, starting with its entry into an ATM or PIN pad and ending with its verification by the issuing host.

1. The PIN is entered into an ATM or PIN pad.
2. The ATM or PIN pad formats the PIN into a PIN block.
3. The ATM or PIN pad encrypts the PIN block and sends it to the host.
4. The host determines whether the PIN corresponds to an account that belongs to its own financial institution or to another institution.
    a. If the PIN corresponds to an account at this financial institution (making it an “on-us” transaction), then the host verifies the PIN and confirms whether sufficient funds are available for the requested transaction.
    b. If the PIN does not correspond to an account at this financial institution (making it a “not-on-us” transaction), then the host translates the PIN and sends it to the switch encrypted under the acquirer’s working key. The switch determines the issuing financial institution, then translates the PIN block and sends it to another switch or to the issuing financial institution encrypted under the issuer working key. When the PIN block arrives at the issuer, the host verifies it and confirms whether sufficient funds are available for the requested transaction.

The following section explains the programming tasks that you must accomplish to facilitate PIN processing.

PIN Processing Tasks

Processing PINs typically involves the following tasks.

- Encrypting PINs or PIN blocks
- Translating PIN blocks
- Verifying incoming PIN blocks and authorizing or denying transaction requests.

Encrypting PINs

This subsection explains how PINs are encrypted in ATM networks and VISA DUKPT POS networks.

PIN Encryption in ATM Networks

In ATM networks, PINs can be encrypted at two different places:

- The point of capture (an ATM) or
- At the host using a Network Security Processor.

Encrypting PINs at an ATM involves two steps:

1. The ATM formats the PIN into a **PIN block**. PIN blocks are packages of data that contain the PIN, pad characters, and sometimes other information like the length of the PIN. The Network Security Processor supports a variety of PIN Block Types.
2. Once the PIN has been formatted into a PIN block, the ATM encrypts it using a PIN Encryption Key that is common to both the ATM and its host. To encrypt PINs at the host's Network Security Processor, the clear-text PIN must travel from the ATM to the host. If the host is unable to verify the PIN, then the PIN is formatted into a PIN block and encrypted using a PIN Encryption Key. Formatting and encrypting the PIN enables it to be transmitted to a node that can verify it.

The point to remember is that PINs never pass to the switch in clear-text format when they have passed first through an intercepting processor.

PIN Encryption In VISA DUKPT Networks

In networks that use VISA DUKPT key management, PIN pads are always responsible for encrypting PINs. The difference between PIN encryption on VISA DUKPT networks and PIN encryption at the ATM in ATM networks is that on VISA DUKPT networks, the PIN Encryption Key used is unique for every transaction.

Translating PIN Blocks

Once an ATM or PIN pad receives a PIN, the objective is to verify that it corresponds to a valid account. If this verification is not done at the ATM or PIN pad, then the PIN block must travel to the host or switch to be verified. If the PIN is verified at the switch or issuer host, then the PIN block must be **translated** each time it stops at an intermediary, or intercept, processor. Translation refers simply to the process of changing the PIN block's type or the PIN Encryption Key in use so that the PIN block can travel from one processor to the next. Typically, the first intercept processor receives the PIN block encrypted in the type supported by the sending ATM or PIN pad, then translates it into an ANSI PIN block. Most networks require ANSI PIN blocks.

Verifying Incoming PIN Blocks

PINs are not verified directly. The PIN is known only to the card holder; no one else – not even the issuing financial institution – knows the PIN's clear-text value. PIN verification is facilitated by means of the **PIN verification number (PVN).** The PIN verification number is derived from an algorithm that takes as its input the PIN and the Primary Account Number (PAN). The result is in turn operated on by the PIN Verification Key; the result is a calculated PIN verification number. The PIN verification number calculated at the verifying node is compared to the PVN that is encoded on consumer's credit or debit card, or stored on a host database. If the two values match, then the PIN has been verified.

The Network Security Processor supports the following methods of PIN verification:

- Identikey
- IBM 3624
- Visa
- Atalla DES Bilevel
- Diebold
- NCR
- Atalla 2x2
- Burroughs

PIN Sanity Error

When an encrypted PIN is translated or verified, it is decrypted with the incoming PIN Encryption Key. The Network Security Processor examines the format of the decrypted PIN block. Option 4B specifies the type of PIN sanity test to be performed. If the Network Security Processor determines that the decrypted PIN block is not valid it returns a PIN Sanity error in the response. The usual causes of PIN sanity errors are:

- The wrong key was specified as the incoming PIN Encryption Key. Or the correct key was specified, however this key was not encrypted under variant 1 of the MFK.
- The PIN length is incorrect. Option A0 can be used to configure the Network Security Processor for a specific minimum PIN length, the default is 4 digits. The maximum PIN length is fixed at 12 digits. If the decrypted PIN does not fall within minimum and maximum range a sanity error will be returned. Option A1 configures the Network Security Processor to return an “L” if the decrypted PIN is less than the minimum PIN length. The Network Security Processor does not allow a PIN greater than 12 digits. When the Network Security Processor decrypts a PIN that is greater than 12 digits, it will always return a sanity error even if option A1 is set to “L”.

- Wrong data in the PIN data block. For example, the ANSI PIN block requires the rightmost 12 digits of the account number excluding the check digit. If the origin and destination do not use the exact same 12 digits in the PIN data block, a sanity error will be returned.

PIN Block Types

The Network Security Processor supports a variety of PIN block types; not all PIN block types are supported in all commands. Each PIN block type requires a specific set of data. This data is provided as separate fields at the end of the command. Each of these extra fields is delimited with a “#”, just like any other field in the command.

PIN Block Type	Value	PIN block data fields added to the end of the command.
ANSI PIN Block	1	1, labeled Field A
IBM 3624 PIN Block	2	3, labeled Fields A, B, and C
PIN/Pad PIN Block	3	2, labeled Fields A and B
Docutel PIN Block	3	2, labeled Fields A and B
IBM Encrypting PIN Pad PIN Block	4	1, labeled Field A
Burroughs PIN Block	5	2, labeled Fields A and B
VISA Derived Unique Key Per Transaction PIN Block	7	3, labeled Fields, A, B, and C
ISO-3 PIN Block	8	1, labeled Field A
IBM 4731 PIN Block	9	4, labeled Fields A, B, C, and D

The following sections define the contents of the PIN Block Data for each supported PIN block type.

ANSI PIN Block

The ANSI PIN block is also referred to as an ISO-0 PIN block. The ANSI PIN block format 1 is not supported in the Network Security Processor.

PIN Block Data

The ANSI PIN Block requires one PIN Block Data field; the last field of the command.

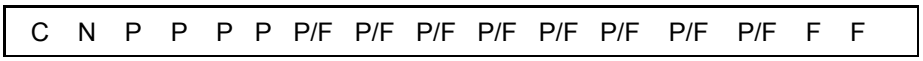
Table 4-1. ANSI - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Twelve rightmost PAN digits (excluding check digits)	12	0 - 9

Constructing an ANSI PIN Block

The ANSI PIN block is the result of performing an exclusive-OR on two data blocks, the PIN block and the account number block.

Figure 4-1. PIN Block



C

The control field. A four bit value; hexadecimal 0.

N

The length of the PIN. A four bit hexadecimal value 4 to 9, A, B, or C. A ten digit PIN is represented as A, an 11 digit PIN is represented as B, and a 12 digit PIN is represented as C.

P

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

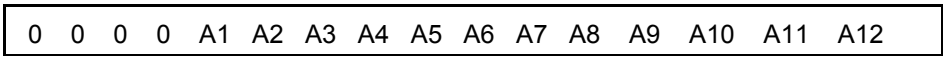
F

Pad character. A four bit value; hexadecimal F.

P/F

A PIN digit or a pad character, depending on the length of the PIN.

Figure 4-2. Account Number Block



0

Pad character. A four bit value; hexadecimal 0.

A1 to A12

The 12 rightmost digits of the Primary Account Number (PAN), **excluding** the check digit. A1 is the most significant digit; A12 is the digit that immediately precedes the Primary Account Number's check digit.

Example

```
PIN = 1234

Primary Account Number = 5999997890123457

PIN Block             =  041234FFFFFFFFFF

Account Number Block =  0000999789012345

exclusive-OR
the PIN and
Account Number Blocks

ANSI PIN Block =         0412AD6876FEDCBA
```

IBM 3624 PIN Block

This encrypted PIN block is 18 hexadecimal characters. When a command contains an encrypted IBM 3624 PIN block, the last field of the Network Security Processor's response will be the two digit sequence number.

PIN Block Data

The IBM 3624 PIN block requires three PIN Block Data fields; the last three fields of the command.

Table 4-2. IBM 3624 - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Pad character*	1	0 - 9, A - F, X, W
B	Twelve digit; required but only used in command 39.	12	0 - 9
C	$E_{MFK.2}$(KC)**	16	0 - 9, A - F

* Legal pad characters are a hexadecimal value, X and W. The value X indicates that the pad character is unspecified but can be any hexadecimal character. The value W indicates that the sanity check, which tests for the existence of pad digits and valid PIN digits, will not be performed.

** Can be a volatile table location.

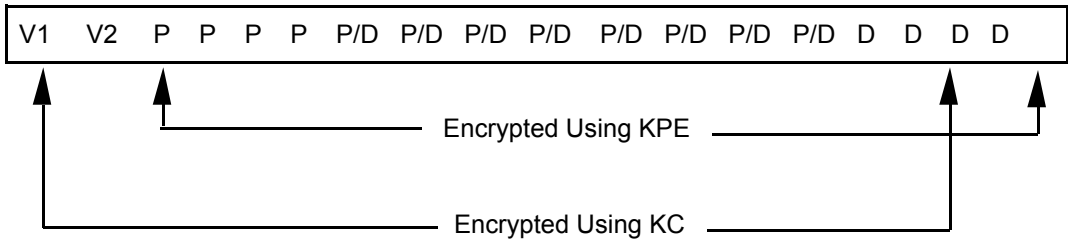
PIN Block

The IBM 3624 PIN block is produced in two steps:

1. Encrypt the eight rightmost bytes (16 hexadecimal characters) using the PIN Encryption Key (KPE).
2. Encrypt the eight leftmost bytes (16 hexadecimal characters) using the Communications Key.

The resulting cryptogram is written as $E_{KC}(E_{KPE}(\text{PIN Block}))$.

Figure 4-3. IBM 3624 PIN Block



`V1 V2`

Sequence number; two hexadecimal characters.

P

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

D

Pad character. A four bit hexadecimal value.

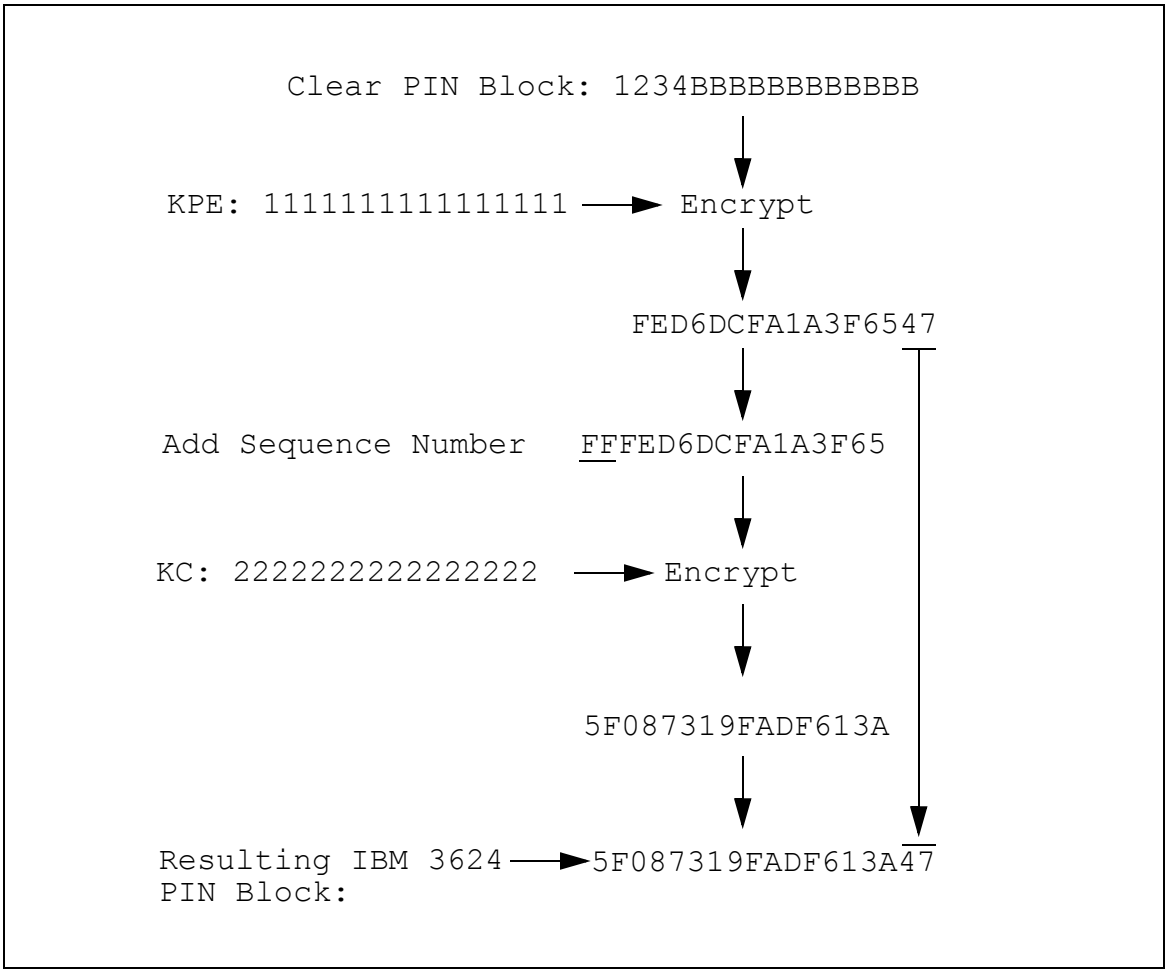
P/D

A PIN digit or a pad character, depending on the PIN's length.

Example

```
KPE = 1111 1111 1111 1111
KC = 2222 2222 2222 2222
PIN = 1234
Pad = B
Sequence Number = FF
```

Figure 4-4. Encrypted IBM 3624 PIN Block



PIN/Pad PIN Block

This PIN block is used by Diebold and some other ATM and PIN pad vendors.

PIN Block Data

The PIN/pad character PIN block requires two PIN Block Data fields; the last two fields of the command.

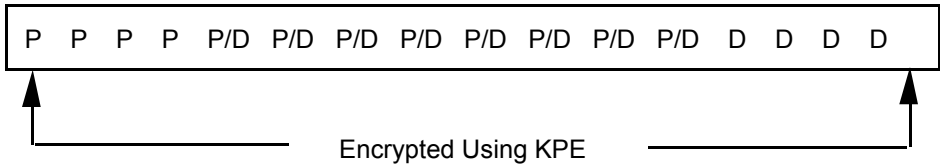
Table 4-3. PIN/Pad - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Pad character*	1	0 - 9, A - F, X, W
B	Twelve digit; required but only used in command 39.	12	0 - 9

* Legal pad characters are a hexadecimal value, X and W. The value X indicates that the pad character is unspecified but can be any hexadecimal character. The value W indicates that the sanity check, which tests for the existence of pad digits and valid PIN digits, will not be performed.

PIN Block

Figure 4-5. PIN/Pad Character PIN Block



`P`

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

`D`

Pad character. A four bit hexadecimal value. All pad characters must be the same value.

`P/D`

A PIN digit or a pad character, depending on the PIN's length.

Example

```
PIN = 1234
Pad = F
PIN Pad PIN block = 1234FFFFFFFFFFF
```

Docutel PIN Block

The PIN digits are followed by a single character F and numeric pad characters.

PIN Block Data

The Docutel PIN block requires two PIN Block Data fields; the last two fields of the command.

Table 4-4. Docutel - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Pad character*	1	0 - 9, X, W
B	Twelve digit; required but only used in command 39.	12	0 - 9

* Legal pad characters are a 0 through 9, X and W. The value X indicates that the pad character is unspecified but can be any hexadecimal character. The value W indicates that the sanity check, which tests for the existence of pad digits and valid PIN digits, will not be performed.

PIN Block

Figure 4-6. Docutel PIN Block

P	P	P	P	P/F	P/F/D	P/F/D	P/F/D	P/F/D	P/F/D	P/F/D	P/F/D	F/D	D	D	D

`P`

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

`F`

The four bit hexadecimal character F. This PIN block can contain only one F; it delimits the PIN.

`D`

Pad character. A four bit hexadecimal value in the range of 0 through 9.

Example

```
PIN = 1234
Pad =  10897645231
Docutel PIN block = 1234F10897645231
```

IBM Encrypting PIN Pad PIN Block

PIN Block Data

The IBM encrypting PIN pad PIN block requires one PIN Block Data data field; the last field of the command.

Table 4-5. IBM Encrypting PIN Pad - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Twelve digit; required but only used in command 39.	12	0 - 9

PIN Block

Figure 4-7. IBM Encrypting PIN Pad

C

The length of the PIN. A four bit hexadecimal value 4 to 9, A, B, or C. A ten digit PIN is represented as A, an 11 digit PIN is represented as B, and a 12 digit PIN is represented as C.

P

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

F

Pad character. A four bit value; hexadecimal F.

P/F

A PIN digit or a pad character, depending on the PIN's length.

S

The sequence number. Two 4 bit hexadecimal characters.

Example

```
PIN = 1234
Sequence Number = 07
Pin Block = 41234FFFFFFFFF07
```

Burroughs PIN Block

This PIN block is similar to the PIN/pad character PIN block, except that the PIN digits are ASCII hexadecimal characters instead of four bit hexadecimal values. A Burroughs PIN Block supports a maximum of eight PIN digits.

PIN Block Data

The Burroughs PIN block requires two PIN Block Data fields; the last two fields of the command.

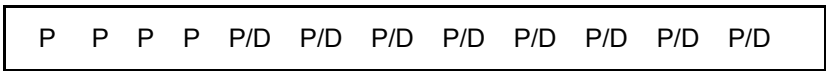
Table 4-6. Burroughs - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Pad character*	1	0 - 9, A - F, X, W
B	Twelve digit; required but only used in command 39.	12	0 - 9

* Legal pad characters are a hexadecimal value, X and W. The value X indicates that the pad character is unspecified but can be any hexadecimal character. The value W indicates that the sanity check, which tests for the existence of pad digits and valid PIN digits, will not be performed.

PIN Block

Figure 4-8. Burroughs PIN Block Type



`P`

PIN digit. Each PIN digit is converted to an ASCII hexadecimal value, 30 through 39 represents the values 0 through 9.

`D`

Pad character. A four bit hexadecimal value.

`P/D`

A PIN digit or a pad character, depending on the PIN's length.

Example

```
PIN = 1234
Pad = F
Burroughs PIN block = 31323334FFFFFFFF
```

ISO-3 PIN Block

The ISO-3 PIN block is the result of performing an exclusive-OR on two data blocks, the PIN block and the account number block.

PIN Block Data

The ISO-3 PIN block requires one PIN Block Data field; the last field of the command.

Table 4-7. ISO-3 - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	Twelve rightmost PAN digits (excluding check digits)	12	0 - 9

PIN Block

Figure 4-9. ISO-3 PIN Block

C N P P P P P/R P/R P/R P/R P/R P/R P/R P/R R R

C

The control field. A four bit value; hexadecimal 3.

N

PIN length. A four bit hexadecimal value in the range of 4 through 9, A, B, or C.

P

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

R

Random pad character. A four bit hexadecimal value in the range of A through F.

Figure 4-10. ISO-3 Account Number Block

0 0 0 0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12

0

Pad character. A four bit hexadecimal value 0.

A1 to A12

The 12 rightmost digits of the Primary Account Number (PAN), **excluding** the check digit. A1 is the most significant digit; A12 is the digit that immediately precedes the Primary Account Number's check digit.

Example

```
PIN = 1234

Primary Account Number = 5999997890123457

Random Pad             = DBFFAEBACE

PIN Block              = 341234DBFFAEBACE

Account Number Block   = 0000999789012345

exclusive-OR
the PIN and
Account Number Blocks

ISO-3 PIN Block =        3412AD4C76AF998B
```

IBM 4731 PIN Block

The ATM Master Key encrypts the PIN, it is not provided in the command.

PIN Block Data

The IBM 4731 PIN block requires four PIN Block Data fields; the last four fields of the command.

Table 4-8. IBM 4731 - PIN Block Data

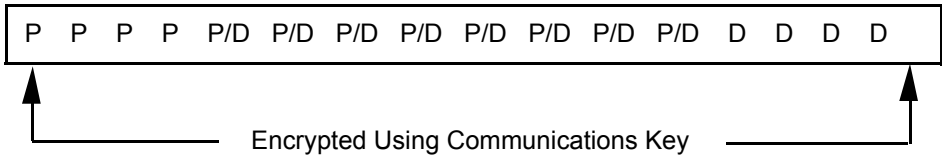
Field #	Contents	Length (bytes)	Legal Characters
A	Pad Character*	1	0 - 9, A - F, X, W
B	PAN	12	0 - 9
C	$E_{MFK.3}(KC)$***	16	0 - 9, A - F
D	ICV	16	0 - 9, A - F

* Legal pad characters are a hexadecimal value, X and W. The value X indicates that the pad character is unspecified but can be any hexadecimal character. The value W indicates that the sanity check, which tests for the existence of pad digits and valid PIN digits, will not be performed.

*** Can be a volatile table location.

PIN Block

Figure 4-11. IBM 4731 PIN Block



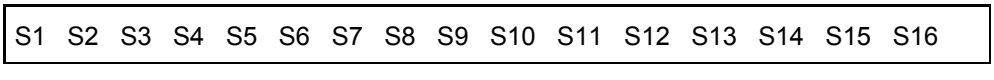
`P`

PIN digit. A four bit hexadecimal value in the range of 0 through 9.

`D`

Pad character. A four bit hexadecimal value.

Figure 4-12. IBM 4731 ICV



`S1 to S16`

A 16 hexadecimal character value.

Example

Master Key = C8B3 047C F7A4 2A70

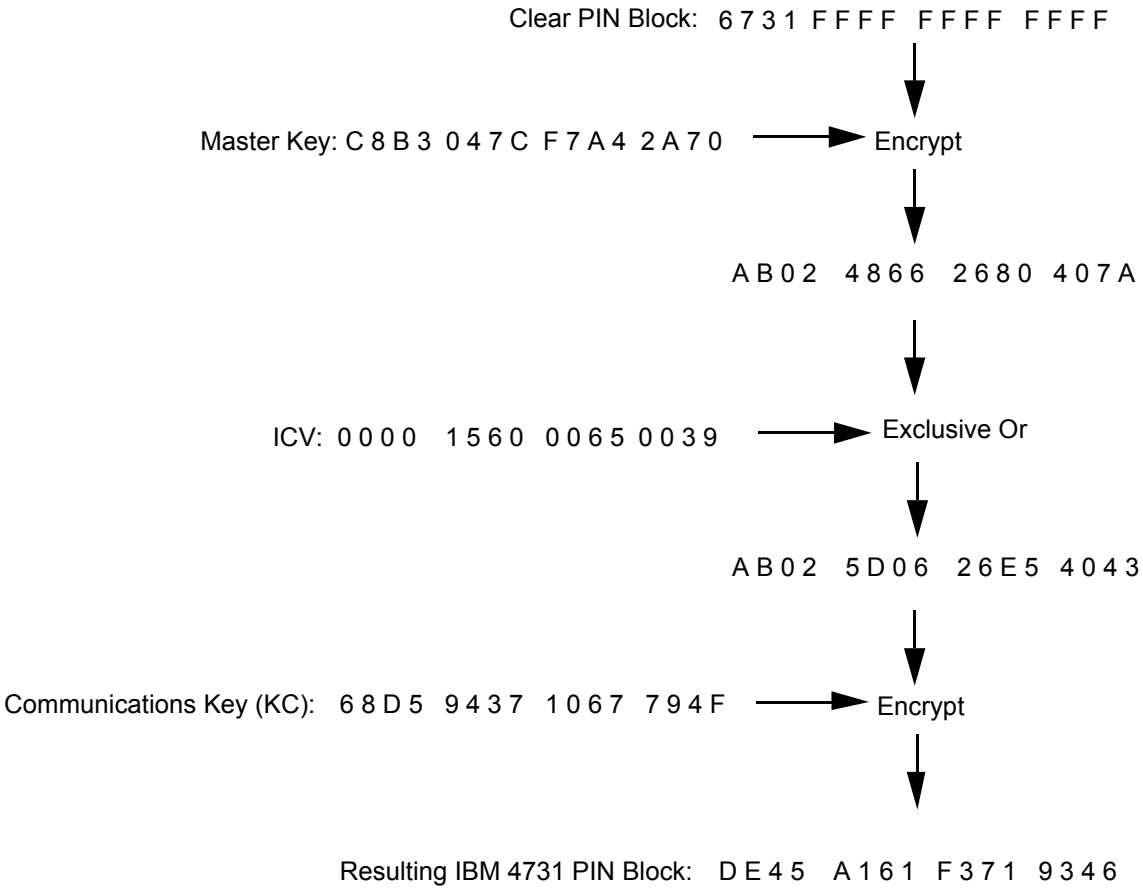
Communication Key = 68D5 9437 1067 794F

ICV = 0000 1560 0065 0039

PIN = 6731

Pad = F

Figure 4-13. Encrypted IBM 4731 PIN Block



VISA Derived Unique Key Per Transaction PIN Block

PIN Block Data

The VISA DUKPT PIN block requires three PIN Block Data fields; the last three fields of the command.

Table 4-9. VISA DUKPT - PIN Block Data

Field #	Contents	Length (bytes)	Legal Characters
A	PAN digits for ANSI PIN block	12	0 - 9
B	Key serial number required to generate current PIN Encryption Key.	10 - 20	0 - 9, A - F
C	PIN encryption key derivation algorithm.	1	1, S, D
	When option A2 set to “B”, the Network Security Processor generates a 1key-3DES (single-length) session key when this field contains either a “1” or “S”, it generates a 2key-3DES (double-length) session key when this field contains the letter “D”.		
	When option A2 is set to “S”, this field must contain the number “1” or the letter “S”.		
	When option A2 is set to “D”, this field must contain the number “1” or the letter “D”.		
	The length of the Base Derivation Key must be greater than or equal to the length of the session key.		

PIN Block

To better understand the example below, it is important to understand specific terms that are unique to the VISA DUKPT methodology.

A 2key-3DES (double-length) key used to encrypt the Initial Key Serial Number (IKSN) to obtain the Initial PIN Encryption Key (IPEK).

A 20 character value that is transmitted from the EFT/POS terminal to the host. It allows the host to determine the key used to encrypt the PIN. The KSN consists of the Initial Key Serial Number (59 bits) + the Encryption Counter (21 bits).

The leftmost 64 bits of the Key Serial Number.

The result of encrypting the IKSN with the DK. (This value is not used to encrypt PIN); see Current PIN Encryption Key.

The result of encryption of the KSN with the IPEK.

Exclusive-OR the last byte of current key with FF.

Exclusive-OR the last two bytes of current key with FFFF.

Example

The purpose of this example is to show how the current single-DES PIN Encryption Key is used to encrypt an ANSI PIN block and also how the Message Authentication Codes are generated. For information on 3DES-DUKPT see

The POS terminal does not use this algorithm to generate keys, see the Visa document for a complete description of the terminal and host security module algorithms.

Generate the current single-DES PIN Encryption Key and encrypt an ANSI PIN Block

Input data

KSN: FFFF 9876 5432 10E0 0001

Derivation Key (DK): 1334 1334 1334 1334

PIN = 1234 5678 901

PAN = 0002 3456 7890

ANSI PIN Block = 0B12 3454 4CC6 676F

- To generate the Initial Key Serial Number (IKSN) take the leftmost 16 characters of the KSN. IKSN = FFFF 9876 5432 10E0.
- To generate the Initial PIN Encryption Key (IPEK) encrypt IKSN with the DK. IPEK = 3466 11AE D3F1 23B4.
- To generate the current key encrypt (using the special VISA technique) the rightmost 16 characters of the KSN with the IPEK.
  1. Exclusive-OR IPEK with KSN = AC10 459C C311 23B5
  2. Encrypt the step 1 result with IPEK = 3D95 A124 8CC9 B178
  3. Exclusive-OR step 2 result with IPEK = 09F3 B08A 5F38 92CC. This is the Current Key.

- To generate the current PIN Encryption Key exclusive-OR the rightmost byte of the current key with FF = 09F3 B08A 5F38 9233.
  This is the Current PIN Encryption Key.
- To generate VISA DUKPT encrypted ANSI PIN Block:

  1. Exclusive-Or the ANSI PIN Block with the current PIN Encryption Key.
     0B12 3454 4CC6 676F exclusive-OR 09F3 B08A 5F38 9233 = 02E1 84DE 13FE F55C.
  2. Encrypt the step 1 result with the current PIN Encryption Key.
     Encrypt 02E1 84DE 13FE F55C with 09F3 B08A 5F38 9233 = CFD0 BB26 8F94 D378.
  3. Exclusive-Or the step 2 result with the Current PIN Encryption Key.
     CFD0 BB26 8F94 D378 exclusive-OR 09F3 B08A 5F38 9233 = C623 0BAC D0AC 414B. This is the VISA DUKPT PIN Block.

Generate the current MAC Key and MAC1, MAC2 and MAC3

PAN: 1234 1234

Debit/Credit Indicator: 567

Amount: $85,678

- To generate the current MAC Key exclusive-OR the rightmost two bytes of the current key with FFFF.
  09F3 B08A 5F38 6D33, this is the Current MAC Key.
- To generate VISA DUKPT MAC1, MAC2, MAC3:.

  1. Concatenate PAN, Debit/Credit Indicator/Amount 1234 1234 5678 5678
     Note: Pad with F to provide a multiple of 16 digits.
  2. Exclusive-OR step 1 result with the Current MAC Key.
     1234 1234 5678 5678 exclusive-OR 09F3 B08A 5F38 6D33 = 1BC7 A2BE 0940 3B4B.
  3. Encrypt step 2 with the Current MAC Key.
     Encrypt 1BC7 A2BE 0940 3B4B with 09F3 B08A 5F38 6D33 = 6B9B A42D 1303 A43D.
  4. Exclusive-Or step 3 with the Current MAC Key.
     6B9B A42D 1303 A43D exclusive-OR 09F3 B08A 5F38 6D33 = 6268 14A7 4C3B C90E.
     If the result in step 1 above is 16 digits, then the VISA UKPT MAC1 is the first 8 digits (6268 14A7). To compute MAC2 and MAC3, skip to step 8. If the result in step 1 above is 32 digits then perform steps 5, 6, and 7.
  5. Exclusive-Or the rightmost 16 digits of the step 1 result with the step 4 result. Take this result and exclusive-OR with the current MAC Key.
  6. Encrypt the step 5 result with the Current MAC Key.

7. Exclusive-OR Or the step 6 result with the Current MAC Key. This is the VISA UKPT MAC1.
8. Exclusive-Or the result from step 4 or step 8 with the Current MAC Key. In this example the step 4 result is used since step 1 result is 16 digits. 6268 14A7 4C3B C90E exclusive-OR 09F3 B08A 5F38 6D33 = 6B9B A42D 1303 A43D.
9. Encrypt the step 8 result with the Current MAC Key. Encrypt 6B9B A42D 1303 A43D with 09F3 B08A 5F38 6D33 = 05D5 DCBD 42D2 D2B6.
10. Exclusive-Or the step 9 result with the Current MAC Key. 05D5 DCBD 42D2 D2B6 exclusive-OR 09F3 B08A 5F38 6D33 = 0C26 6C37 1DEA BF85. The leftmost 8 digits (0C26 6C37) is MAC2. The rightmost 8 digits (1DEA BF85) is MAC3.

PIN Processing Commands

The remainder of this section contains the command and response syntax for the Network Security Processor PIN processing commands.

Quick Reference

Table 4-10 identifies each command by number, name, and purpose. While Table 4-10 organizes the PIN processing commands by category, the commands themselves are presented in numerical order.

Table 4-10. PIN Processing Commands (page 1 of 2)

Command	Name	Purpose
30	Encrypt PIN	Formats a clear-text PIN in the ANSI PIN Block, and encrypts it under a KPE.
90	Decrypt PIN	Decrypts an incoming PIN block and returns the clear-text PIN.
31	Translate PIN	This command supports a variety of PIN block types. It outputs the PIN in an ANSI PIN block. It also translates the PIN from one key to encryption under another key. Support for Visa DUKPT PIN block requires option 62 to be enabled.
33	Translate PIN	Same as command 31 above, except the outgoing PIN Block is not limited to an ANSI PIN Block.
35	Translate PIN	Same as command 31 above, except the incoming and outgoing PIN block may be double encrypted.
39	Translate PIN and Generate MAC	Translates the PIN using 1key-3DES (single-length) DES keys and Generates a MAC using 1key-3DES (single-length) KMAC key. The outgoing PIN Block type is ANSI.
BA	PIN Translate ANSI to PLUS and Generate MAC	Translates the PIN using 1key-3DES (single-length) DES keys and Generates a MAC using 1key-3DES (single-length) KMAC key. The outgoing PIN Block type is PLUS.
BB	PIN Translate ANSI to PLUS and VerifyMAC	Translates the PIN using 1key-3DES (single-length) DES keys and Verifies a MAC using 1key-3DES (single-length) KMAC key. The outgoing PIN Block type is PLUS.

Table 4-10. PIN Processing Commands (page 2 of 2)

Command	Name	Purpose
BD	Translate PIN and Generate MAC	Translates the PIN and Generates a MAC. The outgoing PIN Block type is ANSI it can be included in the MAC generation process.
335	PIN Translate	Supports multiple incoming and outgoing PIN block types.
32	Verify PIN	Decrypts an incoming PIN and verifies it using a variety of PIN algorithms. Support for the Visa DUKPT requires option 63 to be enabled.
32C	Verify ePIN	Verifies the entered ePIN using the ePIN Object.
36	Verify Double-Encrypted PIN	Decrypts an incoming double-encrypted PIN and verifies it according to the specified PIN algorithm.
37	PIN Change	Decrypts an incoming PIN and verifies the old PIN using a variety of PIN algorithms, and calculates new PVN using the new PIN.
D0	Verify Clear PIN	Verifies a clear PIN using either the Identikey, IBM 3624, or Visa PIN algorithms.
3D	Generate PVN and Offset	Generates an Identikey PVN and IBM 3624 Offset for a PIN and account number.
11E	Generate Atalla 2x2 PVN	Generates an Atalla 2x2 PVN based on clear-text input.
30A	Calculate PIN Offset	Generates a new PIN Offset based on the PIN.
37B	Generate ePIN Offset	Generates an ePIN offset based on the ePIN and PAN.

Encrypt PIN - ANSI Format 0 (Command 30)

Command 30 encrypts a clear-text PIN. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<30#$E_{MFK.1}$(KPE)#PIN#PAN#>

Response

<40#E_{KPE}(ANSI PIN Block)#>[CRLF]

Calling Parameters

30

Field 0, the command identifier.

$E_{MFK.1}$(KPE)

Field 1, the PIN Encryption Key (KPE). This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. If option 6A is enabled, this field can contain a replicated 1key-3DES (single-length) key.

PIN

Field 2, the clear-text PIN. This field contains a numeric value. Option A0 defines the minimum PIN length. The maximum PIN length is 12 digits.

PAN

Field 3, the Primary Account Number (PAN) digits used to form the ANSI PIN block; the 12 rightmost digits, excluding the check digit. This field contains a 12 digit numeric value.

Table 4-11. Command 30: Encrypt PIN (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	30
1	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F

Table 4-11. Command 30: Encrypt PIN (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
2	PIN	4 - 12	0 - 9
3	PAN	12	0 - 9

*Can be a volatile table location.

Responding Parameters

`40`

Field 0, the response identifier.

E_{KPE}`(ANSI PIN Block)`

Field 1, the encrypted ANSI PIN block. This field contains 16 hexadecimal characters.

Table 4-12. Response 40: Encrypt PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier.	2	40
1	E_{KPE}(ANSI PIN Block).	16	0- 9, A - F

Usage Notes

- Generate the PIN Encryption Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Encrypt PIN

- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333 5555 6666 7777 8888. The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F102C2D4DE29C41DE1CF689E9699D6
- PIN: 12345678901
- Twelve rightmost digits of the Primary Account Number excluding the check digit: 000234567890

The command looks like this:

```
<30#47F102C2D4DE29C41DE1CF689E9699D6#12345678901#000234567890
#>
```

The Network Security Processor returns the following response:

```
<40#054935D6E2DA00E2#>
```

Translate PIN (Command 31)

Command 31 translates an encrypted PIN block from encryption under an incoming PIN Encryption Key to an outgoing PIN Encryption Key. The translated PIN block will be in ANSI PIN Block format. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<31#PIN Block Type#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#
E_{KPE_I}(PIN Block)#PIN Block Data#>

Response

<41#E_{KPE_O}(ANSI PIN Block)#Sanity Check Indicator#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

`31`

Field 0, the command identifier.

`PIN Block Type`

Field 1, the incoming PIN block type. This field is 1 byte, it can contain the numbers 1, 2, 3, 4, 5 or 9. When option 46 is enabled, this field can only contain the value 1 (ANSI).

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
IBM 4731	9
VISA DUKPT	See Translate PIN – VISA DUKPT (Command 31)

$E_{MFK.1}(KPE_I)$

Field 2, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}(KPE_O)$

Field 3, the Outgoing PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, an error response is returned if the length of the (KPEo) is not equal to or greater than the length of the (KPEi).

E_{KPEI}(PIN Block)

Field 4, the incoming PIN block encrypted under the Incoming PIN Encryption Key. This field contains a 16 or 18 byte hexadecimal value.

PIN Block Data

Field 5, PIN Block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-13. Command 31: Translate PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	31
1	PIN block type	1	1 - 5, 9
2	$E_{MFK.1}(KPE_I)$*	16,32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16,32	0 - 9, A - F
4	E_{KPEI}(PIN Block)	16, 18	0 - 9, A - F
5	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

41

Field 0, the response identifier.

E_{KPEO}(ANSI PIN Block)

Field 1, the PIN in ANSI PIN block format, encrypted under the Outgoing PIN Encryption Key. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check Indicator`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-14. Response 41: Translate PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	41
1	E_{KPEO}(ANSI PIN Block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L
3	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN in an ANSI PIN Block.

- Clear-text Incoming PIN Encryption Key: 2233 2233 2233 2233.
  The Incoming PIN Encryption Key encrypted under variant 1 of the MFK: 8C2A 7691 A708 A88D.
- Clear-text Outgoing PIN Encryption Key: 4455 4455 4455 4455.
  The Outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 72E7 AEF6 9147 1872.
- Clear-text ANSI PIN block: 0512 AC29 ABCD EFED. The PIN is 12345.
  The encrypted incoming PIN block: 7B58 719B 354B 147A.

- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number excluding the check digit: 9876 5432 1012.

The command looks like this:

```
<31#1#8C2A7691A708A88D#72E7AEF691471872#7B58719B354B147A#
987654321012#>
```

The Network Security Processor returns the following response:

```
<41#06087B12E397F5B6#Y#>
```

Translate PIN – VISA DUKPT (Command 31)

Command 31 – VISA DUKPT translates an ANSI PIN block that is encrypted using a VISA DUKPT session key to an ANSI PIN block encrypted under a single or 2key-3DES (double-length) outgoing PIN Encryption Key.

This command, by default, will generate a 1key-3DES (single-length) session key. Use option A2 and field 7-Algorithm, to control the length of the generated session key.

This command is a standard command and is enabled in the Network Security Processor's default security policy.

Command

<31#7#$E_{MFK.8}$(Derivation Key)#$E_{MFK.1}$(KPE_O)#E_{KPEn}(PIN Block)#
PAN Digits#Key Serial Number#Algorithm#>

Response

<41#E_{KPEO}(ANSI PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

31

Field 0, the command identifier.

7

Field 1, the ANSI PIN block encrypted under a DUKPT key. This field contains a 1 byte decimal value of 7.

$E_{MFK.8}$(Derivation Key)

Field 2, the single or 2key-3DES (double-length) Derivation Key encrypted under variant 8 of the MFK. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to "S".

$E_{MFK.1}$(KPE_O)

Field 3, the Outgoing PIN Encryption Key encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, an error response is returned if the length of the outgoing PIN encryption key is not equal to or greater than the length of the session key used to encrypt the incoming PIN.

E_{KPEn}(PIN Block)

Field 4, the incoming PIN, encrypted using the VISA DUKPT session key management technique. This field contains 16 hexadecimal characters.

PAN Digits

Field 5, the 12 PAN digits used to form the ANSI PIN block. This field contains a 12 byte decimal value.

Key Serial Number

Field 6, the 10 to 20 digit Key Serial Number (KSN) from the PIN pad. This field contains a 10 to 20 byte hexadecimal value.

Algorithm

Field 7, this field is used to determine the length of the session key only when option A2 is set to “B”. With option A2 set to “B” the Network Security Processor will generate a 1key-3DES (single-length) session key when this field contains either a “1” or “S”, and will generate a 2key-3DES (double-length) session key when this field contains the letter “D”.

The Network Security Processor will always generate a 1key-3DES (single-length) session key when option A2 is set to “S”, therefore this field must contain the number “1” or the letter “S”.

The Network Security Processor will always generate a 2key-3DES (double-length) session key when option A2 is set to “D”, therefore this field must contain the number “1” or the letter “D”.

Table 4-15. Command 31: Translate PIN – VISA DUKPT

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	31
1	PIN block type	1	7
2	$E_{MFK.8}$(Derivation key)*	16, 32	0 - 9, A - F
3	$E_{MFK.1}$(KPE_O)*	16, 32	0 - 9, A - F
4	E_{KPEn}(PIN block)	16	0 - 9, A - F
5	PAN digits for ANSI PIN block	12	0 - 9
6	Key serial number to generate current PIN Encryption Key	10 - 20	0 - 9, A - F
7	Algorithm	1	1, S, D

* Can be a volatile table location.

Responding Parameters

`41`

Field 0, the response identifier.

E_{KPEO}(ANSI PIN Block)

Field 1, the PIN formatted in an ANSI PIN block, encrypted under the outgoing PIN Encryption Key. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check Indicator`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-16. Response 41: Translate PIN – VISA DUKPT

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	41
1	E_{KPEO}(ANSI PIN Block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the outgoing PIN Encryption Key and the Derivation key.
- To use this command option 62 must be enabled in the NSP's security policy.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN in a VISA DUKPT PIN block.

- Option A2 is set to "S" or "B" in the Network Security Processor's security policy.
- Clear-text Derivation Key: 1334 1334 1334 1334.
  The Derivation Key encrypted under variant 8 of the MFK: 4A79 F2A0 E61F EECF.

- Clear-text Outgoing PIN Encryption Key: 4455 4455 4455 4455.
  The Outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 72E7 AEF6 9147 1872.
- Clear-text ANSI PIN Block: 0512 AC29 ABCD EFED.
  The DUKPT encrypted PIN block: 8AED F7F9 5963 F4D8.
- PIN block data:
  - Twelve rightmost Primary Account Number digits: 9876 5432 1012.
  - Key serial number: 9876 5432 10E0 0008.
  - PIN encryption key derivation algorithm number: 1.

The command looks like this:

```
<31#7#4A79F2A0E61FEECF#72E7AEF691471872#8AEDF7F95963F4D8#
987654321012#9876543210E00008#1#>
```

The Network Security Processor returns the following response:

```
<41#06087B12E397F5B6#Y#>
```

This example shows the syntax when the option A2 is set to “B” or “S” and field 7 is set to “S”.

```
<31#7#AAA57E4E99AE9B0328F6BA950E1664FA#BC62A2AD72516EA1AE86D4
17E64E07E0#BC14A8602228A412#000234567890#9876543210E00008#S#>
```

The Network Security Processor returns the following response:

```
<41#50DD506F53C3828A#Y#>
```

Translating a PIN in a VISA DUKPT PIN block using a 2key-3DES (double-length) session key.

- Option A2 is set to “B”.
- Clear-text Base Derivation Key: 0123456789ABCDEF FEDCBA9876543210
  The Base Derivation Key encrypted under variant 8 of the MFK: AAA57E4E99AE9B0328F6BA950E1664FA
- Clear-text Outgoing PIN Encryption Key: 4455 4455 4455 4455
  The Outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 72E7AEF691471872

  Clear-text ANSI PIN block: 041274EDCBA9876F. The PIN is 1270.
  Twelve rightmost digits of the Primary Account Number excluding the check digit: 0412 3456 7890. The encrypted incoming PIN block: 7A21BD10F36DC41D.
- PIN block data:
  - Twelve rightmost Primary Account Number digits: 0412 3456 7890.
  - Key serial number: 9876 5432 10E0 0012.
  - PIN encryption key derivation algorithm number: D

The command looks like this:

```
<31#7#AAA57E4E99AE9B0328F6BA950E1664FA#72E7AEF691471872#7A21B
D10F36DC41D#041234567890#9876543210E00012#D#>
```

The Network Security Processor returns the following response:

```
<41#8E3D883AB4FD13A7#Y#>
```

This example shows the syntax when the option A2 is set to “D” and field 7 is set to “1”.

```
<31#7#AAA57E4E99AE9B0328F6BA950E1664FA#72E7AEF691471872#7A21B
D10F36DC41D#041234567890#9876543210E00012#1#>
```

The Network Security Processor returns the following response:

```
<41#8E3D883AB4FD13A7#Y#>
```

Verify PIN – Identikey (Command 32)

Command 32 – Identikey decrypts an incoming encrypted PIN block and verifies it using the Atalla Identikey PIN verification method. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<32#1#PIN Block Type#E_{KPE}(PIN Block)#$E_{MFK.1}$(KPE)#
Bank ID#PVN#Comparison Indicator#Partial PAN#PIN Block Data#>

Response

<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

1

Field 1, the PIN verification method; Identikey.

PIN Block Type

Field 2, incoming PIN block type. This field is 1byte, it can contain the numbers 1, 2, 3, 4, 5, 7 or 9.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
VISA DUKPT	7
IBM 4731	9

E_{KPE}(PIN Block)

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}(KPE)$

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

When the PIN block type is VISA DUKPT (field 2 =7), this field will contain the Derivation Key encrypted under variant 8 of the MFK. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”.

Bank ID

Field 5, the Bank ID; clear-text or encrypted. The clear-text Bank ID is specified by the issuer, it can be a 2, 6, or 8 digit number.

Bank ID	Allowable Size (bytes)
Backward index (algorithm number less than 65)	2
ISO number	6
Route and transfer number	8

The encrypted Bank ID is a 16 hexadecimal character value comprised of the following four data fields ll, bbbbbbbb, p, and cc. It is encrypted under variant 4 of the MFK.

ll - a two-digit number; the length of the Bank ID:

- 02 – The Bank ID in backward index format; the algorithm number must be less than 65.
- 06 – The Bank ID is a six digit ISO number.
- 08 – The Bank ID is an eight digit route-and-transfer number.

bbbbbbbb - The bank ID number (digits 0 - 9); must be the same length as ll.

p - The pad character F, right pads the combined length of the bank ID length (ll) and the bank ID value (bb - bbbbbbbb) resulting in 14 hexadecimal characters. Four pad characters are required when the bank ID is an eight digit value. Six pad characters are required when the bank ID is an six digit value. Ten pad characters are required when the bank ID is a two digit value.

cc - The two hexadecimal character comparison indicator. This field specifies the group (left, middle, or right) of four digits of the six-digit Identikey PIN Verification Number that will be used for the comparison.

- 4C – Compare the leftmost four digits.
- 4D – Compare the middle four digits.
- 52 – Compare the rightmost four digits.

`PVN`

Field 6, the PIN Verification Number. The PVN can be four, six, or eight digits in length, containing the numbers 0 to 7.

`Comparison Indicator`

Field 7, a comparison indicator that specifies which four digits (left, middle, or right) of the six-digit PVN will be compared. This field is 1 byte, and can contain the character L, M, or R. When the PVN is six or eight digits in length or field 5 contains an encrypted bank ID, the value of this field is not evaluated by the Network Security Processor.

`Partial PAN`

Field 8, the portion of the Primary Account Number to be used for verification. This field contains a 4 to 19 byte decimal value.

`PIN Block Data`

Field 9, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-17. Command 32: Verify PIN – Identikey

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	Identikey	1	1
2	PIN block type	1	1 - 5, 7, 9
3	E_{KPE}(PIN block)	16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)* or $E_{MFK.8}$(DK)*	16, 32	0 - 9, A - F
5	Bank ID	2, 6, 8, or 16	0 - 9 or 0 - 9, A - F
6	PIN verification number	4, 6, 8	0 - 7
7	Comparison indicator	1	L, M, R
8	Partial PAN	4 - 19	0 - 9
9	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-18. Response 42: Verify PIN – Identikey

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the PIN Encryption Key.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.
- Generate the Derivation Key when the incoming PIN block is VISA DUKPT.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Identikey PIN Verification - clear-text Bank ID

- Verification method: Identikey (1).
- PIN block type: ANSI (1).
- Clear-text PIN block: 0B12 3454 4CC6 676F.
  The Encrypted PIN block: 48E8 8008 12B0 C9EF.

- Clear-text PIN Encryption Key: 0000 1111 2222 3333.
  The PIN Encryption Key encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Bank ID: 9876 5432.
- PIN verification number: 7532 75.
- Comparison indicator: L (not used).
- Partial PAN: 2345 6789 0.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number excluding the check digit: 0002 3456 7890.

The command looks like this:

```
<32#1#1#48E8800812B0C9EF#47F102C2D4DE29C4#98765432#753275#L#
234567890#000234567890#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Identikey PIN Verification - encrypted Bank ID

This example uses the same data values as shown above.

- Encrypted Bank ID: A1D9408A417D925D

The command looks like this:

```
<32#1#1#48E8800812B0C9EF#47F102C2D4DE29C4#A1D9408A417D925D#
753275#L#234567890#000234567890#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Identikey PIN Verification - DUKPT encrypted PIN block

Option A2 is set to “B”.

- Verification method: Identikey (1).
- PIN block type: VISA DUKPT (7).
- Clear-text PIN block: 0B12 3454 4CC6 676F.
  The encrypted PIN Block: C623 0BAC D0AC 414B.
- Clear-text Derivation Key: 1334 1334 1334 1334 1334 1334 1334 1334.
  The Derivation Key encrypted under variant 8 of the MFK: 4A79 F2A0 E61F EECF 4A79 F2A0 E61F EECF.
- Identikey data:
  - Bank ID: 9876 5432.
  - PIN verification number: 7532 75.

  - Comparison indicator: L.
  - Partial PAN: 2345 6789 0.
- Twelve rightmost digits of the Primary Account Number: 0002 3456 7890.
- Key serial number: 9876 5432 10E0 0001.
- Algorithm: 1.

The command looks like this:

```
<32#1#7#C6230BACD0AC414B#4A79F2A0E61FEECF4A79F2A0E61FEECF#
98765432#753275#L#234567890#000234567890#9876543210E00001#1#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – IBM 3624 (Command 32)

Command 32 – IBM 3624 decrypts an incoming encrypted PIN block and verifies it using the IBM 3624 PIN Verification method. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<32#2#PIN Block Type#E_KPE(PIN Block)#E_MFK.1(KPE)#
Conversion Table#Offset#Validation Data#Pad#
Check-Length#E_MFK.4(KPV)#PIN Block Data#>
```

Response

```
<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]
```

Calling Parameters

`32`

Field 0, the command identifier.

`2`

Field 1, the PIN verification method; IBM 3624.

`PIN Block Type`

Field 2, the incoming PIN block type. This field is 1byte, it can contain the numbers 1, 2, 3, 4, 5, 7 or 9.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
VISA DUKPT	7
IBM 4731	9

E_{KPE}(PIN Block)

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

When the PIN block type is VISA DUKPT (field 2 = 7), this field will contain the Derivation Key encrypted under variant 8 of the MFK. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”.

Conversion Table

Field 5, a table that maps hexadecimal digits (0 through 9, A through F) to decimal digits (0 through 9). This field contains a 16 byte decimal value containing the clear-text Conversion Table or a volatile table location. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK.

When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

Offset

Field 6, an offset value applied to the algorithm-generated PIN before comparing it with the customer-entered PIN. This field contains a 4 to 16 byte decimal value.

Validation Data

Field 7, validation data. This value is unique for each card holder and is typically the account number. This field contains a 4 to 16 byte hexadecimal value. When the PIN block type is ANSI (field 1 = 1) and option 4C is enabled, the value supplied in this field must be 12 digits in length and equal to the PIN Block Data value supplied in field 11.

Pad

Field 8, the pad character used to right-pad the validation data. This field contains a one byte hexadecimal value. The pad character is only used if the validation data is less than 16 digits.

`Check-Length`

Field 9, the check-length. This value is typically the PIN length and determines the number of PIN digits to be compared. This field contains one hexadecimal character in the range of 4 through C.

$E_{MFK.4}$(KPV)

Field 10, the PIN Verification Key (KPV) encrypted under variant 4 of the MFK. This field contains either a 16 or 32 byte hexadecimal value, or a volatile table location.

`PIN Block Data`

Field 11, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types for information on PIN block data.

Table 4-19. Command 32: Verify PIN – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	PIN verification method (IBM 3624)	1	2
2	PIN block type	1	1 - 5, 7, 9
3	E_{KPE}(PIN Block)	16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)* or $E_{MFK.8}$(DK)*	16, 32	0 - 9, A - F
5	Conversion table*	16	0 - 9
6	Offset	4 - 16	0 - 9
7	Validation data	4 - 16	0 - 9, A - F
8	Pad	1	0 - 9, A - F
9	Check-length	1	4 - 9, A - C
10	$E_{MFK.4}$(KPV)*	16, 32	0 - 9, A - F
11	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-20. Response 42: Verify PIN – IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming PIN Encryption Key.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.
- Generate the Derivation Key when the incoming PIN block is VISA DUKPT.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using the IBM 3624 verification method.

- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0936 1436 270E EEEE.
  The encrypted ANSI PIN block: 0558 007D 2156 C394.

- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Conversion table: 8351 2964 7746 1538.
- Offset: 6694 537.
- Validation data: 3333 3333.
- Pad character: D.
- Check-length: 7.
- Clear-text PIN Verification Key (KPV): 89B0 7B35 A1B3 F47E.
  The PIN Verification Key encrypted under variant 4 of the MFK: BB79 3110 FD6D 9BB4.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number: 0000 3331 1111.

The command looks like this:

```
<32#2#1#0558007D2156C394#47F102C2D4DE29C4#8351296477461538#
6694537#33333333#D#7#BB793110FD6D9BB4#000033311111#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – VISA (Command 32)

Command 32 decrypts an incoming encrypted PIN block and verifies it using the VISA Verification Method. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<32#3#PIN Block Type#E_{KPE}(PIN Block)#$E_{MFK.1}$(KPE)#
$E_{MFK.4}$(Key Left)#$E_{MFK.4}$(Key Right)#PVV#PVKI#PAN#
PIN Block Data#>

Response

<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

3

Field 1, the verification method; VISA.

PIN Block Type

Field 2, the incoming PIN type. This field is 1byte, it can contain the numbers 1, 2, 3, 4, 5, 7 or 9.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
VISA DUKPT	7
IBM 4731	9

E_{KPE}(PIN Block)

Field 3, the incoming PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Table 4-21. Command 32: Verify PIN – VISA

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	Verification method (VISA)	1	3
2	PIN block type	1	1 - 5, 7, 9
3	E_{KPE}(PIN Block)	16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)* or $E_{MFK.8}$(DK)*	16, 32	0 - 9, A - F
5	$E_{MFK.4}$(Key Left)*	16	0 - 9, A - F
6	$E_{MFK.4}$(Key Right)*	16	0 - 9, A - F
7	PVV	4	0 - 9
8	PVKI	1	0 - 9
9	PAN	11	0 - 9
10	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-22. Response 42: Verify PIN – VISA

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming PIN Encryption Key.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.
- Generate the PIN Verification Key pair, KL and KR.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using the VISA verification method.

- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0638 7283 FFFF FFFF.
  The ANSI PIN block encrypted under the PIN Encryption Key: 0129 001C E625 BA43.
- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Clear-text Key Left: 4CA2 1616 37D0 133E.
  The Key Left encrypted under variant 4 of the MFK: 026C A1B5 23BE 5DC4.
- Clear-text Key Right: 5E15 1AEA 45DA 2A16.
  The Key Right (KR) encrypted under variant 4 of the MFK: 96D9 3C11 D370 53E2.
- PIN verification value: 3691.
- PIN Verification Key indicator: 3.
- The 11 rightmost digits of the Primary Account Number excluding the check digit: 1234 5678 901.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number: 1234 5678 9019.

The command looks like this:

```
<32#3#1#0129001CE625BA43#47F102C2D4DE29C4#026CA1B523BE5DC4#
96D93C11D37053E2#3691#3#12345678901#123456789019#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – Atalla DES BiLevel (Command 32)

Command 32 – Atalla DES Bilevel decrypts an incoming PIN and verifies it using the Atalla DES Bilevel method. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<32#4#PIN Block Type#E_{KPE}(PIN Block)#$E_{MFK.1}$(KPE)#Bank ID#
Partial PAN#$E_{MFK.4}$(KPV)#PVN-2#PVN-2 Type#PVN-1 Flag#
PVN-2 Start-Compare Flag#PIN Block Data#>

Response

<42#Sanity Check Indicator/Verification Flag#
[IBM Sequence Number#]>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

4

Field 1, the PIN verification method; Atalla DES Bilevel.

PIN Block Type

Field 2, incoming PIN block type. This field is 1 byte, it can contain the numbers 1, 2, 3, 4, 5, 7 or 9.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
VISA DUKPT	7
IBM 4731	9

E_{KPE}(PIN Block)

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

When the PIN block type is VISA DUKPT (field 2 =7), this field will contain the Derivation Key encrypted under variant 8 of the MFK. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”.

Bank ID

Field 5, the bank ID field for the Identikey card issuer. The ID is specified by the issuer, it can be a two-, six-, or eight byte decimal value.

Data Type	Allowable Size (bytes)
Backward index (algorithm number less than 65)	2
ISO number	6
Route and transfer number	8

Partial PAN

Field 6, validation data. This value is unique for each card holder, and in the case of this command, is the partial Primary Account Number (PAN). This field contains a 4 to 19 byte decimal value.

$E_{MFK.4}$(KPV)

Field 7, the PIN Verification Key encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

PVN-2

Field 8, the PIN Verification Number to be verified. This field contains a 4 to 16 byte hexadecimal value.

PVN-2 Type

Field 9, the PVN-2 type. This field indicates whether the PVN-2 should be converted to a decimal value. This field is 1 byte, and contains the numbers 0 or 1. The following table identifies the numerical code for each type of PVN-2.

Action	Code
Convert PVN-2 to a decimal value	0
Don't convert PVN-2; leave it as a hexadecimal value	1

`PVN-1 Flag`

Field 10, a flag that indicates that 8 digits of the PVN-1 value should be used to compute the PVN-2. This field is 1 byte, and contains the number 8.

`PVN-2 Start-Compare Flag`

Field 11, a flag that specifies the starting position within the generated PVN-2 for the comparison. This field is 1 byte, and contains the number 1.

`PIN Block Data`

Field 12, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block

passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-24. Response 42: Verify PIN – Atalla DES Bilevel

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming PIN Verification Key.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.
- Generate the PIN Verification Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using Atalla DES Bilevel.

- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0512 345F FFFF FFFF.
  The ANSI PIN block encrypted under the PIN Encryption Key (KPE): D492 0F0B 1BF0 39F2.
- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Bank ID: 591210.

- Validation data: 5678901.
- Clear-text PIN Verification Key (KPV): ABCD EF01 2345 6789.
  The PIN Verification Key (KPV) encrypted under variant 4 of the MFK: 2BDA 26A1 D559 FF71.
- PVN-2: 6341 6081 3974 3500.
- PVN-2 type: 0.
- PVN-1 flag: 8.
- PVN-2 start-compare flag: 1.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number: 0000 0000 0000.

The command looks like this:

```
<32#4#1#D4920F0B1BF039F2#47F102C2D4DE29C4#591210#5678901#
2BDA26A1D559FF71#6341608139743500#0#8#1#000000000000#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – Diebold (Command 32)

Command 32 – Diebold decrypts an incoming encrypted PIN block and verifies it using the Diebold PIN Verification method. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<32#5#PIN Block Type#E_{KPE}(PIN Block)#$E_{MFK.1}$(KPE)#
Validation Data#Offset#Algorithm Number#
Diebold Key Table Location#PIN Block Data#>

Response

<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

5

Field 1, the PIN verification method; Diebold.

PIN Block Type

Field 2, incoming PIN block type. This field is 1 byte, it can contain the numbers 1, 2, 3, 4, 5, 7 or 9. When option 46 is enabled, this field can only contain the value 1 (ANSI).

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
VISA DUKPT	7
IBM 4731	9

E_{KPE}(PIN Block)

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

When the PIN block type is VISA DUKPT (field 2 =7), this field will contain the Derivation Key encrypted under variant 8 of the MFK. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”.

Validation Data

Field 5, validation data. This value is unique for each card holder, and in the case of this command, is the Primary Account Number (PAN). This field contains a 4 to 19 byte decimal value.

Offset

Field 6, an offset value applied to the algorithm-generated PIN before comparing it with the customer-entered PIN. This field contains a 4 byte decimal value.

Algorithm Number

Field 7, the Diebold algorithm number. This field is 2 byte decimal value.

Diebold Key Table Location

Field 8, the index to the first volatile table location where the Diebold number table is stored. Thirty-two contiguous table locations hold the Diebold number table. This field contains a 1 to 4 byte decimal value.

PIN Block Data

Field 9, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-25. Command 32: Verify PIN – Diebold (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	PIN verification method (Diebold)	1	5
2	PIN block type	1	1 - 5, 7, 9
3	E_{KPE}(PIN block)	16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)* or $E_{MFK.8}$(DK)*	16, 32	0 - 9, A - F

Table 4-25. Command 32: Verify PIN – Diebold (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
5	Validation data	4 - 19	0 - 9
6	Offset	4	0 - 9
7	Algorithm number	2	0 - 9
8	Diebold key table location	1 - 4	0 - 9
9	PIN block data.**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.
- INVALID NUMBER TABLE – Diebold number table is invalid (it is empty or contains data other than the Diebold number table). This error usually indicates that the Diebold Number Table was not properly loaded into the volatile table.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-26. Response 42: Verify PIN – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1, 15	Y, N, S, L, INVALID NUMBER TABLE
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Preload the Diebold number table using thirty-two command 74s.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.
- By default, this command processes the leftmost four PIN digits. Enable option 027 to process the rightmost four PIN digits.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

This example uses a specific Diebold Number Table your test results will be different.

Verifying an encrypted ANSI PIN block using the Diebold verification method.

- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0464 56ED CB4 876F.
  The ANSI PIN block encrypted under the PIN Encryption Key: AFC5 C290 4C92 2280.
- Clear-text PIN Encryption Key (KPE): 0123 4567 89AB CDEF.
  The PIN Encryption Key encrypted under variant 1 of the MFK: AE86 D417 E64E 07E0.
- Validation data: 1234 5678 90.
- Offset: 0000.
- Algorithm number: 82.
- Diebold key table location: 250.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number: 0012 3456 7890.

The command looks like this:

```
<32#5#1#AFC5C2904C922280#AE86D417E64E07E0#1234567890#0000#82#
250#001234567890#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – NCR (Command 32)

Command 32 – NCR decrypts an incoming encrypted PIN block and verifies it using the NCR method of verification. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<32#6#PIN Block Type#E_{KPE}(PIN Block)#$E_{MFK.1}$(KPE)#
Conversion Table#Offset#Validation Data#Pad#PLEN#$E_{MFK.4}$(KPV)#
Padding Flag#Counting Flag#Start-Count Position#
Select-PLEN Position#PIN Block Data#>

Response

<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

6

Field 1, the PIN verification method; NCR.

PIN Block Type

Field 2, incoming PIN block type. This field is 1byte, it can contain the numbers 1, 2, 3, 4, 5, 7 or 9.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN p	
IBM encrypming PIN p	
s	
ISA DUKtPTI	

E_{KPE}(PIN Block)

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

When the PIN block type is VISA DUKPT (field 2 =7), this field will contain the Derivation Key encrypted under variant 8 of the MFK. This key should be a 2key-3DES (double-length) key. It can be a 1key-3DES (single-length) key only if option A2 is set to “S”.

Conversion Table

Field 5, a table that maps hexadecimal digits (0 through 9, A through F) to decimal digits (0 through 9). This field contains a 16 byte decimal value containing the clear-text Conversion Table or a volatile table location. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK.

When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

Offset

Field 6, an offset value applied to the algorithm-generated PIN before comparing it with the customer-entered PIN. This field contains a 4 to 12 byte decimal value.

Validation Data

Field 7, validation data. This value is unique for each card holder, and in the case of this command, is the partial Primary Account Number (PAN). This field contains a 4 to 16 byte hexadecimal value. When the PIN block type is ANSI (field 1 = 1) and option 4C is enabled, the value supplied in this field must be 12 digits in length and equal to the PIN Block Data value supplied in field 15.

Pad

Field 8, a pad character used to fill out the partial PAN. This field contains a one byte hexadecimal value.

PLEN

Field 9, the number of contiguous PIN digits selected for verification; the PIN length, or PLEN. This field contains a one byte number that can contain the numbers 4 to 9 and the characters A, B, and C.

$E_{MFK.4}$(KPV)

Field 10, the PIN Verification Key (KPV) encrypted under variant 4 of the MFK. This field contains either a 16 or 32 byte hexadecimal value, or a volatile table location.

Padding Flag

Field 11, a flag that indicates whether the validation data (Field 7) is to be padded on the left or on the right. This field is 1 byte, and contains the character L or R.

Counting Flag

Field 12, a flag that indicates whether the counting scheme for selecting the PIN digit for verification is left or right. This field is 1 byte, and contains the character L or R.

Start-Count Position

Field 13, the field that indicates the starting position for the counting scheme measured from either the left or right of the entered PIN depending on field 12. This field is one byte, it can contain a number in the range of 1-9.

Select-PLEN Position

Field 14, the field that indicates the beginning position (from the direction of the counting flag, starting with 0) for selecting PLEN characters from the output of the decimalization step. This field contains a one byte hexadecimal value. If the counting flag is “L”, the leftmost digit of the decimalized result is position zero. If the counting flag is “R,” the rightmost digit of the decimalized result is position zero. This field is one byte, it can contain a character in the range of 0-9, A-C.

PIN Block Data

Field 15, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-27. Command 32: Verify PIN – NCR (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	PIN verification method (NCR)	1	6
2	PIN block type	1	1 - 5, 7, 9
3	E_{KPE}(PIN block)	16, 18	0 - 9, A - F

Table 4-27. Command 32: Verify PIN – NCR (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	$E_{MFK.1}$(KPE)* or $E_{MFK.8}$(DK)*	16, 32	0 - 9, A - F
5	Conversion table*	16	0 - 9,
6	Offset	4 - 16	0 - 9
7	Validation data	4 - 16	0 - 9, A - F
8	Pad	1	0 - 9, A - F
9	PLEN	1	4 - 9, A - C
10	$E_{MFK.4}$(KPV)*	16, 32	0 - 9, A - F
11	Padding flag	1	L, R
12	Counting flag	1	L, R
13	Start-count position	1	1 - 9
14	Select-PLEN position	1	0 - 9, A - C
15	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-28. Response 42: Verify PIN – NCR

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming PIN Encryption Key.
- Generate the PIN Verification Key.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using the NCR verification method.

- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0413 25FF FFFF FFFF.
  The ANSI PIN block encrypted under the PIN Encryption Key: 9A9C 37BF 6B38 8736.
- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Conversion table: 0123 4567 8901 2345.
- Offset: 0000.
- Validation data: 2700 4552 4000 0121.
- Pad character: F.
- PLEN: 4.
- Clear-text PIN Verification Key (KPV): 68BA 0794 F140 641C.
  The PIN Verification Key encrypted under variant 4 of the MFK: FE87 4532 1894 0916.
- Padding flag: R.
- Counting flag: L.
- Start-count position: 1.

- Select-PLEN position: 6.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number: 0455 2400 0012.

The command looks like this:

```
<32#6#1#9A9C37BF6B388736#47F102C2D4DE29C4#0123456789012345#
0000#2700455240000121#F#4#FE87453218940916#R#L#1#6#
045524000012#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – Clear-PIN Comparison (Command 32)

Command 32 – Clear-PIN comparison decrypts an incoming encrypted PIN block and verifies it against the expected Clear-PIN value. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure and is not enabled in the Network Security Processor's default security policy. You must purchase option 60 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

```
<32#7#PIN Block Type#E_KPE(PIN Block)#E_MFK.1(KPE)#
Clear-Text PIN#PIN Block Data#>
```

Response

```
<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]
```

Calling Parameters

`32`

Field 0, the command identifier.

`7`

Field 1, the PIN verification method; Clear-PIN comparison.

`PIN Block Type`

Field 2, the incoming PIN block type. This field is one byte long it can contain the numbers 1, 2, 3, 4, 5 or 9.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
IBM 4731	9

E_{KPE}`(PIN Block)`

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key (KPE) encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

Clear-Text PIN

Field 5, the clear-text PIN. This value will be compared to the PIN in the incoming PIN block. This field contains a 0 to 12 byte decimal value.

PIN Block Data

Field 6, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-29. Command 32: Verify PIN – Clear-PIN Comparison

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	PIN verification method	1	7
2	PIN block type	1	1 - 5, 9
3	E_{KPE}(PIN block)	16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	Clear-text PIN	0 - 12	0 - 9
6	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

42

Field 0, the response identifier.

Sanity Check Indicator/Verification Flag

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.

- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-30. Response 42: Verify PIN – Clear-PIN Comparison

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- This command supports PINs of length zero to twelve characters.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using the Clear-PIN comparison verification method.

- Verification method: Clear-PIN comparison (7).
- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0B12 3454 4CC6 676F.
  The ANSI PIN block encrypted under the PIN Encryption Key: 48E8 8008 12B0 C9EF.
- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Clear-text PIN: 1234 5678 901.
- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number: 0002 3456 7890.

The command looks like this:

```
<32#7#1#48E8800812B0C9EF#47F102C2D4DE29C4#12345678901#
000234567890#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – PIN-Block Comparison (Command 32)

Command 32 – PIN-block comparison decrypts two incoming encrypted PIN blocks and compares the clear-text PIN blocks. This command supports only 1key-3DES (single-length) working keys.

This command has a high security exposure. It is not enabled in the Network Security Processor's default security policy. You must purchase option 61 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<32#8#E_{KPE1}(PIN Block1)#$E_{MFK.1}$(KPE1)#E_{KPE2}(PIN Block2)#
$E_{MFK.1}$(KPE2)#>

Response

<42#Verification Flag#>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

8

Field 1, the PIN verification method; PIN-block comparison.

E_{KPE1}(PIN Block1)

Field 2, the first incoming PIN block encrypted under the first PIN Encryption Key (KPE1). This field contains 16 hexadecimal characters.

$E_{MFK.1}$(KPE1)

Field 3, the first PIN Encryption Key encrypted under variant 1 of the MFK. This key is used to encrypt the first incoming PIN block. This field must be a 16 byte value, or a volatile table location.

E_{KPE2}(PIN Block2)

Field 4, the second incoming encrypted PIN block encrypted under the second PIN Encryption Key (KPE2). This field contains 16 hexadecimal characters.

$E_{MFK.1}$(KPE2)

Field 5, the second PIN Encryption Key (KPE2) encrypted under variant 1 of the MFK. This key is used to encrypt the second incoming PIN block. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 4-31. Command 32: Verify PIN – PIN-Block Comparison

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	Verification method	1	8
2	E_{KPE1}(PIN Block1)	16	0 - 9, A - F
3	$E_{MFK.1}$(KPE1).*	16	0 - 9, A - F
4	E_{KPE2}(PIN Block2)	16	0 - 9, A - F
5	$E_{MFK.1}$(KPE2)*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

42

Field 0, the response identifier.

Verification Flag

Field 1, the verification flag. Starting with the leftmost position, the Network Security Processor scans the decrypted PIN blocks. The scan stops when a non-numeric character is encountered. The numeric digit of both PIN blocks are compared. Based on the comparison result, this field will contain one of the following values:

- Y – both PIN blocks are the same.
- N – both PIN blocks are not the same, or the first character in either PIN block is not a digit.
- S – there are more than 12 digits in one or both of the PIN blocks.

Table 4-32. Response 42: Verify PIN – PIN-Block Comparison

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Verification flag	1	Y, N, S

Usage Notes

- Generate the PIN Encryption Keys.

- This command does not check the entire PIN to be sure its length is legal. This command compares two 16 character strings.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying Different PIN Blocks That Contain the Same PIN

- Clear-text PIN Block1: 9999 9999 F103 3465.
  The PIN Block encrypted under the KPE-1: 2D67 26EC DBCD EC3B.
- Clear-text PIN Encryption Key-1: 634A 00F7 8F96 3784.
  The PIN Encryption Key-1 encrypted under variant 1 of the MFK: B427 A68B 8218 8A76.
- Clear-text PIN Block-2: 9999 9999 AAAA AAAA.
  The PIN Block-2 encrypted under the KPE-2: 3F84 347A 3857 1B13.
- Clear-text PIN Encryption Key-2: FEDC BA98 7654 3210.
  The PIN Encryption Key-2 encrypted under variant 1 of the MFK: BC62 A2AD 7251 6EA1.

The command looks like this:

```
<32#8#2D6726ECDBCDEC3B#B427A68B82188A76#3F84347A38571B13#
BC62A2AD72516EA1#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – Burroughs (Command 32)

Command 32 – Burroughs decrypts an incoming encrypted PIN block and verifies it using the Burroughs method of verification. This command supports single or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command is enabled in the Network Security Processor's default security policy.

Command

<32#F#PIN Block Type#E_{KPE}(PIN Block)#$E_{MFK.1}$(KPE)#PAN#
SECPD#SECTYPE#Offset#$E_{MFK.5}$(Table1 Line0)#
$E_{MFK.5}$(Table1 Line1)#$E_{MFK.5}$(Table2 Line0)#
$E_{MFK.5}$(Table2 Line1)#PIN Block Data#>

Response

<42#Sanity Check Indicator/Verification Flag#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

F

Field 1, the PIN verification method; Burroughs.

PIN Block Type

Field 2, the incoming PIN block type. This field is 1 byte, it contains the numbers 1, 2 or 3.

E_{KPE}(PIN Block)

Field 3, the encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

PAN

Field 5, the Primary Account Number (PAN) to be used for verification. This field is 16 to 19 digits long.

SECPD

Field 6, Security Period. This field contains a 1 byte decimal value.

SECTYPE

Field 7, Security Method Character. This field contains a 1 byte decimal value.

Offset

Field 8, an offset value applied to the algorithm-generated PIN before comparing it with the customer-entered PIN. This field contains a 4 byte decimal value.

$E_{MFK.5}$(Table1 Line0)

Field 9, the first row of the first lookup table encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.5}$(Table1 Line1)

Field 10, the second row of the first lookup table encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.5}$(Table2 Line0)

Field 11, the first row of the second lookup table encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.5}$(Table2 Line1)

Field 12, the second row of the second lookup table encrypted under variant 5 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

PIN Block Data

Field 13, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-33. Command 32: Verify PIN – Burroughs (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	Burroughs	1	F
2	PIN block type	1	1, 2, or 3
3	E_{KPE}(PIN block)	16, 18	0 - 9, A - F

Table 4-33. Command 32: Verify PIN – Burroughs (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	PAN	16-19	0 - 9,
6	SECPD	1	0 - 9
7	SECTYPE	1	0 - 9
8	Offset	4	0 - 9
9	$E_{MFK.5}$(Table1 Line0)*	16	0 - 9, A - F
10	$E_{MFK.5}$(Table1 Line1)*	16	0 - 9, A - F
11	$E_{MFK.5}$(Table2 Line0)*	16	0 - 9, A - F
12	$E_{MFK.5}$(Table2 Line1)*	16	0 - 9, A - F
13	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 2, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-34. Response 42: Verify PIN – Burroughs

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L
2	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming PIN Encryption Key.
- Generate the table cryptograms.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using the Burroughs verification method.

- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0445 62FD 79FF FFFF
  The ANSI PIN block encrypted under the PIN Encryption Key (KPE): 47EB F9B3 877D B5C8.
- Clear-text PIN Encryption Key (KPE): 0000 1111 2222 3333.
  The PIN Encryption Key (KPE) encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- PAN: 0010 0006 0286 0000 00.
- SECPD: 0.
- SECTYPE: 0
- Offset: 0000.
- Clear-text Table 1, Line 0: D7A9 E2FB 6834 05C1.
  Table 1, Line 0 encrypted under variant 5 of the MFK: A101 CCC8 4435 8924.
- Clear-text Table 1, Line 1: 0000 0000 0000 0000.
  Table 1, Line 1 encrypted under variant 5 of the MFK: A5DE 5A32 F809 86F7.
- Clear-text Table 2, Line 0: C8B9 D1F2 A06E 5734.
  Table 2, Line 0 encrypted under variant 5 of the MFK: B88E 92EC 01C6 BA34.
- Clear-text Table 1, Line 1: 0000 0000 0000 0000.
  Table 2, Line 1 encrypted under variant 5 of the MFK: A5DE 5A32 F809 86F7.

- PIN block data; in this case, the 12 rightmost digits of the Primary Account Number (PAN): 0602 8600 0000.

The command looks like this:

```
<32#F#1#47EBF9B3877DB5C8#47F102C2D4DE29C4#0010000602860000000#
0#0#0000#A101CCC844358924#A5DE5A32F80986F7#B88E92EC01C6BA34#
A5DE5A32F80986F7#060286000000#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Verify PIN – Atalla 2x2 (Command 32)

Command 32 – Atalla 2x2 verifies an encrypted ANSI PIN block using the Atalla 2x2 algorithm. The PIN Encryption key can be either single or 2key-3DES (double-length), the PIN Verification Keys must be 1key-3DES (single-length).

This command is enabled in the Network Security Processor's default security policy.

Command

<32#I#PIN Block Type#E_{KPE}(PIN Block)#
$E_{MFK.1}$(PIN Encryption Key)#$E_{MFK.4}$(PIN Verification Key1)#
$E_{MFK.4}$(PIN Verification Key2)#PVN Format#PVN#PAN Digits#>

Response

<42#Sanity Check Indicator/Verification Flag#>[CRLF]

Calling Parameters

32

Field 0, the command identifier.

I

Field 1, the verification method; Atalla 2x2.

PIN Block Type

Field 2, incoming PIN block is ANSI. This field contains the value 1.

E_{KPE}(PIN Block)

Field 3, the ANSI PIN block encrypted under the PIN Encryption Key (KPE). This field contains 16 hexadecimal characters.

$E_{MFK.1}$(KPE)

Field 4, the Incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.4}$(PIN Verification Key1)

Field 5, the PIN Verification Key 1 encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.4}$(PIN Verification Key2)

Field 6, the PIN Verification Key 2 encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

PVN Format

Field 7, specifies the format of the PVN. The choices are hexadecimal or decimal. This field should contain the letter H for hexadecimal format. For decimal format this field should contain the letter D, followed by the 16 byte decimalization table. If you use the default decimalization table of 0123456789012345, this field will contain only the letter D.

PVN

Field 8, the PIN Verification Number to be compared against the computed result. This field contains a 6 to 16 byte hexadecimal value.

PAN Digits

Field 9, the Primary Account Number digits used in the algorithm to generate the PVN. This field contains a 12 byte decimal value.

Table 4-35. Command 32: Verify PIN –Atalla 2x2

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	32
1	Verification method	1	I
2	PIN block type	1	1
3	E_{KPE}(PIN Block)	16	0 - 9, A - F
4	$E_{MFK.1}$(PIN Encryption Key)*	16, 32	0 - 9, A - F
5	$E_{MFK.4}$(PIN Verification Key1)*	16	0 - 9, A - F
6	$E_{MFK.4}$(PIN Verification Key2)*	16	0 - 9, A - F
7	PVN Format	1, 17	H, or D and 0 - 9
8	PVN	6-16	0 - 9, A - F
9	PAN Digits	12	0 - 9

*Can be a volatile table location.

Responding Parameters

`42`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the PIN length is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-36. Response 42: Verify PIN – Atalla 2x2

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	42
1	Sanity check indicator/verification flag	1	Y, N, S, L

Usage Notes

- Generate the incoming PIN Encryption Key
- Generate the PIN Verification Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an encrypted ANSI PIN block using the Atalla 2x2 method.

- Verification method: Atalla 2x2 (I).
- PIN block type: ANSI (1).
- Clear-text ANSI PIN block: 0655 476B EDCB EDCB. The PIN is 555555.
  The encrypted PIN Block: 661A B611 2C5E B5A0.
- Clear-text PIN Encryption Key: 0000 1111 2222 3333.
  The PIN Encryption Key encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.

- Clear-text PIN Verification Key 1: 5555 6666 7777 8888.
  The PIN Verification Key 1 encrypted under variant 4 of the MFK: 953D 33E5 1F16 C884.
- Clear-text PIN Verification Key 2: 9999 AAAA BBBB CCCC.
  The PIN Verification Key 2 encrypted under variant 4 of the MFK: 9950 6F9B 9A69 E03F.
- Hexadecimal Format
- PVN: 3436 593F 00F3 C754.
- Twelve Primary Account Number digits: 1234 1234 1234.

The command looks like this:

```
<32#I#1#661AB6112C5EB5A0#47F102C2D4DE29C4#953D33E51F16C884#
99506F9B9A69E03F#H#3436593F00F3C754#123412341234#>
```

The Network Security Processor returns the following response:

```
<42#Y#>
```

Translate PIN – ANSI to PLUS and PLUS to ANSI (Command 33)

Command 33 – ANSI to PLUS and PLUS to ANSI. This command translates an encrypted PIN block from incoming encryption in an ANSI PIN block to outgoing encryption in the PLUS PIN block, or from incoming encryption in the PLUS PIN block to outgoing encryption in an ANSI PIN block. Both ANSI and PLUS use the same PIN block format. The PLUS PIN block requires the leftmost 12 account number digits, whereas the ANSI PIN block requires the rightmost 12 account number digits excluding the check digit. The incoming PIN Encryption key is designated as KPE_I and the outgoing PIN Encryption Key is designated as KPE_O. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<33#11#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming PAN Digits#Outgoing PAN Digits#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

11

Field 1, the PIN translation method; in this command, both the input and output PIN blocks have the same format, only the account number digits may be different.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. If this key is 2key-3DES (double-length), the key values for key 1 and key 2 must be not be the same. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Incoming PAN Digits

Field 5, the Primary Account Number (PAN) digits used in the incoming PIN block; the 12 leftmost digits for PLUS or the 12 rightmost digits, excluding the check digit, for ANSI. This field contains a 12 byte decimal value. When either option 46 or 47 is enabled, the value of this field and field 6 must be identical.

Outgoing PAN Digits

Field 6, the Primary Account Number (PAN) digits used in the outgoing PIN block; the 12 leftmost digits for PLUS or the 12 rightmost digits, excluding the check digit, for ANSI. This field contains a 12 byte decimal value.

Table 4-37. Command 33: Translate PIN – ANSI to PLUS, PLUS to ANSI

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN block Type	2	11
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming PAN digits	12	0 - 9
6	Outgoing PAN digits	12	0 - 9

*Can be a volatile table location.

Responding Parameters

43

Field 0, the response identifier.

E_{KPE_O}(PIN Block)

Field 1, the outgoing encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

Sanity Check Indicator

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-38. Response 43: Translate PIN – ANSI to PLUS, PLUS to ANSI

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an ANSI PIN block to PLUS PIN block.

- Clear-text incoming PIN Encryption Key (KPE_I): 07CE A74F 4607 5D8F 0000 1111 2222 3333.
  The incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1 47F1 02C2 D4DE 29C4.
- Clear-text outgoing PIN Encryption Key (KPE_O): D029 23D9 AD4F E90B 5555 6666 7777 8888.
  The outgoing PIN Encryption Key (KPE_O) encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F 1DE1 CF68 9E96 99D6.
- Clear-text PIN block: 0453 55F8 BEF7 EBBA.
  The PIN block encrypted under the incoming PIN Encryption Key: F4DB 98CB C7D2 DC14.
- Incoming PAN digits: 1207 4108 1445.
- Outgoing PAN digits: 2074 1081 4457.

The command looks like this:

```
<33#11#3B42CA4278E2DDE147F102C2D4DE29C4#83CBEFA710C6639F1DE1C
F689E9699D6#F4DB98CBC7D2DC14#120741081445#207410814457#>
```

The Network Security Processor returns the following response:

```
<43#CBC0F5BC0ED28BBD#Y#>
```

Translate PIN – ANSI to PIN/Pad (Command 33)

Command 33 – ANSI to PIN/pad. This command translates an encrypted ANSI PIN block to an encrypted PIN/pad character PIN block. The incoming PIN Encryption key is designated as KPE_I and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#13#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#Pad#
PAN Digits#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

13

Field 1, the PIN translation method; in this command, ANSI to PIN pad.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

`Pad`

Field 5, the pad character in the PIN pad block. This field is 1 byte, it can contain a hexadecimal value, X or W. When this field contains the value X or W, the character F will be used as the pad character.

`PAN Digits`

Field 6, the Primary Account Number (PAN) digits used in the incoming ANSI PIN block. This field contains a 12 byte decimal value.

Table 4-39. Command 33: Translate PIN – ANSI to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (ANSI to PIN pad)	2	13
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Pad	1	0 - 9, A - F, X, W
6	PAN digits	12	0 - 9

*Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response identifier.

E_{KPE_O}`(PIN Block)`

Field 1, the outgoing, encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check Indicator`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.

- L – the length of the PIN is out of range.

Table 4-40. Response 43: Translate PIN – ANSI to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an ANSI PIN block to PIN/pad character PIN block.

- Clear-text incoming PIN Encryption Key (KPE_I): 0123456789ABCDEF 0000111122223333.
  The incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK: AE86D417E64E07E047F102C2D4DE29C4.
- Clear-text outgoing PIN Encryption Key (KPEO): 4567 ABCD EF12 3890 5555 6666 7777 8888.
  The outgoing PIN Encryption Key (KPEO) encrypted under variant 1 of the MFK: 5E970C0BFB49402C1DE1CF689E9699D6.
- Clear-text incoming PIN block: 045355F8BEF7EBBA.
  The incoming PIN block encrypted under the incoming PIN Encryption Key (KPE_I): 2299CD5D3804E247.
- Outgoing Pad character: D.
- Incoming PAN: 120741081445.

The command looks like this:

```
<33#13#AE86D417E64E07E047F102C2D4DE29C4#5E970C0BFB49402C1DE1C
F689E9699D6#2299CD5D3804E247#D#120741081445#>
```

The Network Security Processor returns the following response:

```
<43#D3F5F0561FCAAE78#Y#>
```

Translate PIN – ANSI to IBM 4731 (Command 33)

Command 33 – ANSI to IBM 4731. This command translates an encrypted PIN block from incoming encryption in the ANSI PIN block to outgoing encryption in the IBM 4731 PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#19#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming PAN#Outgoing Pad#Outgoing ICV#$E_{MFK.3}$(KC)#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

19

Field 1, the PIN translation method; in this command, ANSI to IBM 4731.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

`Incoming PAN`

Field 5, the Primary Account Number (PAN) used in the incoming PIN block; the 12 rightmost digits, excluding the check digit. This field contains a 12 byte decimal value.

`Outgoing Pad`

Field 6, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal character or the letters X or W. When this field contains either X or W, the pad character in the incoming PIN block will also be used as the outgoing pad character.

`Outgoing ICV`

Field 7, the sequence number for the outgoing PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.3}(KC)$

Field 8, the Communications Key encrypted under variant 3 of the MFK. This key is used in the outer or second encryption of the IBM 4731 PIN block for the outgoing PIN block. This field contains a 16 byte hexadecimal value or a volatile table location.

Table 4-41. Command 33: Translate PIN – ANSI to IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (ANSI to IBM 4731)	2	19
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming PAN	12	0 - 9
6	Outgoing Pad	1	0 - 9, A - F, X, W
7	Outgoing ICV	16	0 - 9, A - F
8	$E_{MFK.3}(KC)$*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response identifier.

E_{KPE_O}(PIN Block)

Field 1, the outgoing, encrypted PIN. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

Sanity Check Indicator

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-42. Response 43: Translate PIN – ANSI to IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate incoming and outgoing PIN Encryption Keys and the Communications Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an ANSI PIN block to an IBM 4731 PIN block.

- Clear-text incoming PIN Encryption Key (KPE_I): 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- Clear-text outgoing PIN Encryption Key (KPE_O): D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key (KPE_O) encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- Incoming PAN: 1207 4108 1445.
- Outgoing Pad character: D.
- Sequence Number (ICV) 1234 1234 1234 1234.

- Clear-text Communications Key: B302 AD91 F504 EA22.
  The Communications Key encrypted under variant 3 of the MFK: FFFF FFFF FFFF FFFF.

The command looks like this:

```
<33#19#3B42CA4278E2DDE1#83CBEFA710C6639F#5196681F910C408C#
120741081445#D#1234123412341234#FFFFFFFFFFFFFFFF#>
```

The Network Security Processor returns the following response:

```
<43#27682B863CD388E8#Y#>
```

Translate PIN – IBM 3624 to IBM 3624 (Command 33)

Command 33 – IBM 3624 to IBM 3624. This command translates an encrypted IBM 3624 PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. The incoming Communications Key is designated as KC_I, and the outgoing Communications Key is designated as KC_O. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#22#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming Pad#$E_{MFK.2}$(KC$_I$)#Outgoing Pad#$E_{MFK.2}$(KC$_O$)#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#
IBM 3624 Sequence Number#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

22

Field 1, the PIN translation method; in this command, IBM 3624 to IBM 3624.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains an 18 byte hexadecimal value.

Incoming Pad

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

$E_{MFK.2}(KC_I)$

Field 6, the incoming Communications Key encrypted under variant 2 of the MFK. This key is used in the outer, or second, encryption of the IBM 3624 PIN block for the incoming PIN. This field contains a 16 byte hexadecimal value, or a volatile table location.

Outgoing Pad

Field 7, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal value, X, or W. The value X or W indicates that the pad character for the incoming PIN block will also be used as the outgoing pad character.

$E_{MFK.2}(KC_O)$

Field 8, the outgoing Communications Key encrypted under variant 2 of the MFK. This key is used in the outer, or second, encryption of the IBM 3624 PIN block for the outgoing PIN block. This field contains a 16 byte hexadecimal value or a volatile table location.

Table 4-43. Command 33: Translate PIN – IBM 3624 to IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN block translation method (IBM 3624 to IBM 3624)	2	22
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPEI}(PIN block)	18	0 - 9, A - F
5	Incoming pad	1	0 - 9, A - F, X, W
6	$E_{MFK.2}(KC_I)$*	16	0 - 9, A - F
7	Outgoing pad	1	0 - 9, A - F, X, W
8	$E_{MFK.2}(KCO)$*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response indicator.

E_{KPE_O}`(PIN Block)`

Field 1, the outgoing encrypted PIN. This field is an 18 byte hexadecimal value. When a PIN sanity error is detected, the value in this field may not be correct.

`Sanity Check Indicator`

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`IBM 3624 Sequence Number`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-44. Response 43: Translate PIN – IBM 3624 to IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	43
1	E_{KPE_O}(PIN block)	18	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L
3	IBM 3624 sequence number	2	0 - 9, A - F

Usage Notes

- Generate the incoming and the outgoing PIN Encryption Keys and Communications Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an IBM 3624 PIN block to IBM 3624 PIN block.

- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- Encrypted incoming PIN block: 9864 AB86 5904 8084 B8.
- Incoming pad character: B.
- Clear-text incoming Communications Key: A15D BAFD F119 F701.
  The incoming Communications Key encrypted under variant 2 of the MFK: 306D0D8C8A2E6414.
- Outgoing pad character: D.
- Clear-text outgoing Communications Key: F72B 85D0 302D 448A.
  The outgoing Communications Key encrypted under variant 2 of the MFK: 1646 F963 48BD 4800.

The command looks like this:

```
<33#22#3B42CA4278E2DDE1#83CBEFA710C6639F#9864AB8659048084B8#
B#306D0D8C8A2E6414#D#1646F96348BD4800#>
```

The Network Security Processor returns the following response:

```
<43#843322E77167AE5384#Y#99#>
```

Translate PIN – IBM 3624 to PIN/Pad (Command 33)

Command 33 – IBM 3624 to PIN/pad. This command translates an incoming encrypted PIN block in the IBM 3624 PIN block to outgoing encryption in PIN/pad character PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#23#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming Pad#$E_{MFK.2}$(KC)#Outgoing Pad#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#
IBM 3624 Sequence Number#> [CRLF]

Calling Parameters

33

Field 0, the command identifier.

23

Field 1, the PIN translation method; in this command, IBM 3624 to PIN/pad.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Incoming Pad

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

$E_{MFK.2}$(KC)

Field 6, the incoming Communications Key encrypted under variant 2 of the MFK. This key is used in the outer, or second, encryption of the IBM 3624 PIN block for the incoming PIN. This field contains 16 hexadecimal characters.

Outgoing Pad

Field 7, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal value, X, or W. The values X or W indicate that the pad character in the incoming PIN block will also be used as the outgoing pad character.

Table 4-45. Command 33: Translate PIN – IBM 3624 to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (IBM 3624 to PIN/pad)	2	23
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN Block)	18	0 - 9, A - F
5	Incoming pad	1	0 - 9, A - F, X, W
6	$E_{MFK.2}$(KC)*	16	0 - 9, A - F
7	Outgoing pad	1	0 - 9, A - F, X, W

*Can be a volatile table location.

Responding Parameters

43

Field 0, the response identifier.

E_{KPE_O}(PIN Block)

Field 1, the outgoing encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

`Sanity Check Indicator`

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`IBM 3624 Sequence Number`

Field 3, the IBM 3624 sequence number. This field contains 2 hexadecimal characters.

Table 4-46. Response 43: Translate PIN – IBM 3624 to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPEO}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L
3	IBM 3624 sequence number	2	0 - 9, A - F

Usage Notes

- Generate the PIN encryption Keys and Communications Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an IBM 3624 PIN block to PIN/pad PIN block.

- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- Encrypted incoming PIN block: 9864 AB86 5904 8084 B8.
- Incoming pad character: B.

- Clear-text Communications Key: A15D BAFD F119 F701.
  The Communications Key encrypted under variant 2 of the MFK: 306D 0D8C 8A2E 6414.
- Outgoing pad character: D.

The command looks like this:

```
<33#23#3B42CA4278E2DDE1#83CBEFA710C6639F#9864AB8659048084B8#
B#306D0D8C8A2E6414#D#>
```

The Network Security Processor returns the following response:

```
<43#F9081E2639080784#Y#99#>
```

Translate PIN – PIN/Pad or Docutel to ANSI (Command 33)

Command 33 – PIN/pad or Docutel to ANSI. This command translates an incoming encrypted PIN block in either a PIN/pad or Docutel PIN block, to outgoing encryption in an ANSI PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if option 46 is enabled.

`Incoming Pad`

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

`PAN`

Field 6, the 12 rightmost digits of the Primary Account Number excluding the check digit. This field contains a 12 byte decimal value.

Table 4-47. Command 33: Translate PIN – PIN/Pad or Docutel to ANSI

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (PIN/pad or Docutel to ANSI)	2	31
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN Block)	16	0 - 9, A - F
5	Incoming pad	1	0 - 9, A - F, X, W
6	PAN	12	0 - 9

*Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response identifier.

E_{KPE_O}`(ANSI PIN Block)`

Field 1, the outgoing encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

`Sanity Check Indicator`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-48. Response 43: Translate PIN – PIN/Pad or Docutel to ANSI

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(ANSI PIN Block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN from PIN/pad to an ANSI PIN block.

- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- Encrypted PIN block: 30DF 0B65 BDFE 91A4.
- Incoming pad character: 7.
- PAN: 1234 5678 9012.

The command looks like this:

```
<33#31#3B42CA4278E2DDE1#83CBEFA710C6639F#30DF0B65BDFE91A4#7#
123456789012#>
```

The Network Security Processor returns the following response:

```
<43#AAC4BCEC8AE1D768#Y#>
```

Translate PIN – PIN/Pad or Docutel to PIN/Pad (Command 33)

Command 33 – PIN/pad or Docutel to PIN/pad. This command translates an incoming encrypted PIN block in either PIN/pad or Docutel PIN block to outgoing encryption in a PIN/pad PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#33#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming Pad#Outgoing Pad#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

33

Field 1, the PIN translation method; in this command, PIN pad character or Docutel to PIN pad.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Incoming Pad

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

Outgoing Pad

Field 6, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal value, X, or W. When this field contains the value W or X the pad character in the incoming PIN block will also be used as the outgoing pad character.

Table 4-49. Command 33: Translate PIN – PIN/Pad or Docutel to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (PIN/Pad to PIN/Pad)	2	33
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming pad	1	0 - 9, A - F, X, W
6	Outgoing pad	1	0 - 9, A - F, X, W

*Can be a volatile table location.

Responding Parameters

43

Field 0, the response identifier.

E_{KPE_O}(PIN Block)

Field 1, the outgoing, encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

Sanity Check Indicator

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.

- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-50. Response 43: Translate PIN – PIN/Pad or Docutel to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in PIN/pad character PIN block to PIN/pad character PIN block.

- Clear-text incoming PIN Encryption Key: 4455445544554455 0000111122223333. The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 72E7AEF691471872 47F102C2D4DE29C4.
- Clear-text outgoing PIN Encryption Key: 2233223322332233 5555666677778888. The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 8C2A7691A708A88D 1DE1CF689E9699D6.
- The clear text PIN block: 987654321F999999.
- The encrypted PIN block: 81A7 8A76 993B E4A7.
- Incoming pad character: 9.
- Outgoing pad character: 9.

The command looks like this:

```
<33#33#72E7AEF69147187247F102C2D4DE29C4#8C2A7691A708A88D1DE1C
F689E9699D6#81A78A76993BE4A7#9#9#>
```

The Network Security Processor returns the following response:

```
<43#C0E6D2796C4B3BFF#Y#>
```

Translate PIN – PIN/Pad or Docutel to IBM 4731 (Command 33)

Command 33 – PIN/Pad or Docutel to IBM 4731. This command translates an incoming encrypted PIN block in either a PIN/Pad or Docutel PIN block, to outgoing encryption in an IBM 4731 PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#39#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming Pad#Outgoing Pad#Outgoing ICV#$E_{MFK.3}$(KC)#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

39

Field 1, the PIN translation method; in this command, PIN/Pad or Docutel to IBM 4731.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Incoming Pad

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

Outgoing Pad

Field 6, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal value, X, or W. When this field contains the value W or X the pad character in the incoming PIN block will also be used as the outgoing pad character.

Outgoing ICV

Field 7, the sequence number for the outgoing PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.3}$(KC)

Field 8, the Communications Key encrypted under variant 3 of the MFK. This key is used in the outer or second encryption of the IBM 4731 PIN block. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 4-51. Command 33: Translate PIN – PIN/Pad or Docutel to IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (ANSI to IBM 4731)	2	39
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming Pad	12	0 - 9, A - F, X, W
6	Outgoing Pad	1	0 - 9, A - F, X, W
7	Outgoing ICV	16	0 - 9, A - F
8	$E_{MFK.3}$(KC)*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response identifier.

E_{KPE_O}`(PIN Block)`

Field 1, the outgoing, encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

`Sanity Check Indicator`

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-52. Response 43: Translate PIN – PIN/Pad or Docutel To IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys, and the Communications Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in PIN/Pad or Docutel PIN block to IBM 4731 PIN block.

- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.

- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- The encrypted PIN block: 86EA C4C4 F7AE 03B8.
- Incoming Pad character: B.
- Outgoing Pad character D.
- Outgoing ICV: 1234 1234 1234 1234.
- Clear-text Communications Key: B302 AD91 F504 EA22.
  The Communications Key encrypted under variant 2 of the MFK: FFFF FFFF FFFF FFFF.

The command looks like this:

```
<33#39#3B42CA4278E2DDE1#83CBEFA710C6639F#86EAC4C4F7AE03B8#B#
D#1234123412341234#FFFFFFFFFFFFFFFF#>
```

The Network Security Processor returns the following response:

```
<43#27682B863CD388E8#Y#>
```

Translate PIN – IBM 4731 to ANSI (Command 33)

Command 33 – IBM 4731 to ANSI translates an incoming encrypted PIN block in the IBM 4731 PIN block to outgoing encryption in an ANSI PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if option 46 is enabled.

Command

<33#91#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming Pad#Incoming ICV#$E_{MFK.3}$(KC)#Outgoing PAN#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

91

Field 1, the PIN translation method; in this command, IBM 4731 to ANSI. When option 46 is enabled, this command will return an error response.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Incoming Pad

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

Incoming ICV

Field 6, the sequence number for the incoming PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.3}$(KC)

Field 7, the Communications Key encrypted under variant 3 of the MFK. This key is used in the outer or second encryption of the IBM 4731 PIN block. This field contains a 16 byte hexadecimal value, or a volatile table location.

Outgoing PAN

Field 8, the Primary Account Number used in the outgoing PIN block. This field contains a 12 byte decimal value.

Table 4-53. Command 33: IBM 4731 to ANSI

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (IBM 4731 to ANSI)	2	91
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming Pad	1	0 - 9, A - F, X, W
6	Incoming ICV	16	0 - 9, A - F
7	$E_{MFK.3}$(KC)*	16	0 - 9, A - F
8	Outgoing PAN	12	0 - 9

*Can be a volatile table location.

Responding Parameters

43

Field 0, the response identifier.

E_{KPE_O}(PIN Block)

Field 1, the outgoing, encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

Sanity Check Indicator

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-54. Response 43: IBM 4731 to ANSI

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an IBM 4731PIN block to an ANSI PIN block.

- Clear-text incoming PIN Encryption Key: 0000 1111 2222 3333.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Clear-text outgoing PIN Encryption Key: 1111 2222 3333 4444.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: D538 A881 DE91 EAF1.
- The encrypted incoming PIN block 3354 3914 C37C FB62.
- Pad character F.
- ICV: 0000 7788 9900 0000.

- Clear-text Communications Key: 4444 5555 6666 7777.
  The Communications Key encrypted under variant 2 of the MFK: E363 8CF7 84F8 4CB0.
- Primary Account Number digits: 7788 9900 0000.

The command looks like this:

```
<33#91#47F102C2D4DE29C4#D538A881DE91EAF1#33543914C37CFB62#F#
0000778899000000#E3638CF784F84CB0#778899000000#>
```

The Network Security Processor returns the following response:

```
<43#CC9FC28E403549DE#Y#>
```

Translate PIN – IBM 4731 to PIN/Pad (Command 33)

Command 33 – IBM 4731 to PIN/Pad translates an incoming encrypted PIN block in the IBM 4731 PIN block to outgoing encryption in a PIN/Pad PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

```
<33#93#E_MFK.1(KPE_I)#E_MFK.1(KPE_O)#E_KPE_I(PIN Block)#
Incoming Pad#Incoming ICV#E_MFK.3(KC)#Outgoing Pad#>
```

Response

```
<43#E_KPE_O(PIN Block)#Sanity Check Indicator#>[CRLF]
```

Calling Parameters

`33`

Field 0, the command identifier.

`93`

Field 1, the PIN translation method; in this command, IBM 4731 to PIN/Pad.

$E_{MFK.1}(KPE_I)$

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}(KPE_O)$

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

`Incoming Pad`

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

`Incoming ICV`

Field 6, the sequence number for the incoming PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.3}(KC)$

Field 7, the Communications Key encrypted under variant 3 of the MFK. This key is used in the outer or second encryption of the IBM 4731 PIN block. This field contains a 16 byte hexadecimal value, or a volatile table location.

`Outgoing Pad`

Field 8, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal character, X, or W. When this field contains the value X or W, the pad character in the incoming PIN block will also be used as the outgoing pad character.

Table 4-55. Command 33: IBM 4731 to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (IBM 4731 to PIN/Pad)	2	93
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming Pad	1	0 - 9, A - F, X, W
6	Incoming ICV	16	0 - 9, A - F
7	$E_{MFK.3}(KC)$*	0, 16	0 - 9, A - F
8	Outgoing Pad	1	0 - 9, A - F, X, W

*Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response identifier.

E_{KPE_O}(PIN Block)

Field 1, the outgoing, encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

Sanity Check Indicator

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-56. Response 43: IBM 4731 to PIN/Pad

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys, and the Communications Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in IBM 4731 PIN block to PIN/Pad PIN block.

- Clear-text incoming PIN Encryption Key: D029 23D9 AD4F E90B.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- Clear-text outgoing PIN Encryption Key: 07CE A74F 4607 5D8F.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- The encrypted PIN block: 2768 2B86 3CD3 88E8.
- Incoming Pad character: D.
- Incoming ICV: 1234 1234 1234 1234.

- Clear-text Communications Key: B302 AD91 F504 EA22.
  The Communications Key encrypted under variant 2 of the MFK: FFFF FFFF FFFF FFFF.
- Outgoing Pad character: B.

The command looks like this:

```
<33#93#83CBEFA710C6639F#3B42CA4278E2DDE1#27682B863CD388E8#D#
1234123412341234#FFFFFFFFFFFFFFFF#B#>
```

The Network Security Processor returns the following response:

```
<43#86EAC4C4F7AE03B8#Y#>
```

Translate PIN – IBM 4731 to IBM 4731 (Command 33)

Command 33 – IBM 4731 to IBM 4731 translates an incoming encrypted PIN block in an IBM 4731 PIN block to outgoing encryption in the IBM 4731 PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<33#99#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPE_I}(PIN Block)#
Incoming Pad#Incoming ICV#$E_{MFK.3}$(Incoming KC)#
Outgoing Pad#Outgoing ICV#$E_{MFK.3}$(Outgoing KC)#>

Response

<43#E_{KPE_O}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

33

Field 0, the command identifier.

99

Field 1, the PIN translation method; in this command, IBM 4731 to IBM 4731.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key (KPE_0) encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Incoming Pad

Field 5, the pad character for the incoming PIN block. The field is one byte, it can contain a hexadecimal value, X, or W. The value X indicates any hexadecimal pad character is allowed. The value W indicates the sanity check will not be performed.

Incoming ICV

Field 6, the sequence number for the incoming PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.3}$(Incoming KC)

Field 7, the incoming Communications Key encrypted under variant 3 of the MFK. This key is used in the outer or second encryption of the IBM 4731PIN block. This field contains a 16 byte hexadecimal value, or a volatile table location.

Outgoing Pad

Field 8, the pad character for the outgoing PIN block. This field is 1 byte, it can contain a hexadecimal value, X, or W. When this field contains the value X or W, the pad character in the incoming PIN block will also be used as the outgoing pad character.

Outgoing ICV

Field 9, the sequence number for the outgoing PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.3}$(Outgoing KC)

Field ten, the outgoing Communications Key encrypted under variant 3 of the MFK. This key is used in the outer or second encryption of the IBM 4731 PIN block. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 4-57. Command 33: IBM 4731 to IBM 4731 (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	33
1	PIN translation method (IBM 4731 to IBM 4731)	2	19
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN block)	16	0 - 9, A - F
5	Incoming Pad	1	0 - 9, A - F, X, W

Table 4-57. Command 33: IBM 4731 to IBM 4731 (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
6	Incoming ICV	16	0 - 9, A - F
7	$E_{MFK.3}$(Incoming KC)*	16	0 - 9, A - F
8	Outgoing Pad	1	0 - 9, A - F, X, W
9	Outgoing ICV	16	0 - 9, A - F
10	$E_{MFK.3}$(Outgoing KC)*	16	0 - 9, A - F

* Can be a volatile table location.

Responding Parameters

`43`

Field 0, the response identifier.

E_{KPE_O}`(PIN Block)`

Field 1, the outgoing, encrypted PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct.

`Sanity Check Indicator`

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-58. Response 43: IBM 4731 to IBM 4731

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	43
1	E_{KPE_O}(PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys, and the Communications Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN formatted in an IBM 4731 PIN block to IBM 4731 PIN block.

- Clear-text incoming PIN Encryption Key: C8B3 047C F7A4 2A70.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 717C 842E 3F0B 8911.
- Clear-text outgoing PIN Encryption Key: 2222 2222 2222 2222.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: C880 88CB 8FE8 46FE.
- The encrypted PIN Block: DE45 A161 F371 9346.
- Incoming Pad character: F.
- Incoming ICV: 0000 1560 0065 0039.
- Clear-text incoming Communications Key: 68D5 9437 1067 794F.
  The Communications Key encrypted under variant 2 of the MFK: D33D 6E7B CC45 E1E6.
- Outgoing Pad Character: D.
- Outgoing ICV: 1234123412341234.
- Clear-text outgoing Communications Key: 0123 4567 89AB CDEF.
  The outgoing Communications Key encrypted under variant 2 of the MFK: 2516 6617 EC74 3AB1.

The command looks like this:

```
<33#99#717C842E3F0B8911#C88088CB8FE846FE#DE45A161F3719346#F#
0000156000650039#D33D6E7BCC45E1E6#D#1234123412341234#
25166617EC743AB1#>
```

The Network Security Processor returns the following response:

```
<43#BA272DB1D8BE0196#Y#>
```

Translate PIN – Double-Encrypted Input or Output (Command 35)

Command 35 – decrypts and re-encrypts an encrypted PIN block, where the input or output is double encrypted. The incoming Communication Key is designated as KC_I, and the outgoing Communications Key is designated as KC_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<35#[$E_{MFK.2}$(KC_I)]#[$E_{MFK.2}$(KC_O)]#PIN Information#>

Response

<45#E_{KPE}(PIN Block)#Sanity Check Indicator#>[CRLF]

Calling Parameters

35

Field 0, the command identifier.

[$E_{MFK.2}$(KC_I)]

Field 1, the incoming Communications Key, used in the outer encryption of the incoming PIN, encrypted under variant 2 of the MFK. This field is either empty, a 16 or 32 byte hexadecimal value or a volatile table location. When this field is empty, the incoming PIN block is single encrypted.

[$E_{MFK.2}$(KC_O)]

Field 2, the outgoing Communications Key, used in the second or outer encryption of the outgoing PIN, encrypted under variant 2 of the MFK. This field is either empty, a 16 or 32 byte hexadecimal value, or a volatile table location. When this field is empty, the outgoing PIN block will be single encrypted.

PIN Information

Field 3, identical to the fields in Commands 31 and 33 starting with Field 1, which specifies the PIN block type for Command 31 or the translation method for

Command 33. The following table identifies the numerical code for each PIN block type.

PIN Block Type	Numerical Code
ANSI	1
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

When option 46 is enabled, this field can only contain the value 1 (ANSI). When option 47 is enabled and option 46 is disabled, the outgoing PIN block type specified in this command must be ANSI.

Table 4-59. Command 35: Translate PIN – Double-Encrypted Input or Output

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	2	35
1	$[E_{MFK.2}(KC_I)]$*	0, 16, 32	0 - 9, A - F
2	$[E_{MFK.2}(KC_O)]$*	0, 16, 32	0 - 9, A - F
3	PIN information		

*Can be a volatile table location.

Responding Parameters

`45`

Field 0, the response identifier.

E_{KPE}`(PIN Block)`

Field 1, the re-encrypted PIN block. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled and the PIN block type (field 3) value is 1, this field will contain 16 zeros.

`Sanity Check Indicator`

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

- Clear-text incoming ANSI PIN block: 0412AC89ABCDEFED.
  The ANSI PIN block encrypted under the incoming PIN Encryption Key: 3ADF0629D11FDAD2.
- The encrypted ANSI PIN block encrypted under the Incoming Communications Key: B37496E8E70673EC
- Twelve rightmost digits of the Primary Account Number: 9876 5432 1012.

The command looks like this:

```
<35#80BCDEAC5703BC84B8880E5C66D21760#C22F5A1F22D1ABF163B2AC82
DBCC9E14#1#47F102C2D4DE29C4D98B3A87979EC8E1#D538A881DE91EAF18
97619CA7FAE7FED#B37496E8E70673EC#987654321012#>
```

The Network Security Processor returns the following response:

```
<45#412FD89E7505CA42#Y#>
```

Verify Double-Encrypted PIN (Command 36)

Command 36 decrypts and verifies an incoming, double-encrypted PIN. The PIN is encrypted with two 1key-3DES (single-length) keys. This command supports only 1key-3DES (single-length) working keys, it does not support 3DES.

This command supports these PIN Verification methods: Identikey, IBM3624, Visa, Atalla DES (Bilevel), Diebold, NCR, Burroughs, and Atalla 2x2.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

```
<36#[E_MFK.2(KC)]#PIN Information#>
```

Response

```
<46#Sanity Check Indicator/Verification Flag#>[CRLF]
```

Calling Parameters

`36`

Field 0, the command identifier.

`[E`$_{MFK.2}$`(KC)]`

Field 1, the Communications Key, used in the second or outer encryption of the incoming PIN, encrypted under variant 2 of the MFK. This field is either empty, a 16 byte hexadecimal value, or a volatile table location. If this field is empty, then the incoming PIN has been single encrypted.

`PIN Information`

Field 2, identical to the fields in Command 32, starting with Field 1, which specifies the PIN verification method. The following table identifies the numerical code for each PIN block type.

PIN Block Type	Numerical Code
ANSI	1
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

Table 4-61. Command 36: Verify Double-Encrypted PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	36
1	$E_{MFK.2}$(KC)*	0, 16	0 - 9, A - F
2	PIN information**		

*Can be a volatile table location.

**Fields from Command 32.

Responding Parameters

`46`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. If the PIN block passes the sanity check the verification check is conducted. This field can contain one of the following values:

- Y – PIN verification was successful.
- N – PIN verification failed.
- S – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-62. Response 46: Verify Double-Encrypted PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	46
1	Sanity check indicator/verification flag	1	Y, N, S, L

Usage Notes

- This command utilizes the logic of command 32, and therefore inherits the same restrictions and requirements.
- Generate the PIN Encryption Keys, Communications Keys and PIN Verification Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a double-encrypted PIN using the Visa PIN verification method.

- Clear-text Communications Key: 3333 3333 3333 3333.
  The Communications Key encrypted under variant 2 of the MFK:
  C22F5A1F22D1ABF1.
- PIN information.
  - Verification method: VISA (3).
  - PIN block type: PIN/pad character (3).
  - Double-encrypted PIN block: 818E39420AA0F83B.
  - Clear-text incoming PIN Encryption Key: 0000 1111 2222 3333.
    The incoming PIN Encryption Key encrypted under variant 1 of the MFK:
    47F102C2D4DE29C4.
  - Clear-text Key Left: 4CA2 1616 37D0 133E.
    The Key Left encrypted under variant 4 of the MFK: 026CA1B523BE5DC4.
  - Clear-text Key Right: 5E15 1AEA 45DA 2A16.
    The Key Right encrypted under variant 4 of the MFK: 96D93C11D37053E2.
  - PIN Verification Value: 3691.
  - PIN Verification Key Indicator: 3.
  - PAN: 12345678901.
  - PIN block data:
    - Pad character: B.
    - Twelve Primary Account Number digits: 123456789019.

The command looks like this:

```
<36#C22F5A1F22D1ABF1#3#3#818E39420AA0F83B#47F102C2D4DE29C4#
026CA1B523BE5DC4#96D93C11D37053E2#3691#3#12345678901#B#
123456789019#>
```

The Network Security Processor returns the following response:

```
<46#Y#>
```

PIN Change – Identikey (Command 37)

Command 37 – Identikey verifies the old PIN using the Atalla Identikey method. If the old PVN verifies, a PVN, based on the new PIN, will be generated. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

This command has the ability to generate a new PVN without verifying the OLD PIN. This functionality has a high security exposure. You must purchase option 66 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<37#1#PIN Block Type#E_{KPE}(Old PIN Block)#$E_{MFK.1}$(KPE)#
Bank ID#PVN#Comparison Indicator#Partial PAN#
E_{KPE}(New PIN Block)#PIN Block Data#>

Response

<47#Sanity Check Indicator#PVN#[IBM 3624 Sequence Number#]>
[CRLF]

Calling Parameters

37

Field 0, the command identifier.

1

Field 1, the PVN verification/generation technique; Identikey.

PIN Block Type

Field 2, specifies the old and new PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2

PIN Block Type	Numerical Code
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

E_{KPE}(Old PIN Block)

Field 3, the old encrypted PIN. When this field is empty and option 66 is enabled, the PIN verification step is not performed before the new PVN is generated. This field is empty, or a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 byte, or 32 byte hexadecimal value, or a volatile table location.

Bank ID

Field 5, the Bank ID; clear-text or encrypted. The clear-text Bank ID is specified by the issuer, it can be a 2, 6, or 8 digit number.

Bank ID	Allowable Size (bytes)
Backward index (algorithm number less than 65)	2
ISO number	6
Route and transfer number	8

The encrypted Bank ID is a 16 hexadecimal character value comprised of the following four data fields ll, bbbbbbbb, p, and cc. It is encrypted under variant 4 of the MFK.

ll - a two-digit number; the length of the Bank ID:

- 02 – The Bank ID in backward index format; the algorithm number must be less than 65.
- 06 – The Bank ID is a six digit ISO number.
- 08 – The Bank ID is an eight digit route-and-transfer number.

bbbbbbbb. The bank ID number (digits 0 - 9); must be the same length as ll.

p. The pad character F, right pads the combined length of the bank ID length (ll) and the bank ID value (bb - bbbbbbbb) resulting in 14 hexadecimal characters. Four pad characters are required when the bank ID is an eight digit value. Six pad characters are required when the bank ID is an six digit value. Ten pad characters are required when the bank ID is a two digit value.

cc. The two hexadecimal character comparison indicator. This field specifies the group (left, middle, or right) of four digits of the six-digit Identikey PIN Verification Number that will be used for the comparison.

- 4C – Compare the leftmost four digits.
- 4D – Compare the middle four digits.
- 52 – Compare the rightmost four digits.

`PVN`

Field 6, the PIN Verification Number. The PVN can be four, six, or eight digits in length, containing the numbers 0 to 7.

`Comparison Indicator`

Field 7, a comparison indicator that specifies which four digits (left, middle, or right) of the six-digit PVN will be compared. This field is 1 byte, and can contain the character L, M, or R. When the PVN is six or eight digits in length or field 5 contains an encrypted bank ID, the value of this field is not evaluated by the Network Security Processor.

`Partial PAN`

Field 8, the portion of the Primary Account Number to be used for verification. This field contains a 4 to 19 byte decimal value.

E_{KPE}`(New PIN Block)`

Field 9, the new encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

`PIN Block Data`

Field 10, PIN block data depends on the PIN block type used, see PIN Block Types on page 4-4.

Table 4-63. Command 37: PIN Change – Identikey (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	37
1	PIN verification method (Identikey)	1	1
2	PIN block type	1	1 - 5
3	E_{KPE}(Encrypted Old PIN Block)	0,16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	Bank ID	2,6,8	0 - 9
6	PIN verification number	4,6,8	0 - 9
7	Comparison indicator	1	L,M,R
8	Partial PAN	4 - 19	0 - 9

Table 4-63. Command 37: PIN Change – Identikey (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
9	E_{KPE}(New PIN Block)	16, 18	0 - 9, A - F
10	PIN block data**	Variable	

* Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`47`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

Value	Description
Y	Old PIN verified successfully.
N	Old PIN failed to verify.
LO	Old PIN length error. See option A1.
NO	No Old PIN. See option 66.
SO	Old PIN sanity error. See PIN Sanity Error.
LN	New PIN length error. See option A1.
SN	New PIN sanity error. See PIN Sanity Error.

`PVN`

Field 2, the PVN associated with the new PIN if the operation completed successfully. This field will be empty if Field 1 is not “Y” or “NO”.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-64. Response 47: PIN Change – Identikey

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	47
1	PIN block OK or Sanity Error	1,2	Y, N, SO, SN, LO, LN, or NO

Table 4-64. Response 47: PIN Change – Identikey

Field #	Contents	Length (bytes)	Legal Characters
2	PVN	0,4,6,8	0 - 9
3	IBM 3624 sequence number*	2	0 - 9, A - F

*Optional field; returned only if the IBM3624 PIN block is used.

Usage Notes

- The new PIN that can be a different length than the old PIN.
- The new and old PIN blocks used in the command must always be the same PIN block type, and encrypted under the same PIN Encryption Key (KPE).

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a PIN and Generating a PVN using the Atalla Identikey method.

- Verification method: Identikey (1).
- PIN block type: ANSI (1).
- Clear-text old ANSI PIN block: 0412 26CB A9ED CBA9.
  The old ANSI PIN block encrypted under the PIN Encryption Key: C84F 6825 74BB AA20.
- Clear-text PIN Encryption Key: 1111 1111 1111 1111.
  The PIN Encryption Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.
- Clear-text new ANSI PIN block: 0443 33CB A9ED CBA9.
  The new ANSI PIN block encrypted under the PIN Encryption Key: 090E 8CA3 CF5D 2AD8.
- The Bank ID is 26.
- The expected PVN is 62732551.
- Since all eight PVN digits are provided, the comparison indicator is not used. The letter “L” is being used strictly as a placeholder.
- The PAN digits used in the algorithm are 1234 5612 3456.

The command looks like this:

```
<37#1#1#C84F682574BBAA20#C6283830AE9E875A#26#62732551#
L#123456123456#090E8CA3CF5D2AD8#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#Y#31724120#>
```

Generating a new PVN without verifying the old PIN.

The command looks like this:

```
<37#1#1##C6283830AE9E875A#26#00000000#
L#123456123456#090E8CA3CF5D2AD8#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#NO#31724120#>
```

PIN Change – IBM 3624 (Command 37)

Command 37 – IBM 3624 verifies the old PIN using the IBM 3624 method of PIN verification. If the old offset is verified, an offset, based on the new PIN, will be generated. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

This command has the ability to generate a new Offset without verifying the old PIN. This functionality has a high security exposure. You must purchase option 66 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<37#2#PIN Block Type#E_{KPE}(Old PIN Block)#$E_{MFK.1}$(KPE)#
Conversion Table#Offset#Validation Data#Pad#Check-Length#
$E_{MFK.4}$(KPV)#E_{KPE}(New PIN Block)#PIN Block Data#>

Response

<47#Sanity Check Indicator#Offset#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

37

Field 0, the command identifier.

2

Field 1, the offset verification/generation technique; IBM 3624.

PIN Block Type

Field 2, specifies the old and new PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2

PIN Block Type	Numerical Code
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

E_{KPE}(Old PIN Block)

Field 3, the old encrypted PIN. When this field is empty and option 66 is enabled, the PIN verification step is not performed before the new offset is generated. This field is empty, or a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key encrypted under variant 1 of the MFK.This field can be either a 16 byte, or 32 byte hexadecimal value, or a volatile table location.

Conversion Table

Field 5, a table that maps hexadecimal digits (0 through 9, A through F) to decimal digits (0 through 9). This field contains a 16 byte decimal value containing the clear-text Conversion Table or a volatile table location. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK.

When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

Offset

Field 6, an offset value applied to the algorithm-generated PIN before comparing it with the customer-entered PIN. This field contains a 4 to 12 byte decimal value.

Validation Data

Field 7, validation data. This is typically the Primary Account Number (PAN). This field contains a 4 to 16 byte hexadecimal value. When the PIN block type is ANSI (field 1 = 1) and option 4C is enabled, the value supplied in this field must be 12 digits in length and equal to the PIN Block Data value supplied in field 12.

Pad

Field 8, the pad character used to right-pad the validation data. This field contains a one byte hexadecimal value. The pad character is only used if the validation data is less than 16 digits.

`Check-Length`

Field 9, the check-length. This value is typically the PIN length and determines the number of PIN digits to be compared. This field contains one hexadecimal character in the range of 4 through C.

$E_{MFK.4}$`(KPV)`

Field 10, the PIN Verification Key (KPV) encrypted under variant 4 of the MFK. This field contains either a 16 or 32 byte hexadecimal value, or a volatile table location.

E_{KPE}`(New PIN Block)`

Field 11, the new encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

`PIN Block Data`

Field 12, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types for information on PIN block data.

Table 4-65. Command 37: PIN Change - IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	37
1	PIN verification method (IBM 3624)	1	2
2	PIN block type	1	1 - 5
3	E_{KPE}(Old PIN Block)	0, 16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16	0 - 9, A - F
5	Conversion table*	16	0 - 9
6	Offset	4 - 12	0 - 9
7	Validation data	4 - 16	0 - 9, A - F
8	Pad	1	0 - 9, A - F
9	Check-Length	1	4 - 9, A - C
10	$E_{MFK.4}$(KPV)*	16	0 - 9, A - F
11	E_{KPE}(New PIN Block)	16, 18	0 - 9, A - F
12	PIN block data**	variable	

* Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`47`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

Value	Description
Y	Old PIN verified successfully.
N	Old PIN failed to verify.
LO	Old PIN length error. See option A1.
NO	No Old PIN. See option 66.
SO	Old PIN sanity error. See PIN Sanity Error.
LN	New PIN length error. See option A1.
SN	New PIN sanity error. See PIN Sanity Error.

`Offset`

Field 2, the Offset associated with the new PIN if the operation completed successfully. This field will be empty if Field 1 is not “Y” or “NO”.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-66. Response 47: PIN Change - IBM 3624

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	47
1	Sanity Check Indicator	1,2	Y, N, SO, SN, LO, LN, or NO
2	Offset	0, 4 - 12	0 - 9
3	IBM 3624 Sequence Number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- The design of Command 37 allows the customer to select a new PIN that can be a different length than their old PIN.

The command looks like this:

```
<37#2#2##C6283830AE9E875A#0123456789012345#0000#123456123456#
F#4#F10C384BC20A721F#978621BD64212AAE92#B#123456123456#
6B5B659A01B7DA63#>
```

The Network Security Processor returns the following response:

```
<47#NO#6140#FF#>
```

PIN Change – VISA (Command 37)

Command 37 – VISA verifies the old PIN using the VISA verification method. If the old PIN Verification Value (PVV) is verified a PVV based on the new PIN will be generated. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

This command has the ability to generate a new PVV without verifying the OLD PIN. This functionality has a high security exposure. You must purchase option 66 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<37#3#PIN Block Type#E_{KPE}(Old PIN Block)#$E_{MFK.1}$(KPE)#
$E_{MFK.4}$(Key Left)#$E_{MFK.4}$(Key Right)#PVV#PVKI#PAN#
E_{KPE}(New PIN Block)#PIN Block Data#>

Response

<47#Sanity Check Indicator#PVV#[IBM 3624 Sequence Number#]>
[CRLF]

Calling Parameters

37

Field 0, the command identifier.

3

Field 1, the PVV verification/generation technique; VISA.

PIN Block Type

Field 2, specifies the old and new PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2

PIN Block Type	Numerical Code
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

E_{KPE}(Old PIN Block)

Field 3, the old encrypted PIN. When this field is empty and option 66 is enabled, the PIN verification step is not performed before the new PVV is generated. This field is empty, or a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.4}$(Key Left)

Field 5, the Key Left encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.4}$(Key Right)

Field 6, the Key Right encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

PVV

Field 7, the PIN Verification Value used to compare to the calculated value. This field contains a 4 byte decimal value. If there is no old PIN to verify, this field should contain four zeros.

PVKI

Field 8, the PIN Verification Key Indicator used to calculate the PIN Verification Value. This field is 1 byte, it can contain the numbers 0 through 9.

PAN

Field 9, the partial Primary Account Number. The VISA algorithm specifies this to be the 11 rightmost PAN digits, excluding the check digit. This field contains a 11 byte decimal value. When the PIN block type is ANSI (field 1 = 1) and option 4C is enabled, the value must be present in the PIN Block Data value supplied in field 11.

E_{KPE}(New PIN Block)

Field 10, the new encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

`PIN Block Data`

Field 11, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4.

Table 4-67. Command 37: PIN Change – VISA

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	37
1	PIN verification method (VISA)	1	3
2	PIN block type	1	1 - 5
3	E_{KPE} (Encrypted Old PIN Block)	0, 16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	$E_{MFK.4}$ (Key Left)*	16	0 - 9, A - F
6	$E_{MFK.4}$ (Key Right)*	16	0 - 9, A - F
7	PVV	4	0 - 9
8	PVKI	1	0 - 9
9	PAN	11	0 - 9
10	E_{KPE} (New PIN Block)	16, 18	0 - 9, A - F
11	PIN block data**	Variable	

* Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`47`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

Value	Description
Y	Old PIN verified successfully.
N	Old PIN failed to verify.
LO	Old PIN length error. See option A1.
NO	No Old PIN. See option 66.
SO	Old PIN sanity error. See PIN Sanity Error.
LN	New PIN length error. See option A1.
SN	New PIN sanity error. See PIN Sanity Error.

`PVV`

Field 2, the PIN Verification Value associated with the new PIN if the operation completed successfully. This field will be empty if Field 1 is not “Y” or “NO”.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-68. Response 47: PIN Change – VISA

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	47
1	Sanity Check Indicator	1,2	Y, N, SO, SN, LO, LN, or NO
2	PVV	0, 4	0 - 9
3	IBM 3624 Sequence Number*	2	0 - 9, A - F

*Optional field; returned only if the IBM 3624 PIN block type is used.

Usage Notes

- The new and old PIN blocks must always be the same PIN block type, and encrypted using the same KPE.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a PIN and Generating a PVV using the Visa method.

- Verification method: VISA (3).
- PIN block type: PIN/pad character (3).
- Clear-text old PIN Pad PIN block: 1234 FFFF FFFF FFFF.
  The old PIN Pad PIN block encrypted under the PIN Encryption Key: EA40 9665 44AB 4654.
- Clear-text PIN Encryption Key: 1111 1111 1111 1111.
  The PIN Encryption Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.
- Clear-text Key Left: 3333 3333 3333 3333.
  The Key Left encrypted under variant 4 of the MFK: F10C 384B C20A 721F.

- Clear-text Key Right: 4444 4444 4444 4444.
  The Key Right encrypted under variant 4 of the MFK: 6F04 64BC 7B03 A41C.
- Clear-text new PIN Pad PIN block: 4321 FFFF FFFF FFFF.
  The new PIN Pad PIN block encrypted under the PIN Encryption Key: B296 DB18 36A3 F011.

The command looks like this:

```
<37#3#3#EA40966544AB4654#C6283830AE9E875A#F10C384BC20A721F#
6F0464BC7B03A41C#9015#1#12345612345#B296DB1836A3F011#F#
123456123456#>
```

The Network Security Processor returns the following response:

```
<47#Y#8449#>
```

Generating a PVV without an old PIN.

The command looks like this:

```
<37#3#3##C6283830AE9E875A#F10C384BC20A721F#6F0464BC7B03A41C#
0000#1#12345612345#B296DB1836A3F011#F#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#NO#8449#>
```

PIN Change – Atalla DES Bilevel (Command 37)

Command 37 – Atalla DES Bilevel verifies the old PIN using the Atalla DES Bilevel method. If the old PVN-2 is verified, a PVN2, based on the new PIN, will be generated. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

This command has the ability to generate a new PVN-2 without verifying the old PIN. This functionality has a high security exposure. You must purchase option 66 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<37#4#PIN Block Type#E_{KPE}(Old PIN Block)#$E_{MFK.1}$(KPE)#
Bank ID#Validation Data#$E_{MFK.4}$(KPV)#PVN-2#PVN-2 Type#
PVN-1 Flag#PVN-2 Start-Compare Flag#E_{KPE}(New PIN Block)#
PIN Block Data#>

Response

<47#Sanity Check Indicator#PVN-2#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

37

Field 0, the command identifier.

4

Field 1, the PVN-2 verification/generation technique; Atalla DES Bilevel.

PIN Block Type

Field 2, specifies the old and new PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2

PIN Block Type	Numerical Code
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

E_{KPE}(Old PIN Block)

Field 3, the old encrypted PIN. When this field is empty and option 66 is enabled, the PIN verification step is not performed before the new PVN-2 is generated. This field is empty, or a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

Bank ID

Field 5, the bank ID field; a 2,6, or 8 byte decimal value.

Data Type	Allowable Size (bytes)
Backward index (algorithm number less than 65)	2
ISO number	6
Route and transfer number	8

Validation Data

Field 6, validation data. The partial Primary Account Number. This field contains a 4 to 19 byte decimal value.

$E_{MFK.4}$(KPV)

Field 7, the PIN Verification Key encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

PVN-2

Field 8, the PIN Verification Number-2. This field contains a 4 to 16 byte hexadecimal value.

If there is no old PIN to verify, this field should contain a PVN-2 of zeros that is equal in length to the desired PVN-2 length for the new PIN. For example, if the PVN-2 length for the new PIN is 6 digits, this field would contain a six zeros.

`PVN-2 Type`

Field 9, the PVN-2 type. This field indicates whether the PVN-2 should be converted to a decimal value. This field is 1 byte, and contains the numbers 0 or 1. The following table identifies the numerical code for each type of PVN-2.

Action	Code
Convert PVN-2 to a decimal value	0
Do not convert PVN-2; leave it as a hexadecimal value	1

`PVN-1 Flag`

Field 10, a flag which indicates that 8 digits of the PVN-1 value are used to compare to PVN-2. This field is 1 byte, and contains the number 8.

`PVN-2 Start-Compare Flag`

Field 11, a PVN-2 Start-Compare Flag that specifies the starting position within the generated PVN-2 for the comparison. This field is 1 byte, and contains the number 1.

E_{KPE}`(New PIN Block)`

Field 12, the new encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

`PIN Block Data`

Field 13, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4.

Table 4-69. Command 37: PIN Change – Atalla DES BiLevel (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	37
1	PIN verification method (Atalla DES BiLevel)	1	4
2	PIN block type	1	1 - 5
3	E_{KPE}(Old PIN Block)	0, 16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	Bank ID	2, 6, 8	0 - 9
6	Validation Data	4 - 19	0 - 9
7	$E_{MFK.4}$(KPV)*	16	0 - 9, A - F
8	PVN-2	4 - 16	0 - 9, A - F
9	PVN-2 Type	1	0, 1
10	PVN-1 Flag	1	8
11	PVN-2 Start-Compare Flag	1	1

Table 4-69. Command 37: PIN Change – Atalla DES BiLevel (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
12	E_{KPE}(New PIN Block)	16, 18	0 - 9, A - F
13	PIN block data**		

* Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`47`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

Value	Description
Y	Old PIN verified successfully.
N	Old PIN failed to verify.
LO	Old PIN length error. See option A1.
NO	No Old PIN. See option 66.
SO	Old PIN sanity error. See PIN Sanity Error.
LN	New PIN length error. See option A1.
SN	New PIN sanity error. See PIN Sanity Error.

`PVN-2`

Field 2, the PVN-2 associated with the new PIN if the operation completed successfully. This field will be empty if Field 1 is not “Y” or “NO”.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-70. Response 47: PIN Change – Atalla DES BiLevel (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	47
1	Sanity Check Indicator	1, 2	Y, N, SO, SN, LO, LN, NO

Table 4-70. Response 47: PIN Change – Atalla DES BiLevel (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
2	PVN-2	0, 4 - 16	0 - 9, A -F
3	IBM 3624 Sequence Number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- The design of Command 37 allows the customer to select a new PIN that can be a different length than their old PIN.
- The new and old PIN blocks used in the command must always be the same PIN block type, and encrypted using the same KPE.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a PIN and Generating a PVN-2 using the Atalla Bilevel method.

- Verification method: Atalla DES BiLevel (4).
- PIN block type: IBM Encrypting PIN Pad (4).
- Clear-text old IBM Encrypting PIN Pad PIN block: 4123 4FFF FFFF FF00.
  The old IBM Encrypting PIN Pad PIN block encrypted under the PIN Encryption Key: 214A 1EFD CFFD 0A1C.
- Clear-text PIN Encryption Key: 1111 1111 111 1111.
  The PIN Encryption Key encrypted under variant 1 of the MFK. C628 3830 AE9E 875A.
- Clear-text PIN Verification Key: 3333 3333 3333 3333.
  The PIN Verification Key encrypted under variant 4 of the MFK: F10C 384B C20A 721F.
- Clear-text new IBM Encrypting PIN Pad PIN block: 4321 FFFF FFFF FF00.
  The new IBM Encrypting PIN Pad PIN block encrypted under the PIN Encryption Key: 0A94 856C 8E80 DF5C.

The command looks like this:

```
<37#4#4#214A1EFDCFFD0A1C#C6283830AE9E875A#26#123456123456#
F10C384BC20A721F#35D96902C6D972C0#1#8#1#0A94856C8E80DF5C#
123456123456#>
```

The Network Security Processor returns the following response:

```
<47#Y#990BE68EF7ECAB92#>
```

Generating a PVN-2 without verifying the old PIN.

The command looks like this:

```
<37#4#4##C6283830AE9E875A#26#123456123456#F10C384BC20A721F#
0000000000000000#1#8#1#0A94856C8E80DF5C#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#NO#990BE68EF7ECAB92#>
```

PIN Change – Diebold (Command 37)

Command 37 – Diebold verifies the old PIN using Diebold method. If the old offset is verified, an offset, based on the new PIN, will be generated. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

This command has the ability to generate a new Offset without verifying the OLD PIN. This functionality has a high security exposure. You must purchase option 66 in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<37#5#PIN Block Type#E_{KPE}(Old PIN Block)#$E_{MFK.1}$(KPE)#
Validation Data#Offset#Algorithm Number#
Diebold Key Table Location#E_{KPE}(New PIN Block)#
PIN Block Data#>

Response

<47#Sanity Check Indicator#Offset#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

37

Field 0, the command identifier.

5

Field 1, the offset verification/generation technique; Diebold.

PIN Block Type

Field 2, specifies the old and new PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2

PIN Block Type	Numerical Code
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

E_{KPE}(Old PIN Block)

Field 3, the old encrypted PIN. When this field is empty and option 66 is enabled, the PIN verification step is not performed before the new offset is generated. This field is empty, or a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 byte, or 32 byte hexadecimal value, or key a volatile table location.

Validation Data

Field 5, validation data. The Primary Account Number (PAN). This field contains a 4 to 19 byte decimal value.

Offset

Field 6, an offset value applied to the algorithm-generated PIN before comparing it with the customer entered PIN. This field contains a 4 byte decimal value. If there is no old PIN to verify, this field should contain an offset of four zeros.

Algorithm Number

Field 7, the Diebold algorithm number. This field contains a 2 byte decimal value.

Diebold Key Table Location

Field 8, the index to the first volatile table location where the Diebold Number Table is stored. This field contains a 1 to 4 byte decimal value.

E_{KPE}[New PIN Block]

Field 9, the new encrypted PIN. This field contains a 16 or 18 byte hexadecimal value.

PIN Block Data

Field 10, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4.

Table 4-71. Command 37: PIN Change – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	37
1	PIN verification method (Diebold)	1	5
2	PIN block type	1	1 - 5
3	E_{KPE}(Encrypted Old PIN Block)	0, 16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	Validation data	4 - 19	0 - 9
6	Offset	4	0 - 9
7	Algorithm Number	2	0 - 9
8	Diebold Key Table Location	1 - 4	0 - 9
9	E_{KPE}(Encrypted New PIN Block)	16, 18	0 - 9, A - F
10	PIN block data**		

* Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`47`

Field 0, the response identifier.

`Sanity Check Indicator/Verification Flag`

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

Value	Description
Y	Old PIN verified successfully.
N	Old PIN failed to verify.
LO	Old PIN length error. See option A1.
NO	No Old PIN. See option 66.
SO	Old PIN sanity error. See PIN Sanity Error.
LN	New PIN length error. See option A1.
SN	New PIN sanity error. See PIN Sanity Error.

`Offset`

Field 2, the offset associated with the new PIN if the operation completed successfully. This field will be empty if Field 1 is not “Y” or “NO”.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-72. Response 47: PIN Change – Diebold

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	47
1	Sanity Check Indicator	1,2	Y, N, SO, SN, LO, LN, NO, INVALID NUMBER TABLE.
2	Offset	0, 4 - 12	0 - 9
3	IBM 3624 Sequence Number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- The design of Command 37 allows the customer to select a new PIN that can be a different length than their old PIN.
- The new and old PIN blocks used in the command must always be the same PIN block type, and encrypted using the same PIN Encryption Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a PIN and Generating a offset using the Diebold method.

The Diebold Number Table must be loaded using command 74 prior to executing this PIN verification command.

- Verification method: Diebold (5).
- PIN block type: Burroughs (5).
- Clear-text old PIN block: 3132 3334 FFFF FFFF.
  The old PIN block encrypted under the PIN Encryption Key: 8814 2C26 5175 6E94.
- Clear-text PIN Encryption Key: 1111 1111 1111 1111.
  The PIN Encryption Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.
- Clear-text new PIN block: 3433 3231 FFFF FFFF.
  The new PIN block encrypted under the PIN Encryption Key: 190A 2878 81D7 1524.

The command looks like this:

```
<37#5#5#88142C2651756E94#C6283830AE9E875A#1234567890#5222#
82#250#190A287881D71524#F#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#Y#2135#>
```

Generating a new offset without verifying the old PIN.

The command looks like this:

```
<37#5#5##C6283830AE9E875A#1234567890#0000#82#250#
190A287881D71524#F#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#NO#2135#>
```

PIN Change – NCR (Command 37)

Command 37 – NCR verifies the old PIN using the NCR method. If the old offset is verified, an offset, based on the new PIN, will be generated.This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

This command has the ability to generate a new offset without verifying the old PIN. This functionality has a high security exposure. You must purchase option 66 in the form of a command 105, and enable it in the Network Security Processor's security policy.

Command

<37#6#PIN Block Type#E_{KPE}(Old PIN Block)#$E_{MFK.1}$(KPE)#
Conversion Table#Offset#Validation Data#Pad#PLEN#$E_{MFK.4}$(KPV)#
Padding Flag#Counting Flag#Start-Count Position#
Select-PLEN Position#E_{KPE}(New PIN Block)#PIN Block Data#>

Response

<47#Sanity Check Indicator#Offset#
[IBM 3624 Sequence Number#]>[CRLF]

Calling Parameters

37

Field 0, the command identifier.

6

Field 1, the offset verification/generation technique NCR.

PIN Block Type

Field 2, specifies the old and new PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

E_{KPE}(Old PIN Block)

Field 3, the old encrypted PIN. When this field is empty and option 66 is enabled, the PIN verification step is not performed before the new offset is generated. This field is empty, or a 16 or 18 byte hexadecimal value.

$E_{MFK.1}$(KPE)

Field 4, the PIN Encryption Key encrypted under variant 1 of the MFK. This field can be either a 16 or 32 byte hexadecimal value, or a volatile table location.

Conversion Table

Field 5, a table that maps hexadecimal digits (0 through 9, A through F) to decimal digits (0 through 9). This field contains a 16 byte decimal value containing the clear-text Conversion Table or a volatile table location. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK.

When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

Offset

Field 6, an offset value applied to the algorithm-generated PIN before comparing it with the customer-entered PIN. This field contains a 4 to 16 byte decimal value.

Validation Data

Field 7, validation data. This value is unique for each card holder, and in the case of this command, is the partial Primary Account Number (PAN). This field contains a 4 to 16 byte hexadecimal value. When the PIN block type is ANSI (field 1 = 1) and option 4C is enabled, the value supplied in this field must be 12 digits in length and equal to the PIN Block Data value supplied in field 16.

`Pad`

Field 8, a pad character that right-pads the validation data. This field contains a one byte hexadecimal value.

`PLEN`

Field 9, the number of contiguous PIN digits selected for verification; the PIN length, or PLEN. This field is one byte; it can contain the numbers 4 through 9 and the characters A, B, and C.

$E_{MFK.4}$(KPV)

Field 10, the PIN Verification Key encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

`Padding Flag`

Field 11, a flag that indicates whether the validation data (Field 7) is to be padded on the left or right. This field is 1 byte, and contains the character L or R.

`Counting Flag`

Field 12, a flag that indicates whether the counting scheme for selecting the PIN digit for verification is left or right. This field is 1 byte, and contains the character L or R.

`Start-Count Position`

Field 13, the field that indicates the starting position for the counting scheme measured from either the left or right of the entered PIN depending on field 12. This field is one byte, it can contain a number in the range of 1 through 9.

`Select-PLEN Position`

Field 14, the field that indicates the beginning position (from the left or right, depending upon the counting flag, starting with 0) for selecting PLEN characters from the output of the DES encryption step. This field is one byte, it can contain a character in the range of 0 through 9, A through C.

E_{KPE}(New PIN Block)

Field 15, the encrypted new PIN block. This field contains a 16 or 18 byte hexadecimal value.

`PIN Block Data`

Field 16, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4.

Table 4-73. Command 37: PIN Change – NCR

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	37
1	PIN verification method (NCR)	1	6
2	PIN block type	1	1 - 5
3	E_{KPE}(Old PIN Block)	16, 18	0 - 9, A - F
4	$E_{MFK.1}$(KPE)*	16, 32	0 - 9, A - F
5	Conversion table*	16	0 - 9
6	Offset	4 - 16	0 - 9
7	Validation data	4 - 16	0 - 9, A - F
8	Pad	1	0 - 9, A - F
9	PLEN	1	4 - 9, A - C
10	$E_{MFK.4}$(KPV)*	16	0 - 9, A - F
11	Padding Flag	1	L, R
12	Counting Flag	1	L, R
13	Start-Count Position	1	1 - 9
14	Select-PLEN Position	1	0 - 9, A - C
15	E_{KPE}(New PIN Block)	16, 18	0 - 9, A - F
16	PIN block data**		

* Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

47

Field 0, the response identifier.

Sanity Check Indicator/Verification Flag

Field 1, the sanity check indicator and verification flag. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

Value	Description (page 1 of 2)
Y	Old PIN verified successfully.
N	Old PIN failed to verify.
LO	Old PIN length error. See option A1.
NO	No Old PIN. See option 66.

Value	Description (page 2 of 2)
SO	Old PIN sanity error. See PIN Sanity Error.
LN	New PIN length error. See option A1.
SN	New PIN sanity error. See PIN Sanity Error.

`Offset`

Field 2, the offset associated with the new PIN if the operation completed successfully. This field will be empty if Field 1 is not “Y” or “NO”.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

Table 4-74. Response 47: PIN Change – NCR

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	47
1	Sanity Check Indicator	1,2	Y, N, SO, SN, LO, LN, NV
2	Offset	4 - 12	0 - 9
3	IBM 3624 Sequence Number*	2	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- The new and old PINs must be the same length.
- The new and old PIN blocks used in the command must always be the same PIN block type, and encrypted using the same PIN Encryption Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a PIN and Generating a offset using the NCR method.

- Verification method: NCR (6).
- PIN block type: ANSI (1).
- Clear-text old PIN block: 0412 26CB A9ED CBA9.
  The old PIN, ANSI PIN block encrypted under the PIN Encryption Key: C84F 6825 74BB AA20.

- Clear-text PIN Encryption Key: 1111 1111 1111 1111.
  The PIN Encryption Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.
- Clear-text PIN Verification Key: 68BA 0794 F140 641C.
  The PIN Verification Key encrypted under variant 4 of the MFK: FE87 4532 1894 0916.
- Clear-text new PIN block: 0443 33CB A9ED CBA9.
  The new PIN block encrypted under the PIN Encryption Key: 090E 8CA3 CF5D 2AD8.

The command looks like this:

```
<37#6#1#C84F682574BBAA20#C6283830AE9E875A#0123456789012345#
0919#2700455240000121#F#4#FE87453218940916#R#L#1#6#
090E8CA3CF5D2AD8#123456123456#>
```

The Network Security Processor returns the following response:

```
<47#Y#3006#>
```

Translate PIN And Generate MAC (Command 39)

Command 39 – translates an encrypted PIN from encryption under one key to encryption under another and generates a Message Authentication Code (MAC) from data contained in the command. The outgoing PIN block type is ANSI. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_O. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<39#PIN Block Type#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#$E_{KPE.I}$(PIN Block)#
PIN Block Data#$E_{MFK.3}$(KMAC)#Flag#Data#>

Response

<49#E_{KPEO}(ANSI PIN block)#Sanity Check Indicator#
[IBM 3624 Sequence Number#]MAC#KMAC Check Digits#>[CRLF]

Calling Parameters

39

Field 0, the command identifier.

PIN Block Type

Field 1, incoming PIN block type. This field is 1 byte, it can contain the numbers 1 to 5.When option 46 is enabled, this field can only contain the value 1 (ANSI).

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}(KPE_O)$

Field 3, the outgoing PIN Encryption Key encrypted under variant 1 of the MFK. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2). This field contains a 16 or 32 byte hexadecimal value or a volatile table location.

$E_{KPE.I}$(PIN Block)

Field 4, the incoming encrypted PIN encrypted under the PIN Encryption Key. This field contains a 16 or 18 byte hexadecimal value.

PIN Block Data

Field 5, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

$E_{MFK.3}$(KMAC)

Field 6, the Message Authentication Code key encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Flag

Field 7, a flag. If you will be including the translated PIN block in the MAC generation, set this field to 1; otherwise, set this field to 0. If the flag is set to 1, the translated PIN block will precede the data to be MACed.

Data

Field 8, the data to be authenticated. This field can be up to 239 bytes, it can contain the numbers 0 through 9 and the characters A to Z, as well as commas, periods, and blanks.

Table 4-75. Command 39: Translate PIN and Generate MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	39
1	PIN block type	1	1 - 5
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	$E_{KPE.I}$(Encrypted PIN block)	16, 18	0 - 9, A - F
5	PIN block data**		
6	$E_{MFK.3}$(KMAC)*	16	0 - 9, A - F
7	Flag	1	0, 1
8	Data	1 - 239	0 - 9, A - Z , . “ ”

*Can be a volatile table location.

**See PIN Block Types on page 4-4 for information on PIN block data.

Responding Parameters

`49`

Field 0, the response identifier.

$E_{KPE.O}$(ANSI PIN Block)

Field 1, the encrypted outgoing PIN. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check Indicator`

Field 2, the sanity check indicator. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624 otherwise, this field is not used. This field contains a 2 byte hexadecimal value.

`MAC`

Field 4, the Message Authentication Code. This field contains an 8 byte hexadecimal value. This field is empty when the PIN block fails the sanity check.

`KMAC Check Digits`

Field 5, check digits; the first four digits that result of encrypting zeros using the Message Authentication Code key. If option 88 is enabled, this field will contain the first six digits of the result. This field is empty when the PIN block fails the sanity check.

Table 4-76. Response 49: Translate PIN and Generate MAC (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	2	49
1	$E_{KPE.O}$(ANSI PIN block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L
3*	IBM 3624 Sequence Number	2	0 - 9, A - F

Table 4-76. Response 49: Translate PIN and Generate MAC (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	MAC	0, 8	0 - 9, A - F
5	KMAC Check Digits	0, 4 or 6	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.
- Generate the Message Authentication Code key.
- Generate the ATM Communications Key if the incoming PIN block is IBM 3624.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating an ANSI formatted PIN and generating Message Authentication Code.

- PIN block type: ANSI (1).
- Clear-text incoming PIN Encryption Key: 0000 1111 2222 3333.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 47F1 02C2 D4DE 29C4.
- Clear-text outgoing PIN Encryption Key: 1111 2222 3333 4444.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: D538 A881 DE91 EAF1.
- Clear-text incoming PIN block: 0C12 3456 7890 12FF.
  The incoming PIN block encrypted under the PIN Encryption Key: 4476 A5ED F270 3FF8.
- PIN block data; in this case, the 12 digits of the Primary Account Number: 7788 9900 0000.
- Clear-text Message Authentication Code key: FEDC BA98 7654 3210.
  The Message Authentication Code key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- Flag: 0.
- Data to be authenticated: ABCD 1234 ABCD 1234.

The command looks like this.

```
<39#1#47F102C2D4DE29C4#D538A881DE91EAF1#4476A5EDF2703FF8#
778899000000#1B866280C012DD33#0#ABCD1234ABCD1234#>
```

The Network Security Processor returns the following response.

```
<49#1371A72D914FDE41#Y#68AE2DD2#A68C#>
```

Generate PVN and IBM Offset (Command 3D)

Command 3D generates both an Identikey PVN and an IBM3624 offset from the account number and encrypted PIN. This command supports only 1key-3DES (single-length) working keys.

This command is not enabled in the Network Security Processor's default factory security policy. You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

```
<3D#PIN Block Type#E_KPE(PIN Block)#E_MFK.1(KPE)#Bank ID#
Partial PAN#Conversion Table#Validation Data#Pad#
E_MFK.4(KPV)#PIN Block Data#>
```

Response

```
<4D#PVN/Sanity Check Indicator#[IBM 3624 Offset#]
[IBM 3624 Sequence Number#]>
```

Calling Parameters

`3D`

Field 0, the command identifier.

`PIN Block Type`

Field 1, the incoming PIN block type.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3

E_{KPE}(PIN Block)

Field 2, the encrypted PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.1}$(KPE)

Field 3, the PIN Encryption Key encrypted under variant 1 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

`Bank ID`

Field 4, the Bank ID. This field can be a 2, 6, or 8 byte decimal value.

`Partial PAN`

Field 5, the portion of the Primary Account Number to be used in the Identikey PVN generation process. This field contains a 4 to 19 byte decimal value.

`Conversion Table`

Field 6, a table that maps hexadecimal digits (0 through 9, A through F) to decimal digits (0 through 9). This field contains a 16 byte decimal value containing the clear-text Conversion Table or a volatile table location. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK.

When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

`Validation Data`

Field 7, validation data. This value is unique for each card holder and is typically the account number. This field contains a 4 to 16 byte hexadecimal value. When the PIN block type is ANSI (field 1 = 1) and option 4C is enabled, the value supplied in this field must be 12 digits in length and equal to the PIN Block Data value supplied in field 10.

`Pad`

Field 8, the pad character to be used to form the validation data. This field contains a 1 byte hexadecimal value.

$E_{MFK.4}$(KPV)

Field 9, the PIN Verification Key encrypted under variant 4 of the MFK. This key is used in the IBM3624 offset generation process. This field contains 16 hexadecimal characters.

`PIN Block Data`

Field 10, PIN block data. The content and number of fields depend on the PIN block type. See PIN Block Types on page 4-4 for information on PIN block data.

Table 4-77. Command 3D: Generate PVN and IBM Offset

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	3D
1	PIN block type	1	1 - 3
2	E_{KPE}(PIN Block)	16	0 - 9, A - F
3	$E_{MFK.1}$(KPE)*	16	0 - 9, A - F
4	Bank ID	2, 6, 8	0 - 9
5	Partial PAN	4 - 16	0 - 9
6	Conversion Table*	16	0 - 9
7	Validation data	4 - 16	0 - 9, A - F
8	Pad	1	0 - 9, A - F
9	$E_{MFK.4}$(KPV)*	16	0 - 9, A - F
10	PIN block data**		

*Can be a volatile table location.

**See PIN Block Types on page 4-4 earlier in this section for information on PIN block data.

Responding Parameters

`4D`

Field 0, the response identifier.

`PVN or a Sanity Check Indicator`

Field 1, the PVN associated with the PIN is returned if the command executed successfully. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. When the sanity test fails this field will contain one of the following values:

- S – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Offset#]`

Field 2, the offset associated with the PIN. The offset length is the same as the PIN length. This field will not be present if the PIN fails the sanity test.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 PIN block sequence number. This field is returned only if the PIN Block type is IBM 3624.This field contains a 2 byte hexadecimal value.

Table 4-78. Response 4D: Generate PVN and IBM Offset

Usage Notes

- Generate the PIN Encryption Key.

Example

Generating a PVN and IBM offset from an ANSI PIN block.

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- PIN block type: ANSI (1).
- Encrypted PIN block: 090E8CA3CF5D2AD8.
- Clear-text PIN Encryption Key: 1111 1111 1111 1111.
  The PIN Encryption Key encrypted under variant 1 of the MFK: C628 3830 AE9E 875A.
- Bank ID: 26.
- The partial PAN used for Identikey: 123456123456.
- Conversion Table: 0123456789012345.
- Validation data: 123456123456.
- Pad: F
- Clear-text PIN Verification Key: 3333 3333 3333 3333.
  The PIN Verification Key encrypted under variant 4 of the MFK: F10C 384B C20A 721F.
- The PIN block data: 123456123456.

The command looks like this.

```
<3D#1#090E8CA3CF5D2AD8#C6283830AE9E875A#26#123456123456#
0123456789012345#123456123456#F#F10C384BC20A721F#
123456123456#>
```

The Network Security Processor returns the following response.

```
<4D#31724120#6140#>
```

Decrypt PIN (Command 90)

Command 90 decrypts an incoming PIN block and returns the clear-text PIN. This command supports both 1key-3DES (single-length) and 2key-3DES (double-length) working keys.

You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

Command

```
<90#PIN Block Type#E_KPE(PIN Block)#E_MFK.1(KPE)#
PIN Block Data#>
```

Response

```
<A0#Clear-Text PIN or Sanity Check Indicator#>[CRLF]
```

Calling Parameters

`90`

Field 0, the command identifier.

`PIN Block Type`

Field 1, the incoming PIN block type.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3

E_{KPE}(PIN Block)

Field 2, the encrypted PIN block. This field contains 16 hexadecimal characters.

$E_{MFK.1}$(KPE)

Field 3, the PIN Encryption Key encrypted under variant 1 of the MFK. This field contains either a 16 or 32 byte hexadecimal value, or a volatile table location.

`PIN Block Data`

Field 4, PIN block data. Its contents depend on the PIN block type. See PIN Block Types on page 4-4.

Table 4-79. Command 90: Decrypt PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	90
1	PIN block type	1	1 - 3
2	E_{KPE}(PIN Block)	16	0 - 9, A - F
3	$E_{MFK.1}$(KPE)*	16 or 32	0 - 9, A - F
4	PIN block data**		

*Can be a volatile table location.

*See PIN Block Types on page 4-4 information on PIN block data.

Responding Parameters

`A0`

Field 0, the response identifier.

`Clear-Text PIN or Sanity Check Indicator`

Field 1, the clear-text PIN is present if the PIN block passed the sanity test. When the sanity test fails, this field will contain one of the following values:

- S – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

Table 4-80. Response A0: Decrypt PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A0
1	Clear-text PIN or sanity check indicator	1 - 12 if clear-text PIN is returned; otherwise, 1	0 - 9, S, L

Usage Notes

- Generate the PIN Encryption Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Decrypting an encrypted PIN block.

- Clear-text PIN block: 1234 5FFF FFFF FFFF.
  The PIN block encrypted under the PIN Encryption Key: 7B58 719B 354B 147A.
- Clear-text PIN Encryption Key: 2233 2233 2233 2233.
  The PIN Encryption Key encrypted under variant 1 of the MFK: 8C2A 7691 A708 A88D.
- PIN block data; 12 digits of the Primary Account Number: 9876 5432 1012.

The command looks like this:

```
<90#1#7B58719B354B147A#8C2A7691A708A88D#987654321012#>
```

The Network Security Processor returns the following response:

```
<A0#12345#>
```

PIN Translate (ANSI to PIN/Pad) and MAC Verification (Command BA)

Command BA performs two functions in a single command. It translates PINs from encryption under one key to encryption under another, and it verifies a Message Authentication Code (MAC). The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. This command will return an error if either option 46 or 47 is enabled.

Command

<BA#13#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPEI}(PIN Block)#Pad#
ANSI PAN Digits#$E_{MFK.3}$(KMAC)#Data#MAC#>

Response

<CA#E_{KPEO}(PIN/Pad PIN block)#Sanity Check#KMAC Check Digits#
Verification Flag#>

Calling Parameters

BA

Field 0, the command identifier.

13

Field 1, ANSI PIN block to PIN/pad block.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key encrypted under variant 1 of the MFK. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2). This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

E_{KPEI}(PIN Block)

Field 4, the incoming PIN Block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

Pad

Field 5, the pad character in the PIN pad block. This field is 1 byte, it can contain a hexadecimal value, X or W. When this field contains the value X or W, the pad character used in the incoming PIN block will also be used as the outgoing pad character.

ANSI PAN Digits

Field 6, the Primary Account Number digits used in the incoming ANSI PIN block. This field contains a 12 byte decimal value.

$E_{MFK.3}$(KMAC)

Field 7, the MAC Key encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Data

Field 8, this data will be authenticated according to ANSI specification X9.9. This field can be from one to 240 bytes, it can contain the characters A to Z, the numbers 0 through 9, and “,”, “.”, and “ ”.

MAC

Field 9, the 8-bit MAC to be verified. This is an 8 byte hexadecimal value.

Table 4-81. Command BA: PIN Translate (ANSI to PIN/Pad) and MAC Verification

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	BA
1	ANSI PIN block	2	13
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPEI}(Encrypted PIN Block)	16	0 - 9, A - F
5	PIN/Pad Character	1	0 - 9, A - F, W, X
6	ANSI PAN Data	12	0 - 9
7	$E_{MFK.3}$ (KMAC)*	16	0 - 9, A - F
8	Data per ANSI X9.9 only	1-240	0 - 9, A - Z, , . “ ”
9	MAC	8	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`CA`

Field 0, the response identifier.

E_{KPE_O}`(PIN/Pad PIN block)`

Field 1, the encrypted PIN in PIN pad format. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`KMAC Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the message authentication key. If option 88 is enabled, this field will contain the first six digits of the result.

`Verification Flag`

Field 4, the MAC verification flag.

Y – the MAC verified

N – the MAC did not verify

Table 4-82. Response CA: PIN Translate (ANSI to PIN/Pad) and MAC Verification

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	CA
1	E_{KPE_O}(PIN/Pad PIN block)	16	0 - 9, A - F
2	Sanity Check	1	Y, N, L
3	KMAC Check Digits	4 or 6	0 - 9, A - F
4	Verification Flag	1	Y, N

Usage Notes

- Generate the Message Authentication Code Key and the incoming and outgoing PIN Encryption Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN from ANSI to PIN/Pad PIN block and verifying a MAC.

- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- The PIN block encrypted under the PIN Encryption Key: 5196 681F 910C 408C.
- ANSI Primary Account Number digits: 1207 4108 1445.
- Pad character: F.
- Clear-text Message Authentication Code Key: D377 30CD D619 FE8A.
  The Message Authentication Code Key encrypted under variant 3 of the MFK: 8FF4 98F1 B661 5151.
- Data to be authenticated:
  A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X.
- The MAC to be verified: 4316C2C1.

The command looks like this.

```
<BA#13#3B42CA4278E2DDE1#83CBEFA710C6639F#5196681F910C408C#F#
120741081445#D37730CDD619FE8A#A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5
P6Q7R8S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9
T0U1V2W3X#4316C2C1#>
```

The Network Security Processor returns the following response.

```
<CA#7DBE8020E51B8C36#Y#1DE3#Y#>
```

Translate PIN (ANSI to PLUS) and Verify MAC (Command BB)

Command BB translates PINs from encryption under one key to another and verifies a MAC. The Network Security Processor decrypts the incoming ANSI PIN block, verifies the MAC, and encrypts the outgoing PIN block. The incoming PIN Encryption key is designated as KPE_I, and the outgoing PIN Encryption Key is designated as KPE_0. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<BB#11#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#E_{KPEI}(ANSI PIN Block)#
ANSI PAN Digits#PLUS PAN Digits#$E_{MFK.3}$(KMAC)#Data#MAC#>

Response

<CB#E_{KPEO}(PLUS PIN Block)#Sanity Check Indicator#
KMAC Check Digits#Verification Flag#>[CRLF]

Calling Parameters

BB

Field 0, the command identifier.

11

Field 1, the PIN block type; in this command, ANSI.

$E_{MFK.1}$(KPE$_I$)

Field 2, the incoming PIN Encryption Key encrypted under variant 1 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

$E_{MFK.1}$(KPE$_O$)

Field 3, the outgoing PIN Encryption Key encrypted under variant 1 of the MFK. When option 49 is enabled, the length of the KPEo must be equal to or greater than the length of the KPEi (field 2). This field contains a 16 or 32 byte hexadecimal value, or a volatile table location.

E_{KPEI}(ANSI PIN Block)

Field 4, the incoming ANSI PIN Block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

`ANSI PAN Digits`

Field 5, the ANSI Primary Account Number; the 12 rightmost digits of the Primary Account Number excluding the check digit. This field contains a 12 byte decimal value. When either option 46 or 47 is enabled, the value of this field and field 6 must be identical.

`PLUS PAN Digits`

Field 6, the PLUS Primary Account Number; the 12 leftmost digits of the Primary Account Number. This field contains a 12 byte decimal value.

$E_{MFK.3}$(KMAC)

Field 7, the Message Authentication Code Key encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

`Data`

Field 8, the data to be authenticated. This field can be up to 240 bytes, it can contain the numbers 0 through 9, the characters A to Z, “,”, “.”, and “ ”.

`MAC`

Field 9, the MAC to be verified. This field contains an 8 byte hexadecimal value.

Table 4-83. Command BB: Translate PIN (ANSI to PLUS) and Verify MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	BB
1	ANSI PIN block	2	11
2	$E_{MFK.1}(KPE_I)$*	16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	16, 32	0 - 9, A - F
4	E_{KPEI}(ANSI PIN Block)	16	0 - 9, A - F
5	ANSI PAN Digits	12	0 - 9
6	PLUS PAN Digits	12	0 - 9
7	$E_{MFK.3}$(KMAC)*	16	0 - 9, A - F
8	Data	1 - 240	0 - 9, A - Z, , . “ ”
9	MAC	8	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`CB`

Field 0, the response identifier.

E_{KPEO}`(PLUS PIN Block)`

Field 1, the PIN in Plus format encrypted under the outgoing PIN Encryption Key. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field can contain one of the following values:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`KMAC Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the Message Authentication Code Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Verification Flag`

Field 4, the MAC verification flag. This field returns Y if the MAC is verified; otherwise, it returns N.

Table 4-84. Response CB: Translate PIN (ANSI to PLUS) and Verify MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	CB
1	E_{KPEI}(PLUS PIN Block)	16	0 - 9, A - F
2	Sanity check indicator	1	Y, N, L
3	KMAC Check Digits	4 or 6	0 - 9, A - F
4	Verification flag	1	Y, N

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.
- Generate the Message Authentication Code key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating a PIN from ANSI to PLUS PIN block and verifying a MAC.

- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 3B42 CA42 78E2 DDE1.
- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CB EFA7 10C6 639F.
- The PIN block encrypted under the incoming PIN Encryption Key: 5196 681F 910C 408C.
- ANSI Primary Account Number digits: 1207 4108 1445.
- PLUS Primary Account Number digits: 2074 1081 4457.
- Clear-text Message Authentication Code Key: 8FF4 98F1 B661 5151.
  The Message Authentication Code Key encrypted under variant 3 of the MFK: D377 30CD D619 FE8A.
- Data to be authenticated:
  A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X.
- The MAC to be verified: 4316C2C1.

The command looks like this.

```
<BB#11#3B42CA4278E2DDE1#83CBEFA710C6639F#5196681F910C408C#
120741081445#207410814457#D37730CDD619FE8A#A1B2C3D4E5F6G7H8I
9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M
3N4O5P6Q7R8S9T0U1V2W3X#4316C2C1#>
```

The Network Security Processor returns the following response.

```
<CB#7BB41A6FAA3BF848#Y#1DE3#Y#>
```

Translate PIN and Generate MAC (Command BD)

Command BD translates an encrypted PIN from encryption under one key to encryption under another and generates a Message Authentication Code (MAC) from data contained in the message. The translated PIN cryptogram can be included in the data for MAC generation. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) PIN Encryption Keys (KPE)s and Message Authentication Keys (KMAC)s.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<BD#PIN Block Type#$E_{MFK.1}$(KPE$_I$)#$E_{MFK.1}$(KPE$_O$)#
E_{KPE_I}(PIN Block)#$E_{MFK.V}$(KMAC)#MAC Length#
[$E_{MFK.6}$(IV)]#Insertion Position#Insertion Type#Data Type#
Data Length#Data#[Variant(V)]#[PIN Block Data]#>

Response

<CD#E_{KPE_O}(ANSI PIN Block)#Sanity Check Indicator#
[IBM 3624 Sequence Number#]MAC Length#
MAC or $E_{MFK.6}$(Ending IV)#KMAC Check Digits#KPE$_I$ Check Digits#
KPE$_O$ Check Digits#>

Calling Parameters

BD

Field 0, the command identifier.

PIN Block Type

Field 1, the incoming PIN block type. If this field is empty, only the MAC generation operation will be performed and fields 2, 3, 4, 8, 9, and 14 must also be empty. This field contains a 1 byte decimal value which can be 1 through 5 or 9, or is empty. When option 46 is enabled, this field can contain the value 1 (ANSI) or be empty.

PIN Block Type	Numerical Code
ANSI	1
IBM 3624	2
PIN/pad character / Docutel	3
IBM encrypting PIN pad	4
Burroughs	5
IBM 4731	9

$E_{MFK.1}(KPE_I)$

Field 2, the incoming PIN Encryption Key encrypted under variant 1 of the MFK. If this field is empty, it indicates that only MAC generation operation will be performed and fields 1 through 4 and field 14 must be empty. This field contains a 16 or 32 byte hexadecimal value, or volatile table location, or is empty.

$E_{MFK.1}(KPE_O)$

Field 3, the outgoing PIN Encryption Key encrypted under variant 1 of the MFK. If this field is empty, it indicates that only MAC generation operation will be performed and fields 1 through 4 and field 14 must be empty. This field contains a 16 or 32 byte hexadecimal value, a volatile table location, or is empty. When option 49 is enabled, an error response is returned if the length of the (KPEo) is not equal to or greater than the length of the (KPEi).

E_{KPE_I}(PIN Block)

Field 4, the incoming PIN block encrypted under the incoming PIN Encryption Key. If this field is empty, it indicates that only MAC generation operation will be performed and fields 1 through 4 and field 14 must be empty. This field contains a 16 or 18 byte hexadecimal value, or is empty.

$E_{MFK.V}$(KMAC)

Field 5, the Message Authentication Code Key encrypted under the variant, specified in field 13, of the MFK. This field contains a 16 or 32 byte hexadecimal value, or volatile table location.

MAC Length

Field 6, the size of the Message Authentication Code to be generated. The following table indicates the possible MAC sizes and the code to enter in this field

for each one. If this field is set to zero, then the fields 4 and 5 should be empty; otherwise, an error is returned.

Returned-MAC Size	Numerical Code
More data expected; no MAC verified	0
32 bits	1
48 bits	2
64 bits	3

A 32-bit Message Authentication Code is expressed as eight hexadecimal digits (0-9, A-F) and written as two groups of four digits, separated by a space. A 48- or 64-bit Message Authentication Code is expressed as three or four groups of four hexadecimal digits, separated by a space.

$[E_{MFK.6}(IV)]$

Field 7, the Initialization Vector encrypted under variant 6 of the MFK. If this command contains the first block of multiple blocks of data, or if only one block of data will be authenticated this field must be empty. If this command contains data subsequent to the first block in a multi-block series, this field should contain the ending Initialization Vector from the previously sent data block. This field contains a 16 byte hexadecimal value, a volatile table location, or is empty.

Insertion Position

Field 8, the number indicates where the translated PIN block is inserted into the data **in this command** for MAC generation. If you will not be including the translated PIN block in the MAC generation, set this field to 0. The number means inserting the PIN block in between binary data position -1 and For unpacked data, the same rule applies, but two unpacked characters are considered one binary data. This field can also contain character 'F' which indicates the first location in the data and 'L' indicates the last location in the data. If the data type is Unpacked ('U'), then this field must contain an even number. This field should be empty if any of the fields 1 through 4, or 14 are empty.

Example:

31 32 33 34 35 36	input data in unpacked
12 34 56	input data in binary
31 32^33 34 35 36	position 2 in unpacked
12^34 56	position 2 in binary
^31 32 33 34 35 36	position F
12 34 56 12 34 56^	position L

^ denotes where the encrypted PIN block goes. The PIN block is converted to the type specified in Field 9 before including into the data to be MACed.

Insertion Type

Field 9, the PIN Block insertion type (A, B, E).

- A – PIN block will be converted to ASCII (unpacked) hex before including in the MACed data at the position indicated in Field 8.
- B – PIN block will be converted to binary before including in the MACed data at the position indicated in Field 8.
- E – PIN block will be converted to EBCDIC before including in the MACed data at the position indicated in Field 8.

These conversions will take place regardless of the value indicated in Field 10. This field should be empty if any of the fields 1 through 4 or 14 is empty, or if field 8 contains 0.

`Data Type`

Field 10, the data type. The data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

`Data Length`

Field 11, the length of data. This command will authenticate up to 4096 bytes of data. If more data is being sent in the next command – indicated by Field 6 being set to zero – then the data length must be a multiple of eight. If no more data is being sent, the Network Security Processor will right-pad the data field with binary zeros (nulls, 0x00) such that the resulting data length will be a multiple of eight. This field contains a 1 to 4 byte decimal value.

`Data`

Field 12, the input data. This field can be from one to 4096 bytes long and in binary or unpacked ASCII hexadecimal format. If the data is in unpacked ASCII hexadecimal format, then this field can contain the numbers 0 through 9 and the characters A through F.

`[Variant(V)]`

Field 13, the variant of the MFK used to encrypt the Message Authentication Code key (KMAC). This field is optional; if present, it can be one or two bytes long, and may contain the numbers 3 or 18. If this field is empty the default variant, 3, is used.

`[PIN Block Data]`

Field 14, the PIN block data. Its contents depend on the PIN block type used. See PIN Block Types on page 4-4. If this field is empty, it indicates that only MAC generation will be performed and fields 1 through 4 must be empty.

Table 4-85. Command BD: Translate PIN and Generate ATM MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	BD
1	PIN block type	0, 1	1-5, 9
2	$E_{MFK.1}(KPE_I)$*	0, 16, 32	0 - 9, A - F
3	$E_{MFK.1}(KPE_O)$*	0, 16, 32	0 - 9, A - F
4	E_{KPE_I}(PIN Block)	0, 16, 18	0 - 9, A - F
5	$E_{MFK.V}$(KMAC)*	16, 32	0 - 9, A - F
6	MAC Length	1	0 - 3
7	[$E_{MFK.6}$(IV)]	0, 16	0 - 9, A - F
8	Insertion Position	0 - 4	0 - 9, L, F
9	Insertion Type	0, 1	A, B, E
10	Data Type	1	U or B
11	Data Length	1 - 4	0 - 9
12	Data	1 - 4096	0 - 9, A - F if unpacked ASCII
13	[Variant (V)]	0 - 2	3, 18
14	[PIN Block Data]**	Variable	

*Can be a volatile table location.

**See PIN Block Types earlier in this section.

Responding Parameters

CD

Field 0, the response identifier.

E_{KPE_O}(ANSI PIN block)

Field 1, the outgoing PIN block encrypted under the PIN Encryption Key. This field contains 16 hexadecimal characters. This field is empty if no PIN translation operation is performed.

Sanity Check Indicator

Field 2, the sanity check indicator. This test looks for synchronization between the sending and receiving nodes by checking for the existence of valid pad characters and PIN digits in the PIN block. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block.

This field can contain one of the following values:

- Y – PIN block passes the sanity check

- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`[IBM 3624 Sequence Number#]`

Field 3, the IBM 3624 sequence number. This field is returned only if the PIN block type is IBM 3624. When present, this field contains 2 hexadecimal characters.

`MAC Length`

Field 4, the length of the Message Authentication Code. The following table indicates the possible returned-MAC lengths and the corresponding codes that appear in this field. If this field is set to 0, then more data is expected and Field 2 will contain the ending Initialization Vector. If this field is set to 1, 2, or 3, then Field 2 will contain the Message Authentication Code.

Returned-MAC Size	Numerical Code
More data expected; no MAC returned	0
32 bits (eight characters)	1
48 bits (12 characters)	2
64 bits (16 characters)	3

A 32-bit Message Authentication Code is expressed as two groups of four hexadecimal digits, separated by a space. A 48- or 64-bit Message Authentication Code is expressed as three or four groups of four hexadecimal digits, separated by a space.

`MAC or` $E_{MFK.6}$`(Ending IV)`

Field 5, If Field 1 is set to 0, this field will contain the ending Initialization Vector encrypted under variant 6 of the MFK. If Field 1 is set to 1, 2, or 3, this field will contain the Message Authentication Code. If your use of this command results in the generation of an ending Initialization Vector in this field, use it as the starting initialization vector in the subsequent Message Authentication Code command to continue generating Message Authentication Code. This field contains a 9, 14, 16, or 19 byte hexadecimal value that contains hexadecimal values or spaces.

`KMAC Check Digits`

Field 6, check digits; the first four digits that result from encrypting zeros using the Message Authentication Code key. If option 88 is enabled, this field will contain the first six digits of the result.

KPE_I `Check Digits`

Field 7, check digits; the first four digits that result from encrypting zeros using the incoming PIN Encryption Key. If option 88 is enabled, this field will contain the first

six digits of the result. This field is empty if a sanity error is returned, or no PIN translation operation is performed.

`KPE`$_O$ `Check Digits#`

Field 8, check digits; the first four digits that result from encrypting zeros using the outgoing PIN Encryption Key. If option 88 is enabled, this field will contain the first six digits of the result. This field is empty if a sanity error is returned, or no PIN translation operation is performed.

Table 4-86. Response CD: Translate PIN and Generate ATM MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	CD
1	E_{KPE_O}(PIN block)	0, 16	0 - 9, A - F
2	Sanity check indicator	0, 1	Y, N, L
3	IBM 3624 Sequence Number*	0, 2	0 - 9, A - F
4	MAC Length	0, 1	0 - 3
5	MAC or $E_{MFK.6}$(Ending IV)	0, 9, 14, 16, 19	0 - 9, A - F, “ ”
6	KMAC Check Digits	0, 4 or 6	0 - 9, A - F
7	KPE_I Check Digits	0, 4 or 6	0 - 9, A - F
8	KPE_O Check Digits	0, 4 or 6	0 - 9, A - F

*Optional field; returned only if the PIN block type is IBM 3624.

Usage Notes

- Generate the incoming and outgoing PIN Encryption Keys.
- Generate the Message Authentication Code Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear text ANSI PIN block: 0412 26CB A987 6FED
- Clear-text incoming PIN Encryption Key: 07CE A74F 4607 5D8F.
  The incoming PIN Encryption Key encrypted under variant 1 of the MFK: 3B42CA4278E2DDE1.
- Clear-text outgoing PIN Encryption Key: D029 23D9 AD4F E90B.
  The outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 83CBEFA710C6639F
- Encrypted ANSI PIN block: 9AA4 3B94 C012 04F3

Verify Clear PIN (Command D0)

Command D0 verifies a clear-text PIN according to the technique you specify when you issue the command.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. You must purchase command 105, and then enable it in the Network Security Processor's security policy.

Command

```
<D0#Verification Method#0#PIN#Reserved#PIN Information#>
```

Response

```
<E0#Verification Flag#>[CRLF]
```

Calling Parameters

`D0`

Field 0, the command identifier.

`Verification Method`

Field 1, the PIN verification method. This field is 1 byte, it can contain the numbers 1, 2 or 3.

Verification Method	Numerical Code
Identikey	1
IBM 3624	2
VISA	3

`0`

Field 2, an indicator that this command is verifying a clear-text PIN.

`PIN`

Field 3, the clear-text PIN. This field contains a 4 to 12 byte decimal value.

`Reserved`

Field 4, reserved for future use. This field must be empty.

`PIN Information`

Field 5, identical to the fields from Command 32 starting at Field 5 and continuing until the second-to-last field.

Table 4-87. Command D0: Verify Clear PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	D0
1	Verification method	1	1 - 3
2	Clear-text PIN indicator	1	0
3	PIN	4 - 12	0 - 9
4	Reserved	0	
5	PIN information*		

*See Command 32 for parameter information.

Responding Parameters

`E0`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the PIN block is successfully verified or N if the PIN block is not successfully verified.

Table 4-88. Response EO: Verify Clear PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	EO
1	Verification flag	1	Y, N

Usage Notes

- This command is typically used for verifying host-based PINs in a proprietary network and for verifying PINs in on-others transactions initiated in a shared network.
- This command does not check the PIN to be sure its length is legal.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying clear-text PINs using the IBM 3624 method.

- Verification method: IBM 3624 (2).
- PIN: 3614 3614 3.
- PIN information:.
  - Conversion table: 8351 2964 7746 1538.
  - Offset: 6694 537.
  - Validation data: 3333 3333.
  - Pad character: D.
  - Check-length parameter: 7.
  - Clear-text PIN Verification Key: 89B0 7B35 A1B3 F47E.
    The PIN Verification Key encrypted under variant 4 of the MFK: BB79 3110 FD6D 9BB4.

The command looks like this:

```
<D0#2#0#361436143##8351296477461538#6694537#33333333#D#7#
BB793110FD6D9BB4#>
```

The Network Security Processor returns the following response:

```
<E0#Y#>
```

Generate Atalla 2x2 PVN (Command 11E)

Command 11E generates a PIN Verification Number using the Atalla 2x2 method.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

```
<11E#I#E_MFK.4(PIN Verification Key 1)#
E_MFK.4(PIN Verification Key 2)#PVN Format#PVN Length#
Data Type#Data Length#Data#>
```

Response

```
<21E#PVN#>[CRLF]
```

Calling Parameters

`11E`

Field 0, the command identifier.

`I`

Field 1, the Atalla 2x2 algorithm identifier. This field contains the letter I.

$E_{MFK.4}$`(PIN Verification Key 1)`

Field 2, the first PIN Verification Key encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value or a volatile table location.

$E_{MFK.4}$`(PIN Verification Key 2)`

Field 3, the second PIN Verification Key encrypted under variant 4 of the MFK. This field contains a 16 byte hexadecimal value or a volatile table location.

`PVN Format`

Field 4, this field specifies the format of the PVN. The choices are hexadecimal or decimal. This field should contain the letter H for hexadecimal format. For decimal format this field should contain the letter D, followed by the 16 byte decimalization table. If you use the default decimalization table of 0123456789012345, this field will contain only the letter D.

`PVN Length`

Field 5, defines the length of the generated PVN. This field contains a 1 to 2 byte decimal value in the range of 6 to 16.

`Data Type`

Field 6, the data type. The data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

`Data Length`

Field 7 defines the length of the data. This field contains a 2 byte decimal value in the range of 16 -24.

`Data`

Field 8, is the clear-text PIN and the 12 account number digits. This field contains 16 to 24 bytes decimal value.

Table 4-89. Command 11E: Generate Atalla 2x2 PVN

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	11E
1	Atalla 2x2 method	1	I
2	$E_{MFK.4}$(PIN Verification Key 1)*	16	0 - 9, A - F
3	$E_{MFK.4}$(PIN Verification Key 2)*	16	0 - 9, A - F
4	PVN Format	1, 17	H, D, 0 - 9
5	PVN Length	1, 2	6 - 16
6	Data Type	1	B, U
7	Data Length	2	16 - 24
8	Data	varies	0 - 9

* Can be a volatile table location.

Responding Parameters

`21E`

Field 0, the response identifier.

`PVN`

Field 1, the generated PVN. This field contains a 6 to 16 byte hexadecimal value.

Table 4-90. Response 21E: Generate Atalla 2x2 PVN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	21E
1	PVN	6-16	0 - 9, A - F

Usage Notes

- Generate the PIN Verification Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Generating a PVN using the Atalla 2x2 method.

- PVN method: Atalla 2x2 (I).
- Clear-text PIN Verification Key 1: 5555 6666 7777 8888.
  The PIN Verification Key 1 encrypted under variant 4 of the MFK: 953D 33E5 1F16 C884.
- Clear-text PIN Verification Key 2: 9999 AAAA BBBB CCCC.
  The PIN Verification Key 1 encrypted under variant 4 of the MFK: 9950 6F9B 9A69 E03F.
- PVN Format: Hexadecimal.
- PVN Length: 16
- Data Type: Unpacked ASCII hexadecimal
- Data Length: 18
- Data: PIN = 555555, Account Number 1234 1234 1234.

The command looks like this:

```
<11E#I#953D33E51F16C884#99506F9B9A69E03F#H#16#U#18#
555551234123412341234#>
```

The Network Security Processor returns the following response:

```
<21E#3436593F00F3C754#>
```

Calculate PIN Offset (Command 30A)

Command 30A uses the old account number and offset to determine the PIN. It then uses the PIN and the new account number to calculate the new IBM 3624 offset. This command supports either 1key-3DES (single-length) or 2key-3DES (double-length) PIN Verification Keys.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

<30A#$E_{MFK.4}$(OldKPV)#$E_{MFK.4}$(NewKPV)#Old Validation Data#
New Validation Data#[#Old Conversion Table#
New Conversion Table#]Old Offset#>

Response

<40A#New Offset#OldKPV Check Digits#
New KPV Check Digits#>[CRLF]

Calling Parameters

30A

Field 0, the command identifier.

$E_{MFK.4}$(OldKPV)

Field 1, the PIN Verification Key encrypted under variant 4 of the MFK. This key is used to generate the old offset. This field can contain a 16 or 32 byte hexadecimal value, or a volatile table index. When option 6A is enabled the OldKPV can be a replicated 1key-3DES (single-length) key.

$E_{MFK.4}$(NewKPV)

Field 2, the PIN Verification Key encrypted under variant 4 of the MFK. This key is used to generate the new offset. This field can contain a16 or 32 byte hexadecimal value, or a volatile table index. If the OldKPV is a 2key-3DES (double-length) key this key must also be 2key-3DES (double-length). If the OldKPV is a 1key-3DES (single-length) key this key can be either a 1key or 2key-3DES key. When option 6A is enabled the NewKPV can be a replicated 1key-3DES (single-length) key.

Old Validation Data

Field 3, the old validation data consists of the old account number digits used to generate the old offset. If less than 16 account number digits were used to

generate the old offset, this field must also contain the pad characters. This field must contain a 16 byte hexadecimal value.

`New Validation Data`

Field 4, the new validation data consists of the new account number digits. If less than 16 account number digits will be used to generate the new offset, this field must also contain the pad characters. This field must contain a 16 byte hexadecimal value.

`[Old Conversion Table#`

Field 5, the old Conversion Table is used to generate the old offset. This field is optional and is only required if the old Conversion Table is **not** 0123456789012345. If the new conversion table is not 0123456789012345 it must be supplied in field 6 and this field must exist but can be empty. If present, this field must be a 16 byte decimal value, a volatile table index, or an empty field if the old conversion table is 0123456789012345. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK. When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

`New Conversion Table#]`

Field 6, the new Conversion Table is used to generate the new offset. This field is optional and is only required if the new Conversion Table is **not** 0123456789012345 or if field 5 is provided in the command. If present, this field must be a 16 byte decimal value, a volatile table index, or an empty field if the new conversion table is 0123456789012345. When option 48 is enabled, this field contains a 16 hexadecimal character value (the conversion table encrypted under variant 6 of the MFK) or a volatile table location. Conversion Tables stored in the volatile table must be encrypted under variant 6 of the MFK. When option 4E is enabled, all three forms of the conversion table (clear-text, decrypted, or value stored in volatile table location) to be processed by the Network Security Processor must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

`Old Offset`

Field 7, the old offset. This field must contain a 4 to 12 byte decimal value.

Table 4-91. Command 30A: Calculate PIN Offset

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	30A
1	$E_{MFK.4}$(OldKPV)*	16, 32	0 - 9, A - F
2	$E_{MFK.4}$(NewKPV)*	16, 32	0 - 9, A - F
3	Old Validation Data	16	0 - 9, A - F
4	New Validation Data	16	0 - 9, A - F
5	[Old Conversion Table#*	0, 16*	0 - 9
6	New Conversion Table#]*	0, 16*	0 - 9
7	Old Offset	4 - 12	0 - 9

*Can be a volatile table location.

Responding Parameters

`40A`

Field 0, the response identifier.

`New Offset`

Field 1, the offset based on the PIN, new validation data, new Conversion Table, and new PIN Verification Key.

`OldKPV Check Digits`

Field 2, check digits; the first four digits that result from encrypting zeros using the oldKPV. If option 88 is enabled, this field will contain the first six digits of the result.

`NewKPV Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the newKPV. If option 88 is enabled, this field will contain the first six digits of the result.

Table 4-92. Response 40A: Calculate PIN Offset

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	40A
1	New Offset	4 - 12	0 - 9
2	OldKPV Check Digits	4 or 6	0 - 9, A - F
3	NewKPV Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Fields 5 and 6 are a pair, they either both do not exist and therefore the old and new conversion table will be 0123456789012345, or they both exist. They must both exist in these two scenarios:
  - If the old conversion table is a value other than 0123456789012345, then field 5 will contain the value of the conversion table and field 6 must also exist. If the new conversion table is 0123456789012345 field 6 can be empty. If the new conversion table is not 0123456789012345 it must be provided in field 6.
  - Similarly if the old conversion table is 0123456789012345 and the new conversion table is a different value, then field 5 must exist, but can be empty, and field 6 will contain the new conversion table.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

G**enerating a new offset**.

- Clear-text old PIN Verification Key: 1234123412341234. Check digits are C2F2. The old PIN Verification Key encrypted under variant 4 of the MFK: 2979 F655 1D00 84AC.
- Clear-text new PIN Verification Key: 4321432143214321. Check digits are 8149. The new PIN Verification Key encrypted under variant 4 of the MFK: 1741 AB42 8020 8D20.
- The old validation data is 0123 4567 89FF FFFF.
- The new validation data is 9876 5432 10FF FFFF.
- The old and new Conversion Table is 0123 4567 8901 2345.
- The old offset is 9920, (the clear-text PIN is 1234).

The command looks like this:

```
<30A#2979F6551D0084AC#1714AB4280208D20#0123456789FFFFFF#98765
43210FFFFFF#9920#>
```

The Network Security Processor returns the following response:

```
<40A#1313#C2F2#8149#>
```

Use default old conversion table and a different new conversion table

Same data as example above except that new conversion table is 9876543210543210

The command looks like this:

```
<30A#2979F6551D0084AC#1714AB4280208D20#0123456789FFFFFF#98765
43210FFFFFF##9876543210543210#9920#>
```

The Network Security Processor returns the following response:

```
<40A#6200#C2F2#8149#>
```

Verify ePIN (Command 32C)

Command 32C is used to verify the ePIN.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

```
<32C#Offset Format#E_MFK.4(KPV)#E_MFK.5(KOP)#ePIN#ePIN Object#>
```

Response

```
<42C#Verification Flag#KPV Check Digits#
KOP Check Digits#>[CRLF]
```

Calling Parameters

`32C`

Field 0, the command identifier.

`Offset Format`

Field 1, the offset format must be 2. This field contains 1 byte, the decimal value 2.

$E_{MFK.4}$`(KPV)`

Field 2, the PIN Verification Key encrypted under variant 4 of the MFK. This key is used to generate the ePIN offset. This field must contain a 32 byte hexadecimal value. When option 6A is enabled, this key can be a replicated 1key-3DES (single-length) key.

$E_{MFK.5}$`(KOP)`

Field 3, the Object PIN Key encrypted under variant 5 of the MFK. This key is used to decrypt the ePIN object.This field must contain a 32 byte hexadecimal value. When option 6A is enabled, this key can be a replicated 1key-3DES (single-length) key.

`ePIN`

Field 4, the entered ePIN. This field must contain a 16 byte hexadecimal value.

`ePIN Ojbect`

Field 5, the ePIN Object. This field must contain a 32 byte hexadecimal value.

Table 4-93. Command 32C: Verify ePIN Offset

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	32C
1	Offset Format	1	2
2	$E_{MFK.4}$(KPV)	32	0 - 9, A - F
3	$E_{MFK.5}$(KOP)	32	0 - 9, A - F
4	ePIN	16	0 - 9, A - F
5	ePIN Object	32	0 - 9, A - F

Responding Parameters

`42C`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field will contain either the letter Y if the ePIN verifies, or the letter N if the ePIN does not verify.

`KPV Check Digits`

Field 2, check digits; the first four digits that result from encrypting zeros using the PIN Verification Key. If option 88 is enabled, this field will contain the first six digits of the result.

`KOP Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the Object PIN Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 4-94. Response 42C: Verify ePIN Offset

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	42C
1	Verification Flag	1	Y, or N
2	KPV Check Digits	4 or 6	0 - 9, A - F
3	KOP Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Generate the PIN Verification and PIN Object Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an ePIN.

- Clear-text PIN Verification Key: 4321432143214321 1234123412341234. Check digits 2ABA. The PIN Verification Key encrypted under variant 4 of the MFK: 1741 AB42 8020 8D20 2979 F655 1D00 84AC.
- Clear-text Object PIN Key: 5678567856785678 8765876587658765. Check digits 686F. The Object PIN Key encrypted under variant 5: B2F1 19E3 78BA 85AB FDF9 C796 CE4A 12B7.
- ePIN: 314A41434B2A2A2A.
- ePIN Object: 27BDDE807F87DDD4589226D1F475CD0E

The command looks like this:

```
<32C#2#1741AB4280208D202979F6551D0084AC#
B2F119E378BA85ABFDF9C796CE4A12B7#314A41434B2A2A2A#
27BDDE807F87DDD4589226D1F475CD0E#>
```

The Network Security Processor returns the following response:

```
<42C#Y#2ABA#686F#>
```

PIN and PIN-Block Translate (Command 335)

Command 335 translates a PIN block from encryption under one key to encryption under a different key. This command can also change the PIN block type.

This command is enabled in the Network Security Processor's default security policy.

Command

<335#Incoming KPE Variant(I)#Incoming PIN Block Type#
Outgoing KPE Variant(O)#Outgoing PIN Block Type#
$E_{MFK.I}$(KPE$_I$)#$E_{MFK.O}$(KPE$_O$)#^{E}KPE$_I$(PIN Block)#
Incoming PIN Block Data#Outgoing PIN Block Data#>

Response

<435#KPE$_O$(Outgoing PIN Block)#
Sanity Check Indicator#KPE$_I$ Check Digits#KPE$_O$ Check Digits#
KC$_I$ Check Digits#KC$_O$ Check Digits#>

Calling Parameters

335

Field 0, the command identifier.

Incoming KPE Variant(I)

Field 1, the variant applied to the MFK to encrypt the incoming PIN Encryption Key (KPE$_I$). This field can contain one or two bytes, decimal values allowed are either 1 or 20 (decrypt only).

Incoming PIN Block Type

Field 2, the type of the incoming PIN block. This field is 1 byte, it can contain the numbers 1, 3, 8 or 9. When option 46 is enabled, this field can contain the value 1 (ANSI) or 8 (ISO-3).

PIN Block Type	Numerical Code
ANSI (ISO-0)	1
PIN/pad character / Docutel	3
ISO-3	8
IBM 4731	9

`Outgoing KPE Variant(O)`

Field 3, the variant applied to the MFK to encrypt the outgoing PIN Encryption Key (KPE_O). This field can contain one or two bytes, decimal values allowed are either 1 or 10 (encrypt only).

`Outgoing PIN Block Type`

Field 4, the outgoing PIN Block type. This field is 1 byte, it can contain the numbers 1, 3, 8 or 9. When option 47 is enabled, this field can contain the value 1 (ANSI) or 8 (ISO-3).

PIN Block Type	Numerical Code
ANSI (ISO-0)	1
PIN/pad character / Docutel	3
ISO-3	8
IBM 4731	9

$E_{MFK.I}$`(KPEI)`

Field 5, the incoming PIN Encryption Key (KPE_I) encrypted under variant 1 or 20 of the MFK. This field contains a 32 byte hexadecimal value, or a volatile table location. When option 6C is enabled, this field can be either 16 or 32 byte hexadecimal value. When option 6A is enabled and 32 characters are present in this field, the leftmost 16 and rightmost 16 characters may be the same.

$E_{MFK.O}$`(`KPE_O`)`

Field 6, the outgoing PIN Encryption Key (KPE_O) encrypted under variant 1 or 10 of the MFK. This field contains a 32 byte hexadecimal value, or a volatile table location. When option 6A is enabled and 32 characters are present in this field, the leftmost 16 and rightmost 16 characters may be the same. When option 49 is enabled, an error response is returned if the length of the (KPEo) is not equal to or greater than the length of the (KPEi).

$^{E}KPE_I$`(PIN Block)`

Field 7, the incoming PIN Block encrypted under the incoming PIN Encryption Key. This field contains 16 hexadecimal characters.

`Incoming PIN Block Data`

Field 8, Incoming PIN Block Data.

If the incoming PIN Block type is **ANSI or ISO-3**, this field will contain twelve bytes; the incoming PAN digits. When any of these options 46, 47 or 6B are enabled, an error is returned if the value in this field does not match the outgoing PIN block data. When option 6B is enabled, an error is returned if this field contains all zeros.

If the incoming PIN Block type is **PIN Pad**, this field will contain a one byte value; the pad character. Valid pad characters are a hexadecimal value, W, or X.

If the incoming PIN Block type is **IBM 4731**, this field will contain three fields:

- a one byte value; the pad character. Valid pad characters are a hexadecimal value, W, or X.
- The incoming ICV; a 16 byte hexadecimal value. When option 6B is enabled, an error is returned if this field contains all zeros or does not match the outgoing ICV.
- The incoming Communications Key (KC_I) encrypted under variant 3 of the MFK. This field contains 16 hexadecimal characters.

`Outgoing PIN Block Data`

Field 9, Outgoing PIN Block Data.

If the outgoing PIN Block type is **ANSI or ISO-3**, this field will contain the twelve bytes; the outgoing PAN digits. When option 6B is enabled, an error is returned if this field contains all zeros.

If the outgoing PIN Block type is **PIN Pad**, this field will contain one byte; the pad character. Valid pad characters are a hexadecimal value, W, or X.

If the outgoing PIN Block type is **IBM 4731**, this field will contain three fields:

- a one byte value; the pad character. Valid pad characters are a hexadecimal value, W, or X.
- The outgoing ICV; a 16 byte hexadecimal value. When option 6B is enabled, an error is returned if this field contains all zeros.
- The outgoing Communications Key (KC_O) encrypted under variant 3 of the MFK. This field contains 16 hexadecimal characters.

Table 4-95. Command 335: PIN and PIN-Block Translate (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	335
1	Incoming KPE Variant(I)	1, 2	1, 20
2	Incoming PIN Block Type	1	1, 3, 8, 9
3	Outgoing KPE Variant(O)	1, 2	1, 10
4	Outgoing PIN Block Type	1	1, 3, 8, 9
5	$E_{MFK.I}(KPE_I)$*	16**, 32	0 - 9, A - F
6	$E_{MFK.O}(KPE_O)$*	32	0 - 9, A - F
7	$^{E}KPE_I$(PIN Block)	16	0 - 9, A - F

Table 4-95. Command 335: PIN and PIN-Block Translate (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
8	Incoming PIN Block Data ANSI	12	0 - 9
	or		
	Incoming PIN Block Data PIN Pad	1	0 9, A- F, W, X
	or		
	Incoming PIN Block Data IBM 4731		
	- Incoming Pad	1	0 - 9, A- F, W, X
	- Field separator	1	#
	- Incoming ICV	16	0 - 9, A - F
	- Field separator	1	#
	- $E_{MFK.3}(KC_I)$	16	0 - 9, A - F
9	Outgoing PIN Block Data ANSI	12	0 - 9
	or		
	Outgoing PIN Block Data PIN Pad	1	0 9, A- F, W, X
	or		
	Outgoing PIN Block Data IBM 4731		
	- Outgoing Pad	1	0 - 9, A- F, W, X
	- Field separator	1	#
	- Outgoing ICV	16	0 - 9, A - F
	- Field separator	1	#
	- $E_{MFK.3}(KC_O)$	16	0 - 9, A - F

* Can be a volatile table location.

Responding Parameters

`435`

Field 0, the response identifier.

`E`KPE_O`(Outgoing PIN Block)`

Field 1, the outgoing PIN block encrypted under KPE_O. This field contains 16 hexadecimal characters. When a PIN sanity error is detected, the value in this field may not be correct. When a PIN sanity error is detected, and option 4B is enabled, this field will contain 16 zeros.

`Sanity Check`

Field 2, the sanity check indicator. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. This field contains one of following:

- Y – PIN block passes the sanity check.
- N – PIN block failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

`KPE$_I$ Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the incoming PIN Encryption Key (KPE_I). If option 88 is enabled, this field will contain the first six digits of the result.

`KPE$_O$ Check Digits`

Field 4, check digits; the first four digits that result from encrypting zeros using the outgoing PIN Encryption Key (KPE_O). If option 88 is enabled, this field will contain the first six digits of the result.

`[KC-I Check Digits#]`

Field 5, check digits; the first four digits that result from encrypting zeros using the incoming Communications Key (KC-I). If option 88 is enabled, this field will contain the first six digits of the result. This field is present only if the incoming PIN Block type is IBM 4731.

`[KC-O Check Digits#]`

Field 6, check digits; the first four digits that result from encrypting zeros using the outgoing Communications Key (KC-O). If option 88 is enabled, this field will contain the first six digits of the result. This field is present only if the outgoing PIN Block type is IBM 4731.

Table 4-96. Response 435: PIN and PIN-Block Translate

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	435
1	$^{E}KPE_O$(Outgoing PIN Block)	16	0 - 9, A - F
2	Sanity Check	1	Y, N, L
3	KPE_I Check Digits	4 or 6	0 - 9, A - F
4	KPE_O Check Digits	4 or 6	0 - 9, A - F
5	KC-I Check Digits*	0, 4 or 6	0 - 9, A - F
6	KC-O Check Digits*	0, 4 or 6	0 - 9, A - F

*This field exists only when either the incoming/outgoing PIN Block type is IBM 4731.

Usage Notes

- Generate the PIN Encryption Keys.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating an ANSI PIN block.

- Incoming Variant: 1
- Incoming PIN Block Type is ANSI; 1
- Outgoing Variant: 1
- Outgoing PIN Block Type is ANSI: 1
- Clear-text Incoming PIN Encryption Key: 4567 89AB CDEF 0123 0123 4567 89AB CDEF. Check Digits are F8DF. The Incoming PIN Encryption Key encrypted under variant 1 of the MFK: E1B2 08F8 83BF E780 AE86 D417 E64E 07E0.
- Clear-text Outgoing PIN Encryption Key: 6789 ABCD EF01 2345 FEDC BA98 7654 3210. Check Digits are 40B5. The Outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 3C8C 9C71 5402 06C9 BC62 A2AD 7251 6EA1.
- Clear-text incoming ANSI PIN block: 041226CBA9876FED.
  The incoming ANSI PIN block encrypted under the incoming PIN Encryption Key: BF8E 1569 561D D33E.
- Incoming PAN Digits: 1234 5678 9012.
- Outgoing PAN Digits: 1234 5678 9012.

The command looks like this:

```
<335#1#1#1#E1B208F883BFE780AE86D417E64E07E0#
3C8C9C71540206C9BC62A2AD72516EA1#
BF8E1569561DD33E#123456789012#123456789012#>
```

The Network Security Processor returns the following response:

```
<435#53F4660894A37C67#Y#F8DF#40B5###>
```

Translating an ANSI PIN block to an IBM 4731 PIN block.

- Incoming Variant: 1
- Incoming PIN Block Type is ANSI; 1
- Outgoing Variant: 1
- Outgoing PIN Block Type is IBM 4731: 9
- Clear-text Incoming PIN Encryption Key: 4567 89AB CDEF 0123 0123 4567 89AB CDEF. Check Digits are F8DF. The Incoming PIN Encryption Key encrypted under variant 1 of the MFK: E1B2 08F8 83BF E780 AE86 D417 E64E 07E0.

- Clear-text Outgoing PIN Encryption Key: 6789 ABCD EF01 2345 FEDC BA98 7654 3210. Check Digits are 40B5. The Outgoing PIN Encryption Key encrypted under variant 1 of the MFK: 3C8C 9C71 5402 06C9 BC62 A2AD 7251 6EA1.
- Clear-text incoming ANSI PIN block: 041226CBA9876FED.
  The incoming ANSI PIN block encrypted under the incoming PIN Encryption Key: BF8E 1569 561D D33E.
- Incoming PAN Digits: 1234 5678 9012.
- Outgoing Pad Character: F.
- ICV: 0123 4567 89AB CDEF.
- Clear-text Outgoing Communications Key: CDEF 0123 4567 89AB. Check Digits are E6D7. The Outgoing Communications Key encrypted under variant 3 of the MFK: 4D52 F329 F993 B11D.

The command looks like this:

```
<335#1#1#1#9#E1B208F883BFE780AE86D417E64E07E0#
3C8C9C71540206C9BC62A2AD72516EA1#BF8E1569561DD33E#
123456789012#F#0123456789ABCDEF#4D52F329F993B11D#>
```

The Network Security Processor returns the following response:

```
<435#07812362E48DC2CE#Y#F8DF#40B5##E6D7#>
```

Generate ePIN Offset (Command 37B)

Command 37B is used to generate the ePIN Offset. This command uses a form of the IBM 3624 algorithm to generate the ePIN Offset, which is contained within the ePIN Object.

You must purchase this command in the form of a command 105, and then enable it in the Network Security Processor's security policy.

Command

```
<37B#Offset Format#E_MFK.4(KPV)#E_MFK.5(KOP)#ePIN#PAN#>
```

Response

```
<47B#ePIN Object#KPV Check Digits#
KOP Check Digits#>[CRLF]
```

Calling Parameters

`37B`

Field 0, the command identifier.

`Offset Format`

Field 1, the offset format must be 2. This field contains 1 byte, the decimal value 2.

$E_{MFK.4}$`(KPV)`

Field 2, the PIN Verification Key encrypted under variant 4 of the MFK. This key is used to generate the ePIN offset. This field must contain a 32 byte hexadecimal value. When option 6A is enabled, this key can be a replicated 1key-3DES (single-length) key.

$E_{MFK.5}$`(KOP)`

Field 3, the Object PIN Key encrypted under variant 5 of the MFK. This key is used to decrypt the ePIN object.This field must contain a 32 byte hexadecimal value. When option 6A is enabled, this key can be a replicated 1key-3DES (single-length) key.

`ePIN`

Field 4, the ePIN. This field must contain a 16 byte hexadecimal value.

`PAN`

Field 5, the Primary Account Number. This field must contain a 16 byte hexadecimal value.

Responding Parameters

`47B`

Field 0, the response identifier.

`ePIN Object`

Field 1, the ePIN Object. This field will contain a 32 byte hexadecimal value.

`KPV Check Digits`

Usage Notes

- Generate the PIN Verification and PIN Object Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an ePIN.

- Clear-text PIN Verification Key: 4321 4321 4321 4321 1234 1234 1234 1234. The check digits are 2ABA. The PIN Verification Key encrypted under variant 4 of the MFK: 1741 AB42 8020 8D20 2979 F655 1D00 84AC.
- Clear-text Object PIN Key: 5678 5678 5678 5678 8765 8765 8765 8765. The check digits are 686F. The Object PIN Key encrypted under variant 5 of the MFK: B2F1 19E3 78BA 85AB FDF9 C796 CE4A 12B7.
- ePIN: 314A41434B2A2A2A.
- PAN: ABCD123456789012

The command looks like this:

```
<37B#2#1741AB4280208D202979F6551D0084AC#
B2F119E378BA85ABFDF9C796CE4A12B7#
314A41434B2A2A2A#ABCD123456789012#>
```

The Network Security Processor returns the following response:

```
<47B#27BDDE807F87DDD4589226D1F475CD0E#2ABA#686F#>
```

5 Processing Transaction Data

The Network Security Processor uses the Data Encryption Algorithm (DEA) as defined in the Data Encryption Standard (DES). See Federal information Processing Standard 46-3 for information on DES.

Processing transaction data using DES, involves three basic steps: encrypting, decrypting, and authenticating. This section explains data encryption and decryption. Section 6, Authenticating Transaction Data explains data authentication.

To skip this introduction, go to Table 5-1 for a list of commands.

Data Processing Tasks

Processing transaction data typically involves the following tasks:

- Establishing a Data Encryption/Decryption Key.
- Deciding which part of each message will be encrypted – the entire message or selected portions of it.
- Encrypting the data for network transmission.
- Transmitting the data.
- Decrypting it at the switch or issuer node.

Establishing a common Data Encryption/Decryption Key is discussed in Section 3, DES key management. Deciding which portions of the message to encrypt and transmitting the data are site-specific tasks that are not covered in this manual.

Encrypting and Decrypting Data

Encryption is the process of using a Data Encryption Key to scramble data so that it cannot be read by someone who does not know the key. Encryption provides privacy.

Supported Encryption/Decryption Methods

Data can be encrypted or decrypted using a variety of schemes. The Network Security Processor supports the following methods, see Federal Information Processing Standard 81 Modes of DES for more information.

- 3DES Cipher block chaining (CBC)
- Cipher block chaining (CBC)
- Cipher feedback, eight bits (CFB-8)
- Cipher feedback, 64 bits (CFB-64)
- Output feedback, 64 bits (OFB-64)
- Electronic Code Book (ECB) (CBC mode can be used indirectly to support ECB).

For the data encryption modes that Atalla supports, encryption can be expressed as the following function.

```
Encrypted data = ƒ(data, IV)
```

In other words, encryption is a function of data and an Initialization Vector. The next few paragraphs discuss Initialization Vectors.

Using Initialization Vectors

An Initialization Vector is a value that the Data Encryption Key uses during encryption to ensure that every clear-text string of data – including identical strings – is encrypted differently.

The following examples illustrate two ways that the device can be used to encrypt the data string and the role that Initialization Vectors play in each case.

Encryption All at Once

The first way the string can be encrypted is all at once. Thus, the starting string is “This is an idea.” and the starting Initialization Vector is X. (X is a randomly generated number.

$$ƒ(\text{This is an idea.}, E_{IV} = X)$$

The result is an encrypted string and the ending Initialization Vector, Y. The ending Initialization Vector is a value that depends on the Data Encryption Key and the data.

$$(\text{abcdefghijklmnop}, E_{IV} = Y)$$

This method of encrypting data – all at once, with a starting and ending Initialization Vector – is sufficient whenever you are working with messages that contain fewer than 4096 bytes of data.

Encryption in Batches

The second way the string can be encrypted is in batches. In this example, the starting string is, “This is” and the starting Initialization Vector is X. X is a randomly generated number.

$$ƒ(\text{This is }, E_{IV} = X)$$

The result is an encrypted string and the Initialization Vector, Z. The value of this Initialization Vector depends on both the Data Encryption Key and the data.

$$(\text{abcdefgh}, E_{IV} = Z)$$

To encrypt the rest of the string, supply the remainder of the data and Z, the Initialization Vector obtained when you encrypted the first part of the string.

$$ƒ(\text{an idea.}, E_{IV} = Z)$$

The result is an encrypted string and the ending Initialization Vector, Y. (Again, the value of this Initialization Vector depends on both the Data Encryption Key and the data.

$$(ijklmnop, E_{IV} = Y)$$

This method of encrypting data – in batches – must be used whenever you are working with messages that contain more than 4096 bytes of data. The Initialization Vector Z is called the **continuing Initialization Vector**. Z is the ending Initialization Vector for the first batch of data and the starting Initialization Vector for the next batch of data. Notice that both methods of encrypting data – all at once and in batches – have the same ending Initialization Vector, Y, because the same data was used. Y is dependent on both the key and the data.

Note. When encrypting data in batches, the length of the data encrypted in each batch – except for the last batch – must be a multiple of eight. The length of the last batch of data encrypted is not restricted to a multiple of 8.

Data Processing Commands

The rest of this section contains the command and response syntax for the Network Security Processor data processing commands.

Quick Reference

Table 5-1 identifies each command by number, name, and purpose. While the table organizes the data processing commands by category, the commands themselves are presented in numerical order.

Table 5-1. Data Processing Commands (page 1 of 2)

Command #	Name	Purpose
55	Data Encrypt, Decrypt, or Translate Link I to Link J	Encrypts clear data using one or two keys, decrypts single- or double-encrypted data, and translates single- or double-encrypted data. This command supports only the ECB mode of DES.
97	Encrypt/Decrypt Data	Encrypts clear data or decrypts ciphered data.
388	3DES DUKPT Encrypt/Decrypt Data	Encrypts clear data or decrypts ciphered data using a 3DES DUKPT data key.
94	Generate Initialization Vector	Generates an Initialization Vector.
95	Reformat Initialization Vector	Reformats an Initialization Vector for communicating on SNA networks.

Table 5-1. Data Processing Commands (page 2 of 2)

Command #	Name	Purpose
96	Verify Initialization Vector	Verifies the format and contents of Initialization Vectors transmitted and received on an SNA network.
93	Generate Random Number	Generates a random number.

Encrypt Or Decrypt Data Or Translate (Command 55)

Command 55 encrypts clear data using one or two keys, decrypts single- and double-encrypted data, and translates single- or double-encrypted data. The mode of DES used in this command is Electronic Code Book (ECB). This command supports only single-length working keys, it does not support triple DES.

This command is not enabled in the Network Security Processor's default factory security policy.

Command

<55#[$E_{MFK.2}(KC_{I1})$]#[$E_{MFK.2}(KC_{I2})$]#[$E_{MFK.2}(KC_{O1})$]#
[$E_{MFK.2}(KC_{O2})$]#Reserved#Data#Reserved#Reserved#Reserved#
Reserved#Reserved#>

Response

<65#Reserved#Data#[KC_{I1} Check Digits]#[KC_{I2} Check Digits]#
[KC_{O1} Check Digits]#[KC_{O2} Check Digits]#>[CRLF]

Calling Parameters

55

Field 0, the command identifier.

[$E_{MFK.2}(KC_{I1})$]

Field 1, the first incoming data-encryption key encrypted under variant 2 of the MFK. This key is used in the inner or first layer of encryption. If the input data is clear text, this field is empty. Otherwise, This field contains a 16 byte hexadecimal value, or a volatile table location.

[$E_{MFK.2}(KC_{I2})$]

Field 2, the second incoming data-encryption key encrypted under variant 2 of the MFK. This key is used in the outer or second layer of encryption. If the input data is single encrypted, this field is empty. Otherwise, This field contains a 16 byte hexadecimal value, or a volatile table location.

[$E_{MFK.2}(KC_{O1})$]

Field 3, the first outgoing data-encryption key encrypted under variant 2 of the MFK. This key is used in the inner or first layer of encryption. If the output data is clear text, this field is empty. Otherwise, This field contains a 16 byte hexadecimal value, or a volatile table location.

$[E_{MFK.2}(KC_{O2})]$

Field 4, the second outgoing data-encryption key encrypted under variant 2 of the MFK. This key is used in the outer or second layer of encryption. If the output data is single encrypted, this field is empty. Otherwise, This field contains a 16 byte hexadecimal value, or a volatile table location.

`Reserved`

Field 5, reserved for future use. Its current value is 00.

`Data`

Field 6, input data. This field contains a multiple of 16 hexadecimal characters (represented as ASCII characters in the command). The number of 16 character blocks is , where is 1 to 10.

`Reserved`

Field 7, reserved for future support of all modes of DES encryption. Its current value is 1111 to specify Electronic Code Book (ECB) for all cryptographic cycles.

`Reserved`

Fields eight to 11, reserved for future use. All four fields are empty.

Table 5-2. Command 55: Encrypt or Decrypt Data or Translate Link L to Link J

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	55
1	$E_{MFK.2}(KC_{I1})$*	0, 16	0 - 9, A - F
2	$E_{MKF.2}(KC_{I2})$*	0, 16	0 - 9, A - F
3	$E_{MFK.2}(KC_{O1})$*	0, 16	0 - 9, A - F
4	$E_{MFK.2}(KC_{O2})$*	0, 16	0 - 9, A - F
5	Reserved	2	00
6	Data**	16	0 - 9, A - F
7	Reserved	4	1111
8	Reserved	0	
9	Reserved	0	
10	Reserved	0	
11	Reserved	0	

*Can be a volatile table location.

**This field is a multiple of 16 hexadecimal characters (represented as ASCII characters in the command). The number of 16-character blocks is , where is from 1 to 10.

Responding Parameters

`65`

Field 0, the response identifier.

`Reserved`

Field 1, reserved for future use. This field currently will contain 00.

`Data`

Field 2, the output data. This field contains a multiple of 16 hexadecimal characters (represented as ASCII characters in the command). The number of 16 character blocks is , where is 1 to 10.

`[KC`$_{I1}$` Check Digits]`

Field 3, the first incoming Data Encryption Key's check digits; the first four digits that result from encrypting zeros using the first incoming Data Encryption Key. If option 88 is enabled, this field will contain the first six digits of the result. (If command Field 1 is empty, this field is also empty.

`[KC`$_{I2}$` Check Digits]`

Field 4, the second incoming Data Encryption Key's check digits; the first four digits that result from encrypting zeros using the second incoming Data Encryption Key. If option 88 is enabled, this field will contain the first six digits of the result. (If command Field 2 is empty, this field is also empty.

`[KC`$_{O1}$` Check Digits]`

Field 5, the first outgoing Data Encryption Key's check digits; the first four digits that result from encrypting zeros using the first outgoing Data Encryption Key. If option 88 is enabled, this field will contain the first six digits of the result. (If command Field 3 is empty, this field is also empty.

`[KC`$_{O2}$` Check Digits]`

Field 6, the second outgoing Data Encryption Key's check digits; the first four digits that result from encrypting zeros using the second outgoing Data Encryption Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 5-3. Response 65: Encrypt or Decrypt Data or Translate (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	65
1	Reserved	2	00
2	Data*	16	0 - 9, A - F
3	KC_{I1} Check Digits	0, 4 or 6	0 - 9, A - F

Table 5-3. Response 65: Encrypt or Decrypt Data or Translate (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	KC_{I2} Check Digits	0, 4 or 6	0 - 9, A - F
5	KC_{O1} Check Digits	0, 4 or 6	0 - 9, A - F
6	KC_{O2} Check Digits	0, 4 or 6	0 - 9, A - F

*This field is a multiple of 16 hexadecimal characters (represented as ASCII characters in the command). The number of 16 character blocks is , where is from 1 to 10.

Usage Notes

- Before using this command, generate the incoming and outgoing communications keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Translating data using two keys.

- Clear-text first incoming Data Encryption Key: 2222 2222 2222 2222; check digits are 0096. The first incoming Data Encryption Key encrypted under variant 2 of the MFK: 6B5B 659A 01B7 DA63.
- Clear-text second incoming Data Encryption Key: 3333 3333 3333 3333; check digits are ADC6. The second incoming Data Encryption Key encrypted under variant 2 of the MFK: C22F 5A1F 22D1 ABF1.
- Clear-text first outgoing Data Encryption Key: 4444 4444 4444 4444; check digits are E2F2. The first outgoing Data Encryption Key encrypted under variant 2 of the MFK: 28C5 CA15 146D ED01.
- Clear-text second outgoing Data Encryption Key: 5555 5555 5555 5555; check digits 0EE1. The second outgoing Data Encryption Key encrypted under variant 2 of the MFK: DABC C8F6 B302 0EE1.
- Data: 1234 5678 9012 3456.

The command looks like this.

```
<55#6B5B659A01B7DA63#C22F5A1F22D1ABF1#28C5CA15146DED01#
DABCC8F6B3020EE1#00#1234567890123456#1111#####>
```

The Network Security Processor issues the following response.

```
<65#00#89F8D54F1DA00CB6#0096#ADC6#E2F2#0CD7#>
```

Generate Random Number (Command 93)

Command 93 generates a random hexadecimal or decimal number.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<93#[Format#Length#]>
```

Response

```
<A3#Random Number#>[CRLF]
```

Calling Parameters

Note. Fields one and two are optional, however field 2 cannot exist without field 1. If you omit both fields, the command automatically generates a16 byte hexadecimal value.

`93`

Field 0, the command identifier.

`[Format#]`

Field 1, the random number's format. This field contains one byte; H for hexadecimal, or D for decimal.

`[Length#]`

Field 2, the number of digits in the random number. This field contains a 1 to 3 byte decimal value in the range of 4 - 128.

Table 5-4. Command 93: Generate Random Number

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	93
1	Format*	1	H, D
2	Length*	1 - 3	4 - 128

*Optional field

Responding Parameters

`A3`

Field 0, the response identifier.

`Random Number`

Field 1, the random number, a decimal or hexadecimal value.

Table 5-5. Response A3: Generate Random Number

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A3
1	Random number	4 - 128	0 - 9, A - F

Usage Notes

- Randomly generated hexadecimal values are typically used as Initialization Vectors.
- Randomly generated decimal numbers are typically used as challenge numbers.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

This command generates a random value your results will be different.

Generating a Random Number without Specifying Format or Length.

The command looks like this.

```
<93#>
```

The Network Security Processor issues the following response.

```
<A3#A23D79FEDB1329AB#>
```

Generating a Four Digit Random Hexadecimal Number

The command looks like this.

```
<93#H#4#>
```

The Network Security Processor issues the following response.

```
<A3#1A7B#>
```

Generating a Six Digit Random Decimal Number

The command looks like this.

```
<93#D#6#>
```

The Network Security Processor issues the following response.

```
<A3#327179#>
```

Generate Initialization Vector (Command 94)

Command 94 generates an Initialization Vector. This command supports only single-length working keys.

This command has high security exposure and is not enabled in the Network Security Processor's default security policy.

Command

<94#$E_{MFK.2}$(KD)#>

Response

<A4#E_{KD}(IV)#$E_{MFK.6}$(IV)#>[CRLF]

Calling Parameters

94

Field 0, the command identifier.

$E_{MFK.2}$(KD)

Field 1, the Data Encryption Key (KD) encrypted under variant 2 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Table 5-6. Command 94: Generate Initialization Vector

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	94
1	$E_{MFK.2}$(KD)*	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

A4

Field 0, the response identifier.

E_{KD}(IV)

Field 1, the generated Initialization Vector encrypted using the Data Encryption Key. This field contains a 16 byte hexadecimal value.

$E_{MFK.6}(IV)$

Field 2, the generated Initialization Vector encrypted under variant 6 of the MFK. This field contains a 16 byte hexadecimal value.

Table 5-7. Response A4: Generate Initialization Vector

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A4
1	$E_{KD}(IV)$	16	0 - 9, A - F
2	$E_{MFK.6}(IV)$	16	0 - 9, A - F

Usage Notes

- This command can be used to generate the Initialization Vector for the following encryption schemes: Cipher block chaining (CBC), cipher feedback – eight bits (CFB-8), cipher feedback – 64 bits (CFB-64), and output feedback – 64 bits (OFB-64).
- Before using this command, generate the Data Encryption Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

This command generates a random value your results will be different.

Generating an Initialization Vector.

- Clear-text Data Encryption Key (KD): 1A23 C4D5 E6F7 8913.
  The Data Encryption Key (KD) encrypted under variant 2 of the MFK: C935 4285 8519 DABF.

The command looks like this.

```
<94#C93542858519DABF#>
```

The Network Security Processor issues the following response.

```
<A4#73711F4C86EE0E5F#2D03E0CE90E4CA46#>
```

Reformat Initialization Vector (Command 95)

Command 95 reformats an Initialization Vector for communicating on SNA networks. This command supports only single-length working keys.

This command has high security exposure and is not enabled in the Network Security Processor's default security policy.

Command

<95#$E_{MFK.2}$(KD)#E_{KD}(IV)#>

Response

<A5#E_{KD}(Reformatted IV)#$E_{MFK.6}$(IV)#>[CRLF]

Calling Parameters

95

Field 0, the command identifier.

$E_{MFK.2}$(KD)

Field 1, the Data Encryption Key (KD) encrypted under variant 2 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

E_{KD}(IV)

Field 2, the Initialization Vector encrypted under the Data Encryption Key. This field contains a 16 byte hexadecimal value.

Table 5-8. Command 95: Reformat Initialization Vector

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	95
1	$E_{MFK.2}$(KD)*	16	0 - 9, A - F
2	E_{KD}(IV)	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

A5

Field 0, the response identifier.

E_{KD}(Reformatted IV)

Field 1, the reformatted Initialization Vector encrypted under the Data Encryption Key. This value is distributed on SNA networks and returned to the originating node for verification. The reformatted Initialization Vector is formed by taking the complement of the original Initialization Vector's first four bytes, then appending to this new value the original Initialization Vector's remaining bytes. This field contains a 16 byte hexadecimal value.

$E_{MFK.6}$(IV)

Field 2, the original Initialization Vector encrypted under variant 6 of the MFK. This field contains a 16 byte hexadecimal value.

Table 5-9. Response A5: Reformat Initialization Vector

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A5
1	E_{KD}(Reformatted IV)	16	0 - 9, A - F
2	$E_{MFK.6}$(IV)	16	0 - 9, A - F

Usage Notes

Perform the following tasks before using Command 95:

- Generate the Data Encryption Key.
- Generate the Initialization Vector.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Reformatting an Initialization Vector.

- Clear-text Data Encryption Key: 1A23 C4D5 E6F7 8913.
  The Data Encryption Key encrypted under variant 2 of the MFK: C935 4285 8519 DABF.
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 7371 1F4C 86EE 0E5F.

The command looks like this.

```
<95#C93542858519DABF#73711F4C86EE0E5F#>
```

The Network Security Processor issues the following response.

```
<A5#001531C92E907DF0#2D03E0CE90E4CA46#>
```

Verify Initialization Vector (Command 96)

Command 96 verifies the format and contents of an Initialization Vector transmitted and received on an SNA network.

This command has high security exposure and is not enabled in the Network Security Processor's default security policy.

Command

<96#$E_{MFK.6}$(IV)#$E_{MFK.2}$(KD)#E_{KD}(Reformatted IV)#>

Response

<A6#Verification Flag#>[CRLF]

Calling Parameters

96

Field 0, the command identifier.

$E_{MFK.6}$(IV)

Field 1, the Initialization Vector encrypted under variant 6 of the MFK. This field contains a 16 byte hexadecimal value.

$E_{MFK.2}$(KD)

Field 2, the Data Encryption Key encrypted under variant 2 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

E_{KD}(Reformatted IV)

Field 3, the reformatted Initialization Vector encrypted under the Data Encryption Key. This field contains a 16 byte hexadecimal value.

Table 5-10. Command 96: Verify Initialization Vector

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	96
1	$E_{MFK.6}$(IV)	16	0 - 9, A - F
2	$E_{MFK.2}$(KD)*	16	0 - 9, A - F
3	E_{KD}(Reformatted IV)	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`A6`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the Initialization Vectors are identical; otherwise, it returns N.

Table 5-11. Response A6: Verify Initialization Vector

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A6
1	Verification flag	1	Y, N

Usage Notes

Perform the following tasks before using Command 96:

- Generate the Initialization Vector.
- Generate the Data Encryption Key.
- Generate the reformatted Initialization Vector.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying an Initialization Vector.

- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Clear-text Data Encryption Key: 1A23 C4D5 E6F7 8913.
  The Data Encryption Key encrypted under variant 2 of the MFK: C935 4285 8519 DABF.
- Clear-text reforrmatted Initialization Vector: DAA7 7AAD 2558 8552.
  The reformatted Initialization Vector encrypted under the Data Encryption Key: 0015 31C9 2E90 7DF0.

The command looks like this.

```
<96#2D03E0CE90E4CA46#C93542858519DABF#001531C92E907DF0#>
```

The Network Security Processor issues the following response.

```
<A6#Y#>
```

Encrypt/Decrypt Data (Command 97)

Command 97 encrypts clear data or decrypts encrypted data. Several DES methods are supported including 3DES. Both binary and ASCII hexadecimal data types are supported.

If the DES method is 3DES and option 6A is enabled, this command will support a replicated single-length key. If option 6A is disabled, which is the default, and the DES method is 3DES, this command requires a true 2key-3DES (double-length) key. All other DES methods support only single-length keys.

This command has high security exposure and is not enabled in the Network Security Processor's default security policy.

Command

<97#Operation#DES Method#$E_{MFK.V}$(KD)#[$E_{MFK.6}$(IV)]#Data Type#
Length#Data#[Variant#]>

Response

<A7#Operation#DES Method#KD Check Digits#$E_{MFK.6}$(IV)#
$E_{MFK.6}$(Ending IV)#Data Type#Length#Data#>[CRLF]

Calling Parameters

97

Field 0, the command identifier.

Operation

Field 1, indicates the operation to be performed on the data. This field contains 1 byte, either E to indicate encryption, or D to indicate decryption.

DES Method

Field 2, the DES method for encryption or decryption are:

DES Method	Value
Cipher block chaining (CBC) (single-length DES)	1
Cipher feedback – eight bits (CFB-8)	2
Cipher feedback – 64 bits (CFB-64)	3
Output feedback – 64 bits (OFB-64)	4
3DES Cipher block chaining (CBC)	6

$E_{MFK.2}(KD)$

Field 3, the Data Key encrypted under variant 2 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location. If the DES method specified in field 2 is 3DES (6) this field contains a 32 byte hexadecimal value, or volatile table location. The KD can also be encrypted under MFK variants 16 or 17.

$[E_{MFK.6}(IV)]$

Field 4, the Initialization Vector encrypted under variant 6 of the MFK.

If the operation is Encryption, this field can be:

- Empty, in which case a randomly generated Initialization Vector is used.
- Contain the letter “D”; use a default IV of all zeros.
- A 16 byte hexadecimal value of the Initialization Vector encrypted under variant 6 of the MFK.

If the operation is Decryption, this field can be either:

- Contain the letter “D”: use a default IV of all zeros.
- A 16 byte hexadecimal value of the Initialization Vector encrypted under variant 6 of the MFK.

Data Type

Field 5, the data types are:

Data Type	Type
Unpacked ASCII hexadecimal	U
Binary	B

See Data formats on page 1-4 for more information on data types.

Length

Field 6, the data's length. This command will encrypt or decrypt up to 4096 bytes of data. For all methods of encryption, except CFB-8, the data will be padded with zeros by the Network Security Processor, to achieve an 8 byte multiple. For all methods of decryption, except CFB-8, the data length must be an 8 byte multiple.

Since CFB-8 operates on 8 bit values the minimum binary data length is 1, and the minimum unpacked ASCII data length is 2.

The maximum data length is 4096. This field contains a1 to 4 byte decimal value.

Data

Field 7, the input data, encrypted or in clear-text format. The length of this field must match what was specified in field 6.

`[Variant#]`

Field 8, the variant used to encrypt the Data Key. This field is optional. Specify 16 if the key is used only for Encryption, or 17 if the key is used only for Decryption.

Table 5-12. Command 97: Encrypt/Decrypt Data

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	97
1	Operation	1	D, E
2	DES method	1	1 - 4, 6
3	$E_{MFK.V}(KD)$*	16, 32	0 - 9, A - F
4	$E_{MFK.6}(IV)$	0, 1, 16	0 - 9, A - F
5	Data type	1	U, B
6	Length	1 - 4	0 - 9
7	Data	1 - 4096	0 - 9, A - F, if unpacked ASCII
8	Variant**	0, 1, 2	2, 16, 17

*Can be a volatile table location.

**Optional field; if this field does not exist, the Network Security Processor uses the default variant, 2.

Responding Parameters

`A7`

Field 0, the response identifier.

`Operation`

Field 1, the operation performed on the data: Encryption (E) or Decryption (D). This field will contain the value specified in field 1 of the command.

`DES Method`

Field 2, the DES method of encryption or decryption used. This field will contain the value specified in field 2 of the command.

`KD Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the Data Key. If option 88 is enabled, this field will contain the first six digits of the result.

$E_{MFK.6}(IV)$

Field 4, the Initialization Vector encrypted under variant 6 of the MFK. The Initialization Vector is specified in the command or generated by the Network

Security Processor. This field contains a 16 byte hexadecimal value. The letter D will be returned in this field if a D was supplied in field 4 of the command.

$E_{MFK.6}$(Ending IV)

Field 5, the ending Initialization Vector encrypted under variant 6 of the MFK. This ending Initialization Vector must be used as the starting Initialization Vector for the next block of data if the amount of data to be encrypted or decrypted will not fit in one command. This field contains a 16 byte hexadecimal value.

Data Type

Field 6, the type of data returned in Field 8: unpacked ASCII hexadecimal or binary. This field will contain the value specified in field 5 of the command.

Length

Field 7, the length of the returned data. This field contains a 1 to 4 byte decimal value.

If you performed an encryption in this command, then the length returned here may be longer than the clear-text data length. When encrypting data, except when in CFB-8 mode, the Network Security Processor pads the input with zeros to achieve an 8 byte multiple.

Data

Field 8, the encrypted or decrypted data. This field can be from 1 to 4096 bytes long.

Table 5-13. Response A7: Encrypt/Decrypt Data

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A7
1	Operation	1	D, E
2	DES method	1	1 - 4, 6
3	KD Check Digits	4 or 6	0 - 9, A - F
4	$E_{MFK.6}$(IV)	1 or 16	0 - 9, A - F
5	$E_{MFK.6}$(Ending IV)	16	0 - 9, A - F
6	Data type	1	U, B
7	Length	1 - 4	0 - 9
8	Data	8 - 4096	0 - 9, A - F if unpacked ASCII

Usage Notes

- If you are encrypting or decrypting large amounts of data, you should specify the ending Initialization Vector, returned in the response for the first block of data, to be

the starting Initialization Vector in the encryption or decryption for the next block of data. Be sure to specify the same key and DES method for each data block.

- Before using this command, generate the Data Key.
- Before using this command, generate the Initialization Vector.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Encrypting Data

Encrypting 8 bytes of data using the default variant, 2.

- Encryption method: Cipher block chaining (1).
- Clear-text Data Key: 1A23 C4D5 E6F7 8913.
  The Data Key encrypted under variant 2 of the MFK: C935 4285 8519 DABF.
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 16.
- The data: 4E6F 7720 6973 2074.

The command looks like this.

```
<97#E#1#C93542858519DABF#2D03E0CE90E4CA46#U#16#
4E6F772069732074#>
```

The Network Security Processor issues the following response.

```
<A7#E#1#BC59#2D03E0CE90E4CA46#8EA7C883432745D3#U#16#
D2D442D8713E99F2#>
```

Decrypting Data

- Decryption method: Cipher block chaining (1).
- Clear-text Data Key: 1A23 C4D5 E6F7 8913.
  The Data Key encrypted under variant 2 of the MFK: C935 4285 8519 DABF.
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 16.

- The data: D2D4 42D8 713E 99F2.
- Variant: two.

The command looks like this.

```
<97#D#1#C93542858519DABF#2D03E0CE90E4CA46#U#16#
D2D442D8713E99F2#2#>
```

The Network Security Processor issues the following response.

```
<A7#D#1#BC59#2D03E0CE90E4CA46#8EA7C883432745D3#U#16#
4E6F772069732074#>
```

Encrypting Data Using Variant 16

- Encryption method: Cipher block chaining (1).
- Clear-text Data Key: 1A23 C4D5 E6F7 8913.
  The Data Key encrypted under variant 16 of the MFK: 6646 E8FB 9599 2446.
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 16.
- The data: 4E6F 7720 6973 2074.
- Variant: 16.

The command looks like this.

```
<97#E#1#6646E8FB95992446#2D03E0CE90E4CA46#U#16#
4E6F772069732074#16#>
```

The Network Security Processor issues the following response.

```
<A7#E#1#BC59#2D03E0CE90E4CA46#8EA7C883432745D3#U#16#
D2D442D8713E99F2#>
```

Decrypting Data Using Variant 17

- Decryption method: Cipher block chaining (1).
- Clear-text Data Key: 1A23 C4D5 E6F7 8913.
  The Data Key encrypted under variant 17 of the MFK: C0DE F3E3 15CB D1EC.
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 16.

- The data: D2D4 42D8 713E 99F2.
- Variant: 17.

The command looks like this.

```
<97#D#1#C0DEF3E315CBD1EC#2D03E0CE90E4CA46#U#16#
D2D442D8713E99F2#17#>
```

The Network Security Processor issues the following response.

```
<A7#D#1#BC59#2D03E0CE90E4CA46#8EA7C883432745D3#U#16#
4E6F772069732074#>
```

Encrypting Data using the 3DES CBC method

- Encryption method: 3DES Cipher block chaining (6).
- Clear-text Data Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The Data Key encrypted under variant 2 of the MFK: 80BC DEAC 5703 BC84 B888 0E5C 66D2 1760.
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 16.
- The data: 4E6F 7720 6973 2074.

The command looks like this.

```
<97#E#6#80BCDEAC5703BC84B8880E5C66D21760#2D03E0CE90E4CA46#
U#16#4E6F772069732074#>
```

The Network Security Processor issues the following response.

```
<A7#E#6#08D7#2D03E0CE90E4CA46#A7D1E8EF41BB45A2#U#16#
BD5F913518727778#>
```

Encrypting Data in multiple blocks

This example will encrypt a total of 16 bytes of data using two data encrypt commands. Notice that the ending IV returned in the response from the first command will be used as the IV in the second command.

Encrypting 8 bytes of data using the default variant, 2.

- Encryption method: Cipher block chaining (1).
- Clear-text Data Key: 1A23 C4D5 E6F7 8913.
  The Data Key encrypted under variant 2 of the MFK: C935 4285 8519 DABF.

- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U).
- Data length for the first command : 16.
- Data for the first command : 4E6F 7720 6973 2074 .

The first command looks like this.

```
<97#E#1#C93542858519DABF#2D03E0CE90E4CA46#U#16#
4E6F772069732074#>
```

The Network Security Processor issues the following response.

```
<A7#E#1#BC59#2D03E0CE90E4CA46#8EA7C883432745D3#U#16#
D2D442D8713E99F2#>
```

- Data length for the second command : 16.
- Data for the second command : 77CF BD32 8C8F 09AE.
- Ending IV from the first command:

The second command looks like this.

```
<97#E#1#C93542858519DABF#8EA7C883432745D3#U#16#
77CFBD328C8F09AE#>
```

The Network Security Processor issues the following response.

```
<A7#E#1#BC59#8EA7C883432745D3#F462AA88E4DD7854#U#16#
51A5D6FDE32D3CB4#>
```

Here is the equivalent single command to encrypt the same 16 bytes. Notice that the encrypted data in this response is the same as the encrypted data result from the first command, concatenated with the encrypted data result in the second command.

```
<97#E#1#C93542858519DABF#2D03E0CE90E4CA46#U#32#
4E6F77206973207477CFBD328C8F09AE#>
```

The Network Security Processor issues the following response.

```
<A7#E#1#BC59#2D03E0CE90E4CA46#F462AA88E4DD7854#U#32#
D2D442D8713E99F251A5D6FDE32D3CB4#>
```

3DES DUKPT Encrypt/Decrypt Data (Command 388)

Command 388 uses the Derived Unique Key Per Transaction (DUKPT) algorithm, a base derivation key, and a key serial number to generate the current key. A one-way function is applied to the current key to generate a session data key. This generated session data key is then used to either encrypt or decrypt data. Cipher Block Chaining (CBC) and Electronic Code Book (ECB) modes of 3DES are supported. The clear or encrypted data must be provided as ASCII hexadecimal characters. Binary data is not supported.

This command has high security exposure and is not enabled in the Network Security Processor's default security policy.

Command

<388#Operation#Mode#$E_{MFK.8}$(BDK)#Key Serial Number#[IV]#Data#>

Response

<488#Data#Ending IV#Base Derivation Key Check Digits#
Data Key Check Digits#>[CRLF]

Calling Parameters

388

Field 0, the command identifier.

Operation

Field 1, indicates the operation to be performed on the data. This field contains one letter, either E to indicate encryption, or D to indicate decryption.

Mode

Field 2, the 3DES mode used to encrypt or decrypt the data are:

Mode	Value
Electronic Code Book (ECB)	0
Cipher Block Chaining (CBC)	1

$E_{MFK.8}$(BDK)

Field 3, the Base Derivation Key (BDK) encrypted under variant 8 of the MFK. This field contains a 32 hexadecimal digit value or a volatile table location. The BDK must be a 2key-3DES key, not a replicated single-length key.

`Key Serial Number`

Field 4, the key serial number used to generate the session data key. This field contains a 10 - 20 hexadecimal digit value. Leading Fs must not be included in this field.

`[IV]`

Field 5, the Initialization Vector.

This field must be empty when the mode is ECB (field 2 contains the number 0).

When the mode is CBC (field 2 contains the number 1), this field must contain 16 hexadecimal digits.

If the amount of data to be encrypted or decrypted exceeds the 4096 hexadecimal digit limit, the data must be split into segments. Each segment is sent in a separate command 388. All commands after the first command in the chain must contain the ending IV which was returned in field 2 of the response to the previous command 388.

`Data`

Field 6, the input data, clear-text or encrypted.

This field must contain clear-text data when field 1 contains the letter E. The length of clear-text data to be encrypted must be within the range of 2 - 4096 hexadecimal digits. If the length of the clear-text input data is not a multiple of 16 the Network Security Processor will right-pad the data with zeros such that the resulting length will be a multiple of 16.

This field must contain encrypted data when field 1 contains the letter D. The length of the encrypted data to be decrypted must be within the range of 16 - 4096 hexadecimal digits and be a multiple of 16.

If the amount of data to be encrypted or decrypted exceeds the 4096 hexadecimal digit limit, the data must be split into segments. Each segment is sent in a separate command 388. When encrypting data, the length of all segments except the last segment must be a multiple of 16, the last segment can be any length as long as it is not greater than 4096, this prevents the Network Security Processor from padding the intermediate segment data with zeros. When decrypting data, the length of all data segments must be a multiple of 16.

Table 5-14. Command 388: 3DES DUKPT Encrypt/Decrypt Data (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	388
1	Operation	1	D, E
2	Mode	1	0, 1
3	$E_{MFK.8}$(BDK)*	32	0 - 9, A - F

Table 5-14. Command 388: 3DES DUKPT Encrypt/Decrypt Data (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	Key Serial Number	10 - 20	0 - 9, A - F
5	IV	0, 16	0 - 9, A - F
6	Data	2 - 4096	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`488`

Field 0, the response identifier.

`Data`

Field 1, the encrypted or clear-text data.

This field will contain clear-text data when field 1 of the command contains the letter D. The size of the clear-text data will be in the range of 16 - 4096 hexadecimal digits. It is the responsibility of the host application to validate/remove any padding.

This field will contain encrypted data when field 1 of the command contains the letter E. The size of the encrypted data will be in the range of 16 - 4096 hexadecimal digits.

`Ending IV`

Field 2, the ending Initialization Vector. This ending IV must be used as the starting IV for the next block of data if the amount of data to be encrypted or decrypted is greater than 4096 hexadecimal digits and the 3DES mode is CBC. This field contains a 16 hexadecimal digit value which is the last 16 hexadecimal digits of response field 1. It is included merely for convenience.

`Base Derivation Key Check Digits`

Field 3, check digits; that is, the first four hexadecimal digits that result from encrypting zeros using the Base Derivation Key. If option 88 is enabled, this field will contain the first six hexadecimal digits of the result.

`Data Key Check Digits`

Field 4, check digits; that is, the first four hexadecimal digits that result from encrypting zeros using the generated session data key. If option 88 is enabled, this field will contain the first six hexadecimal digits of the result.

Table 5-15. Response 488: 3DES DUKPT Encrypt/Decrypt Data

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	488
1	Data	2 - 4096	0 - 9, A - F
2	Ending IV	16	0 - 9, A - F
3	BDK Check Digits	4 or 6	0 - 9, A - F
4	Data Key Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- If you are encrypting or decrypting large amounts of data, using the CBC mode you should specify the ending Initialization Vector, returned in the response for the first block of data, to be the starting Initialization Vector in the encryption or decryption for the next block of data. Be sure to specify the same Base Derivation Key, Key Serial Number and mode for each data block.
- Before using this command, generate the Base Derivation Key encrypted under variant 8 of the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Encryption

- Operation: E
- Mode: Cipher block chaining (1).
- Base Derivation Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits 08D7
  The Base Derivation Key encrypted under variant 8 of the MFK
  AAA57E4E99AE9B0328F6BA950E1664FA
- Key Serial Number: FFFF9876543210E00001
- IV: 0000000000000000
- Data: 4E6F772069732074495D0F96DCF42F40.

The command looks like this.

```
<388#E#1#AAA57E4E99AE9B0328F6BA950E1664FA#9876543210E00001#00
0000000000000#4E6F772069732074495D0F96DCF42F40#>
```

The Network Security Processor returns the following response:

```
<488#98EFA6D1AAC43A805A0B7F205A8808E1#5A0B7F205A8808E1#08D7#1
56B#>
```

Decryption

- Operation: D
- Mode: Cipher block chaining (1).
- Base Derivation Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits 08D7
  The Base Derivation Key encrypted under variant 8 of the MFK
  AAA57E4E99AE9B0328F6BA950E1664FA
- Key Serial Number: FFFF9876543210E00001
- IV: 0000000000000000
- Data: 98EFA6D1AAC43A805A0B7F205A8808E1.

The command looks like this.

```
<388#D#1#AAA57E4E99AE9B0328F6BA950E1664FA#9876543210E00001#00
00000000000000#98EFA6D1AAC43A805A0B7F205A8808E1#>
```

The Network Security Processor returns the following response:

```
<488#4E6F772069732074495D0F96DCF42F40#495D0F96DCF42F40#08D7#1
56B#>
```

6 Authenticating Transaction Data

Data authentication is the process of verifying transmitted data to be sure that it has not been altered during transmission. Authentication thus ensures data integrity. This section outlines the tasks involved in authenticating data.

Federal information Processing Standard 113, and ANSI X9.9 provide detailed information on Message Authentication.

To skip this introduction go to Table 6-1 for a list of commands.

About Data Authentication

A Message Authentication Code (MAC) is used to validate that data has not been altered. The node sending the data, generates a MAC by applying a special, predefined algorithm and a data authentication key to a block of data, the result is the MAC. The data and MAC are sent to the receiving node. The receiving node then applies the same algorithm and key to compute a MAC for the data it receives. If the computed MAC matches the received one, then the data has not been altered during transmission.

The Network Security Processor can authenticate an unlimited amount of data. If you will be sending or verifying a large amount of data (more than 4096 bytes) using Commands 98 or 99, then you must send the data in more than one batch. Sending data in one or multiple batches is explained in Authentication in Batches on page 6-2.

Data Authentication Tasks

Authenticating data typically involves the following tasks:

- Generating the MAC to be transmitted with the data.
- Verifying the MAC at the receiving end.

Whether you are generating or verifying a MAC, the steps involved are the same: generate a MAC key and specifying its cryptogram as a parameter in the appropriate MAC generating or verifying command. The response contains either the MAC or a verification flag. When sending or authenticating a large volume of data – necessitating the use of the MAC continuation commands – a MAC or verification flag is returned on the last block of data.

Data authentication can be expressed as the following function:

Authenticated data = f(data, IV)

In other words, authentication is a function of data and an Initialization Vector. The following section explains how Initialization Vectors are used in data authentication.

Authentication All at Once

The first way the string can be sent or authenticated is all at once. Thus, the starting string is, “This is an idea.” The starting Initialization Vector is all zeros.

f(This is an idea.)

When generating a MAC, the result is a MAC and the ending Initialization Vector, Y. When verifying a MAC, the response is a verification flag and the ending Initialization Vector.

Generating: (MAC, E_{IV} = X)

Verifying: (Flag – Y or N, E_{IV} = X)

This method of authenticating data is sufficient when using Command 98 to send messages that contain fewer than 4096 bytes of data.

Authentication in Batches

The second way the string can be sent or authenticated is in batches. In this example, the starting string is, “This is” and the starting Initialization Vector is all zeros.

f(This is)

The result is the Initialization Vector, Z. A MAC or verification flag is returned on the last block of data.

(E_{IV} = Z)

To send or authenticate the rest of the string, you supply the remainder of the sentence and Z, the Initialization Vector obtained when you authenticated the first part of the string.

f(an idea., E_{IV} = Z)

If you are on the sending node – thus, generating a MAC, then the result is a MAC and the ending Initialization Vector, X. If you are verifying a MAC, then the Network Security Processor returns a verification flag and the ending Initialization Vector.

Generating: (MAC, E_{IV} = X)

Verifying: (Flag -- Y or N, E_{IV} = X)

This method of sending or authenticating data – in batches – must be used when you are using command to send messages that contain more than 4096 bytes of data.

Note. When authenticating data in batches, the length of the data authenticated in each batch – except for the last batch – must be a multiple of eight bytes for binary data, and a multiple of 16 bytes for unpacked ASCII data. The length of the last batch of data authenticated is not restricted.

Verification in VISA UKPT Networks

VISA UKPT (Unique Key Per Transaction) key management uses MACs, but implements them a little differently from the process just described. Specifically, VISA UKPT requires three authentication codes: MAC one, MAC two, and MAC three. MAC one authenticates the transaction data received from the PIN pad. The host calculates and returns MAC two if the transaction is approved; if the transaction is denied, then the host returns MAC three. See Verify and Generate MAC for VISA UKPT (Command 5C) on page 6-25 for the command syntax.

Data Authentication Commands

The rest of this section contains the command and response syntax for the Network Security Processor data authentication commands.

Quick Reference

The following identifies each command by number, name, and purpose. While Table 6-1 organizes the message authentication commands by category, the commands themselves are presented in numerical order.

Table 6-1. Data Authentication Commands (page 1 of 2)

Command #	Name	Purpose
59	Generate MAC and Encrypt or Translate Data	Generates a Message Authentication Code and can either encrypt or translate data.
98	Generate Message Authentication Code	Generates a Message Authentication Code.
386	Generate DUKPT Message Authentication Code	Generates a Message Authentication Code using a Derived Unique Key per Transaction Key.
—	MAC Translate	Verifies a Message Authentication Code, then generates a Message Authentication Code using a different key.
5C	Verify & Generate MAC for VISA UKPT	Verifies a Message Authentication Code and generates an approval or denial Message Authentication Code.
5F	Verify MAC and Decrypt PIN	Verifies a Message Authentication Code and decrypts a PIN.

Table 6-1. Data Authentication Commands (page 2 of 2)

Command #	Name	Purpose
99	Verify Message Authentication Code	Verifies a Message Authentication Code.
9B	Verify ACR Response MAC	Verifies a Message Authentication Code from an Atalla Challenge Response Unit.
348	Verify DUKPT Message Authentication Code	Verifies a Message Authentication Code that was generated using a Derived Unique Key per Transaction Key.

MAC Translate (Command 58)

Command 58 translates a Message Authentication Code from one key to another key. It first verifies an incoming MAC using the incoming MAC key designated as KMAC-I and if successful, generates a MAC using the outgoing MAC key designated as (KMAC-O.)

This command can also be used to either only verify a MAC, or only generate a MAC. If this command will be used to only verify a MAC command fields 6, 7, 8 and 9 must be empty. If this command will be used to only generate a MAC command fields 1, 2, 3, 4, and 5 must be empty.

This command supports only 1key-3DES (single-length) working keys. It has high security exposure and is not enabled in the default security policy.

Command

<58#[$E_{MFK.I}$(KMAC-I)]#[MAC-I Length]#[$E_{MFK.6}$(IV-I)]#
[Incoming Variant]#[MAC-I]#[$E_{MFK.O}$(KMAC-O)]#[MAC-O Length]#
[$E_{MFK.6}$(IV-O)]#[Outgoing Variant]#Data Type#Data Length#Data#>

Response

<68#[MAC Length-I]#[Verification Flag or $E_{MFK.6}$(Ending IV-I)]#
[KMAC-I Check Digits]#[MAC Length-O]#
[MAC or $E_{MFK.6}$(Ending IV-O)]#[KMAC-O Check Digits]#>

Calling Parameters

58

Field 0, the command identifier.

[$E_{MFK.V}$(KMAC-V)]

Field 1, the incoming MAC key (KMAC-I) encrypted under variant 3 or 19 of the MFK. This field contains a 16 byte hexadecimal value, a volatile table location, or is empty.

If this field is empty, then the Fields 2, 3, 4 and 5 must also be empty.

[MAC-I Length]

Field 2, the size of the MAC to be verified. The following table indicates the possible MAC sizes and the codes to enter in this field.

MAC Size	Code
More data expected; no MAC verified	0
32 bits	1
48 bits	2
64 bits	3

A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A-F) and written as two groups of four digits, separated by a space. A 48 bit or 64 bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

This field can contain a 1 byte decimal value or is empty.

If this field contains a 0 (zero), then Field 7 must also contain a 0 (zero). If this field is empty, then Fields 1, 3, 4 and 5 must be empty.

[$E_{MFK.6}$(IV-I)]

Field 3, the incoming Initialization Vector (IV-I) used in the verification of the MAC encrypted under variant 6 of the MFK.

If this command contains the first block of multiple blocks of data, then this field must be empty.

If this command contains data subsequent to the first block in a multiple block series (that is, it contains continuation data), then this field should contain the ending Initialization Vector from the previously sent data block. This field must be empty if any of the fields 1, 2, 4 and 5 are empty.

This field contains a 16 byte hexadecimal value, or is empty.

[Incoming Variant]

Field 4, the variant applied to the MFK when encrypting the KMAC-I key. This field is optional; if used, it can be one or two bytes long and can contain the numbers 3 or 19. The default variant 3 is used if this field is empty.

This field must be empty if any of the Fields 1, 2 3, and 5 are empty.

[MAC-I]

Field 5, the incoming MAC to be verified. This field must empty if more data is coming in subsequent commands.

A 32 bit MAC is expressed as eight hexadecimal digits and written as two groups of four digits, separated by a space.

A 48 bit or 64 bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

Field 5 should be empty, or its length should be 9 bytes (8 characters plus 1 space), 14 bytes (12 characters plus 2 spaces), or 19 bytes (16 characters plus three spaces).

If Field 2 contains a zero, this field must be empty.

If this field is empty, then the Fields 1, 2, 3 and 4 must also be empty.

$[E_{MFK.O}(KMAC\text{-}O)]$

Field 6, the outgoing KMAC Key (KMAC-O) encrypted under variant 3 or 18 of the MFK. This key is used to generate the outgoing MAC. If this field is empty, then Fields 7, 8 and 9 must also be empty.

This field contains a 16 byte hexadecimal value, or a volatile table location.

[MAC-O Length]

Field 7, the length of the outgoing MAC. The following table indicates the possible outgoing MAC lengths and the code to enter in this field for each one.

If Field 2 contains 0, this field also must be 0.

If Field 7 is empty, then the fields 6, 8 and 9 must be empty.

Returned MAC Size	Code
More data expected; no MAC returned	0
32 bits	1
48 bits	2
64 bits	3

A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A-F) and written as two groups of four digits, separated by a space. A 48 bit or 64 bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

$[E_{MFK.6}(IV\text{-}O)]$

Field 8, the outgoing Initialization Vector (IV-O) encrypted under variant 6 of the MFK. This IV-O is used in the generation of the outgoing MAC

If this command contains the first block of multiple blocks of data, then this field must be empty.

If this command contains data subsequent to the first block in a multiple block series (that is, it contains continuation data), then this field contains the ending Initialization Vector from the previously sent data block.

This field contains a 16 byte hexadecimal value. If the length of the Field 3 is 16, this field also must be of length 16. If the Field 3 is empty, this field also must be empty.

This field must be empty if fields 6, 7 or 8 are empty.

`[Outgoing Variant]`

Field 9, the variant used to encrypt the outgoing KMAC Key (KMAC-O).

This field is optional; if used, it can be one or two bytes long and can contain the numbers 3 or 18. If this field is empty, the default variant 3 is used.

This field must be if either field 6, 7, or 8 are empty.

`Data Type`

Field 10, the data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

`Data Length`

Field 11, the data length. This command authenticates up to 4096 bytes of data.

If more data is being sent in the next command, then the data length must be multiples of eight for binary data, or multiples of 16 for Unpacked ASCII data (batch authentication is indicated when Field 2 is set to 0. See Authentication in Batches for additional information).

If data sent is not in batches, the Network Security Processor will pad the data field with binary zeros to a multiple of eight.

This field contains a 1 to 4 byte decimal value.

`Data`

Field 12, the input data. This field can be from 1 to 4096 bytes long and in binary or unpacked ASCII hexadecimal format.

If the data is in unpacked ASCII hexadecimal format, then this field can contain the numbers 0 through 9 and characters A to F.

Table 6-2. Command 58: MAC Translate (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	58
1	[$E_{MFK.I}$(KMAC-I)]*	0, 16	0 - 9, A - F
2	[MAC-I Length]	0, 1	0 - 3
3	[$E_{MFK.6}$(IV-I)]	0, 16	0 - 9, A - F
4	[Incoming Variant]	0 - 2	3, 19
5	[MAC-I]	0, 9, 14, 19	0 - 9, A - F, “ ”
6	[$E_{MFK.O}$(KMAC-O)]*	0, 16	0 - 9, A - F
7	[MAC-O Length]	0, 1	0 - 3

Table 6-2. Command 58: MAC Translate (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
8	[$E_{MFK.6}$(IV-O)]	0, 16	0 - 9, A - F
9	[Outgoing Variant]	0 - 2	3, 18
10	Data Type	1	U, B
11	Data Length	1 - 4	0 - 9
12	Data	1 - 4096	0 - 9, A - F if unpacked ASCII

*Can be a volatile table location.

Responding Parameters

`68`

Field 0, the response identifier.

`MAC Length-I`

Field 1, the length of the incoming MAC.

If this field is set to 0, then more data is expected and Field 2 of the response will contain the ending Initialization Vector.

If this field is set to 1, 2, or 3, then Field 2 will contain the MAC verification flag.

`Verification Flag or` $E_{MFK.6}$`(Ending IV-I)`

Field 2, either verification flag, or if field 1 of the response is 0, the ending Initialization Vector encrypted under variant 6 of the MFK.

If your use of this command results in the generation of an ending Initialization Vector in this field, use it as the starting Initialization Vector in the subsequent MAC command to continue generating MACs.

If your use of this command results in a MAC verification flag, then this field will return Y if the MAC is verified or N if the MAC is not verified.

This field will be empty if the Fields 1, 2, 3, 4 and 5 in the command are empty (that is, this command will only generate a MAC). This field is either empty, or a one byte value Y or N, or 16 byte hexadecimal value.

`KMAC-I Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the incoming MAC key. If option 88 is enabled, this field will contain the first six digits of the result.

This field will be empty if the Fields 1, 2, 3, 4 and 5 in the command are empty.

`MAC Length-O`

Field 4, the length of the generated outgoing MAC.

If this field is set to 0, then more data is expected and Field 2 will contain the ending Initialization Vector.

If this field is set to 1, 2, or 3, then Field 2 will contain the MAC.

`MAC or` $E_{MFK.6}$`(Ending IV-O)`

Field 5, either verification flag, or if field 1 of the response is 0, the Initialization Vector used in the MAC generation process.

If your use of this command results in the generation of an ending Initialization Vector in this field, use it as the starting Initialization Vector in the subsequent MAC command to continue the generation of the MAC. Otherwise, this field will have generated MAC if the incoming MAC is verified, or it will be empty.

This field is contains either a zero byte, 9 byte, 14 byte, 16 byte or 19 byte hexadecimal value, as well as spaces.

If the verification flag (Field 2) is N, then this field is empty. This field will be empty if Fields 6, 7, 8 and 9 in the command are empty.

`KMAC-O Check Digits`

Field 6, check digits; the first four digits that result from encrypting zeros using the outgoing MAC Key. If option 88 is enabled, this field will contain the first six digits of the result.

If the verification flag (Field 2) is N, then this field is empty.

This field will be empty if the Fields 6, 7, 8 and 9 in the command are empty.

Table 6-3. Response 68: MAC Translate

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	68
1	MAC Length-I	0, 1	0 - 3
2	Verification flag or $E_{MFK.6}$(Ending IV-I)	0, 1, 16	0 - 9, A - F, Y, N
3	KMAC-I Check Digits	0, 4 or 6	0 - 9, A - F
4	MAC Length-O	0, 1	0 - 3
5	MAC or $E_{MFK.6}$(Ending IV-O)	0, 9, 14, 16, 19	0 - 9, A - F, “ ”
6	KMAC-O Check Digits	0, 4 or 6	0 - 9, A - F

Usage Notes

- If Fields 1, 2, 3, 4 and 5 in this command are empty, then this command will only generate a MAC.
- If Fields 6, 7, 8 and 9 are empty, then this command will only verify a MAC.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Generating a MAC using Variant 18.

- Clear-text outgoing KMAC key (KMAC-O): FEDC BA98 7654 3210.
  The outgoing KMAC key (KMAC-O) encrypted under variant 18 of the MFK: B227 EE34 8FDE 4FD7.
- MAC length: 64 bits (3)
- Clear-text outgoing Initialization Vector (IV-O): 2558 8552 2558 8552.
  The outgoing Initialization Vector (IV-O) encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Data type: Unpacked ASCII hexadecimal (U)
- Data length: 8 bytes
- Data: 01AB8D89

The command looks like this:

```
<58######B227EE348FDE4FD7#3#2D03E0CE90E4CA46#18#U#8#
01AB8D89#>
```

The Network Security Processor issues the following response.

```
<68####3#0299 23CE A64A D1B0#A68C#>
```

Verifying a MAC with data in Unpacked ASCII format.

- Clear-text incoming MAC Key (KMAC-I): FEDC BA98 7654 3210.
  The incoming MAC Key (KMAC-I) encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC length: 64 bits (3)
- Clear-text outgoing Initialization Vector (IV-O): 2558 8552 2558 8552.
  The outgoing Initialization Vector (IV-O) encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Variant: 3
- MAC: 78FA FA86 68CF 1FC7
- Data type: Unpacked ASCII hexadecimal (U)

- Data length: 6 bytes
- Data: 303430

The command looks like this:

```
<58#1B866280C012DD33#3#2D03E0CE90E4CA46#3#
78FA FA86 68CF 1FC7#####U#6#303430#>
```

The Network Security Processor issues the following response.

```
<68#3#Y#A68C####>
```

Translating a MAC using different incoming and outgoing IVs.

- Clear-text incoming MAC Key (KMAC-I): FEDC BA98 7654 3210.
  The incoming MAC Key (KMAC-I) encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC length: 32 bits (1).
- Clear-text incoming Initialization Vector (IV-I): 2558 8552 2558 8552.
  The incoming Initialization Vector (IV-I) encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- Incoming Variant: 3
- Incoming MAC: 78FA FA86
- Clear-text outgoing MAC Key (KMAC-O): FEDC BA98 7654 3210.
  The outgoing MAC Key (KMAC-O) encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- Response length: 32 bits (1)
- Clear-text outgoing Initialization Vector (IV-O): 1111 2222 3333 4444.
  The outgoing Initialization Vector (IV-O) encrypted under variant 6 of the MFK: 790D FFBC B1B0 E882.
- Outgoing Variant: 3
- Data type: Binary (B)
- Data length: 3 bytes
- Data: 040

The command looks like this:

```
<58#1B866280C012DD33#1#2D03E0CE90E4CA46#3#78FA FA86#
1B866280C012DD33#1#790DFFBCB1B0E882#3#B#3#040#>
```

The Network Security Processor issues the following response.

```
<68#1#Y#A68C#1#F80F C16A#A68C#>
```

Generate MAC and Encrypt or Translate Data (Command 59)

Command 59 generates a MAC and encrypts or translates data. This command supports ECB and CBC modes of DES.

This command supports only 1key-3DES (single-length) keys. It has high security exposure and is not enabled in the Network Security Processor's default security policy.

Multiple Mode Command

Command – ECB-Mode Encryption

<59#0#$E_{MFK.3}$(KMAC)#MAC Data##$E_{MFK.2}$(KD_O)#Clear Data#>

Command – CBC-Mode Encryption

<59#0#$E_{MFK.3}$(KMAC)#MAC Data##$E_{MFK.2}$(KD_O)###[$E_{MFK.6}$(IV-O)]#
Data Type#Length#Clear Data#>

Command – ECB-Mode Translation

<59#0#$E_{MFK.3}$(KMAC)#MAC Data#$E_{MFK.2}$(KD_I)#$E_{MFK.2}$(KD_O)#
Encrypted Data#>

Command – CBC-Mode Translation

<59#0#$E_{MFK.3}$(KMAC)#MAC Data#$E_{MFK.2}$(KD_I)#$E_{MFK.2}$(KD_O)##
[$E_{MFK.6}$(IV-I)]#[$E_{MFK.6}$(IV-O)]#Data Type#Length#Encrypted Data#>

Multiple Mode Response

Response – ECB-Mode

<69#MAC#KMAC Check Digits#Encrypted Data#
[Incoming KD Check Digits]#Outgoing KD Check Digits#>[CRLF]

Response – CBC-Mode

<69#MAC#KMAC Check Digits##[Incoming KD Check Digits]#
Outgoing KD Check Digits#[$E_{MFK.6}$(IV-I)]#$E_{MFK.6}$(IV-O)#
Data Type#Length#Encrypted Data#>[CRLF]

Calling Parameters – ECB-Mode Encryption

59

Field 0, the command identifier.

0

Field 1, the data continuation flag, must be set to 0.

$E_{MFK.3}$(KMAC)

Field 2, the KMAC key encrypted under variant 3 of the MFK. This key is used to generate the MAC. This field contains either a 16 byte hexadecimal value or a volatile table location.

MAC Data

Field 3, data to be authenticated. This field can be from one to 240 bytes long and can contain the characters A to Z, the numbers 0 through 9, and “,”, “.”, and “ ”.

Reserved

Field 4, this field is empty.

$E_{MFK.2}$(KD_O)

Field 5, the outgoing Data Key encrypted under variant 2 of the MFK. This key is used to encrypt the data contained in Field 6.

This field contains a 16 byte hexadecimal value or a volatile table location.

Clear Data

Field 6, the clear data to be encrypted using the outgoing Data Key and employing the Electronic Code Book (ECB) method of DES.

Data less than 16 characters must be right padded with zeros such that the data length is 16 hexadecimal characters.

Table 6-4. Command 59: ECB-Mode Encryption

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	59
1	Data continuation flag	1	0
2	$E_{MFK.3}$(KMAC)*	16	0 - 9, A - F
3	MAC Data	1 - 240	0 - 9, A - Z, , . “ ”
4	Reserved	0	
5	$E_{MFK.2}$(KD_O)*	16	0 - 9, A - F
6	Clear Data	16	0 - 9, A - F

*Can be a volatile table location.

Calling Parameters – CBC-Mode Encryption

`59`

Field 0, the command identifier.

`0`

Field 1, the data continuation flag, must be set to zero.

$E_{MFK.3}$(KMAC)

Field 2, the KMAC key encrypted under variant 3 of the MFK. This key is used to generate the MAC. This field contains either a 16 byte hexadecimal value or a volatile table location.

`MAC Data`

Field 3, data to be authenticated. This field can be from one to 240 bytes long and can contain the characters A to Z, the numbers 0 through 9, and “,”, “.”, and “ ”.

`Reserved`

Field 4, a reserved field, it must be empty.

$E_{MFK.2}$(KD_O)

Field 5, the outgoing Data Key encrypted under variant 2 of the MFK. This key is used to encrypt the data contained in Field 11. This field contains either a 16 byte hexadecimal value, or a volatile table location.

`Reserved`

Field 6, a reserved field, it must be empty.

`Reserved`

Field 7, a reserved field, it must be empty.

`[`$E_{MFK.6}$`(IV-O)]`

Field 8, the Initialization Vector encrypted under variant 6 of the MFK. This IV is used in the outgoing CBC data encryption process. If this field is empty, the default Initialization Vector of all zeros will be used. This field contains a 16 byte hexadecimal value or is empty.

`Data Type`

Field 9, the type of the data in Field 11: Unpacked ASCII hexadecimal or binary. This field contains one byte, either U for unpacked ASCII hexadecimal, or B for binary.

`Length`

Field 10, the length of the data in Field 11. This field contains a 1 to 4 byte decimal value.

`Clear Data`

Field 11, the clear data to be encrypted using the outgoing data encryption key and employing the Cipher Block Chaining (CBC) method of encryption.

This field is from one to 3500 bytes long if the data is in binary format; if the data is in unpacked ASCII hexadecimal format, then this field is two to 3500 bytes long and must be a multiple of 2.

The Network Security Processor will pad the data with binary zeros to achieve a value that is an eight byte multiple (binary data) or a 16 byte multiple (unpacked ASCII data).

Table 6-5. Command 59: CBC-Mode Encryption (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	59
1	Data continuation flag	1	0
2	$E_{MFK.3}$(KMAC)*	16	0 - 9, A - F
3	MAC Data	1 - 240	0 - 9, A - Z, , . “ ”
4	Reserved	0	
5	$E_{MFK.2}$(KD_O)*	16	0 - 9, A - F
6	Reserved	0	
7	Reserved	0	
8	$E_{MFK.6}$(IV-O)	0, 16	0 - 9, A - F
9	Data type	1	U, B

Table 6-5. Command 59: CBC-Mode Encryption (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
10	Length	1 - 4	0 - 9
11	Clear Data	1 - 3500 or 2 - 3500	0 - 0, A - F (if unpacked ASCII hexadecimal

*Can be a volatile table location.

Calling Parameters – ECB-Mode Translation

`59`

Field 0, the command identifier.

`0`

Field 1, the data continuation flag, must be set to 0.

$E_{MFK.3}(KMAC)$

Field 2, the KMAC key encrypted under variant 3 of the MFK. This key is used to generate the MAC. This field contains either a 16 byte hexadecimal value or a volatile table location.

`MAC Data`

Field 3, data to be authenticated. This field can be from one to 240 bytes long and can contain the characters A to Z, the numbers 0 through 9, and “,”, “.”, and “ ”.

$E_{MFK.2}(KD_I)$

Field 4, the incoming Data Key encrypted under variant 2 of the MFK. This key is used to decrypt the data in Field 6. This field contains a 16 byte hexadecimal value or a volatile table location.

$E_{MFK.2}(KD_O)$

Field 5, the outgoing Data Key encrypted under variant 2 of the MFK. This key is used to encrypt the data being translated. This field contains a 16 byte hexadecimal value or a volatile table location.

`Encrypted Data`

Field 6, the data to be translated. This field contains a 16 byte hexadecimal value that can contain the numbers 0 through 9 and the characters A through F.

Table 6-6. Command 59: ECB-Mode Translation

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	59
1	Data continuation flag	1	0
2	$E_{MFK.3}(KMAC)$*	16	0 - 9, A - F
3	MAC Data	1 - 240	0 - 9, A - Z, , . “ ”
4	$E_{MFK.2}(KD_I)$*	16	0 - 9, A - F
5	$E_{MFK.2}(KD_O)$*	16	0 - 9, A - F
6	Encrypted Data	16	0 - 9, A - F

*Can be a volatile table location.

Calling Parameters – CBC-Mode Translation

`59`

Field 0, the command identifier.

`0`

Field 1, the data continuation flag, must be set to 0.

$E_{MFK.3}(KMAC)$

Field 2, the KMAC Key encrypted under variant 3 of the MFK. This key is used to generate the MAC. This field contains a 16 byte hexadecimal value or a volatile table location.

`MAC Data`

Field 3, data to be authenticated. This field can be from one to 240 bytes long and can contain the characters A to Z, the numbers 0 through 9, and “,”, “.”, and “ ”.

$E_{MFK.2}(KD_I)$

Field 4, the incoming Data Key encrypted under variant 2 of the MFK. This key will be used to decrypt the data in Field 6. This field contains a 16 byte hexadecimal value or a volatile table location.

$E_{MFK.2}(KD_O)$

Field 5, the outgoing Data Key encrypted under variant 2 of the MFK. This key is used to re-encrypt the data being translated. This field contains a 16 byte hexadecimal value or a volatile table location.

`Reserved`

Field 6, reserved field, it must be empty.

$[E_{MFK.6}(IV\text{-}I)]$

Field 7, the incoming Initialization Vector (IV-I) encrypted under variant 6 of the MFK. This IV is used during decryption. If this field is empty, the default Initialization Vector of all zeros is used. This field contains a 16 byte hexadecimal value, or is empty.

$[E_{MFK.6}(IV\text{-}O)]$

Field 8, the outgoing Initialization Vector (IVO) encrypted under variant 6 of the MFK. This IV is used during re-encryption. If this field is empty, then the default Initialization Vector of all zeros is used. This field contains a 16 byte hexadecimal value or is empty.

Data Type

Field 9, the data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

Length

Field 10, the length of the data in Field 11.

The data is from eight to 3496 bytes long if the data is in binary format. If the data is in unpacked ASCII hexadecimal format, then the length is from 16 to 3488 bytes long. This field contains a 1 to 4 byte decimal value.

Data

Field 11, the data to be translated, that is decrypted with the incoming data encryption key, and re-encrypted using the outgoing data encryption key.

This field is from eight to 3496 bytes long if the data is in binary format. If the data is in unpacked ASCII hexadecimal format, then this field is from 16 to 3488 bytes long.

When using this command, be sure to pad the data with zeros, if necessary, to achieve a value that is an eight byte multiple (binary) or a 16 byte multiple (ASCII).

Table 6-7. Command 59: CBC-Mode Translation (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	59
1	Data continuation flag	1	0
2	$E_{MFK.3}$(KMAC*	16	0 - 9, A - F
3	MAC Data	1 - 240	0 - 9, A - Z, , . “ ”
4	$E_{MFK.2}(KD_I)$*	16	0 - 9, A - F

Table 6-7. Command 59: CBC-Mode Translation (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
5	$E_{MFK.2}(KD_O)$*	16	0 - 9, A - F
6	Null	0	
7	$E_{MFK.6}$(IV-I)	0, 16	0 - 9, A - F
8	$E_{MFK.6}$(IV-O)	0, 16	0 - 9, A - F
9	Data type	1	U, B
10	Length	1 - 4	0 - 9
11	Encrypted Data	8 - 3496 or 16 - 3488	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters – ECB-Mode

`69`

Field 0, the response identifier.

`MAC`

Field 1, the 32 bit, generated MAC. This field contains an 8 byte hexadecimal value.

`KMAC Check Digits`

Field 2, the MAC Key check digits; the first four digits that result from encrypting zeros using the MAC Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Encrypted Data`

Field 3, the encrypted or translated data. This field contains a 16 byte hexadecimal value.

`[Incoming KD Check Digits]`

Field 4, a variable field, depending on the nature of the command sent. If the command sent was translation, then this field contains the incoming data key check digits; the first four digits that result from encrypting zeros using the incoming Data Key. If option 88 is enabled, this field will contain the first six digits of the result. If the command sent was encryption, then this field is empty.

`Outgoing KD Check Digits`

Field 5, the outgoing data encryption key check digits; the first four digits that result from encrypting zeros using the outgoing Data Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 6-8. Response 69: ECB-Mode

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	69
1	MAC	8	0 - 9, A - F
2	KMAC Check Digits	4 or 6	0 - 9, A - F
3	Encrypted data	16	0 - 9, A - F
4	Incoming KD Check Digits	0, 4 or 6	0 - 9, A - F
5	Outgoing KD Check Digits	4 or 6	0 - 9, A - F

Responding Parameters – CBC-Mode

`69`

Field 0, the response identifier.

`MAC`

Field 1, the 32 bit, generated MAC. This field contains an 8 byte hexadecimal value.

`KMAC Check Digits`

Field 2, the MAC key check digits; the first four digits that result from encrypting zeros using the message authentication key. If option 88 is enabled, this field will contain the first six digits of the result.

`Reserved`

Field 3, a reserved field, it will be empty.

`[Incoming KD Check Digits]`

Field 4, a variable field, depending on the nature of the command sent. If the command sent was translation, then this field contains the incoming data encryption key's check digits; the first four digits that result from encrypting zeros using the encryption key.If option 88 is enabled, this field will contain the first six digits of the result. If the command sent was encryption, then this is empty.

`Outgoing KD Check Digits`

Field 5, the outgoing Data Key check digits; the first four digits that result from encrypting zeros using the outgoing Data Key. If option 88 is enabled, this field will contain the first six digits of the result.

$[E_{MFK.6}(IV\text{-}I)]$

Field 6, a variable field, depending on the nature of the command sent.

If the command sent was translation, then this field contains the ending Initialization Vector encrypted under variant 6 of the MFK. If the command sent was encryption, then this is empty.

$E_{MFK.6}$(IV-O)

Field 7, the ending Initialization Vector encrypted under variant 6 of the MFK. This IV results from encrypting the text or re-encrypting the text that is being translated. This field contains a 16 byte hexadecimal value.

Data Type

Field 8, the data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

Length

Field 9, the length of the data in Field 10. For encryption, the length of the data returned in this field may be longer than the data sent. This field contains a to 4 byte decimal value.

Encrypted Data

Field 10, the encrypted or translated text. This field is from 8 to 3504 bytes long if the data is in binary format; if the data is in unpacked ASCII hexadecimal format, then this field is from 16 to 3504 bytes long.

Table 6-9. Response 69: CBC-Mode

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	69
1	MAC	8	0 - 9, A - F
2	KMAC Check Digits	4 or 6	0 - 9, A - F
3	Reserved	0	
4	Incoming KD Check Digits	0, 4 or 6	0 - 9, A - F
5	Outgoing KD Check Digits	4 or 6	0 - 9, A - F
6	$E_{MFK.6}$(IV-I)	0, 16	0 - 9, A - F
7	$E_{MFK.6}$(IV-O)	16	0 - 9, A - F
8	Data type	1	U, B
9	Length	1 - 4	0 - 9
10	Encrypted data	8 - 3504 or 16 - 3504	0 - 9, A - F (if unpacked ASCII hexadecimal)

Usage Notes

- Generate the MAC Key cryptogram.
- Generate the incoming and outgoing data keys and IVs.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Generating a MAC, and Encrypting Data Using ECB-Mode

- Clear-text MAC key (KMAC): 8FF4 98F1 B661 5151.
  The MAC key (KMAC) encrypted under variant 3 of the MFK: D377 30CD D619 FE8A.
- Data to be authenticated:
  A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X.
- Clear-text outgoing Data Key: 3F78 1D6A B654 AEAD.
  The outgoing Data Key encrypted under variant 2 of the MFK: 192E 9678 8DB2 9500.
- Clear data to be encrypted: 1234567890ABCDEF

The command looks like this:

```
<59#0#D37730CDD619FE8A#A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8
S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V
2W3X##192E96788DB29500#1234567890ABCDEF#>
```

The Network Security Processor issues the following response.

```
<69#4316C2C1#1DE3#7CAA1966B0EFFA55##430D#>
```

Encrypting Data Using CBC-Mode using the default IV

- Clear-text MAC key (KMAC): 8FF4 98F1 B661 5151.
  The MAC key (KMAC) encrypted under variant 3 of the MFK: D377 30CD D619 FE8A.
- Data to be authenticated:
  A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X.
- Clear-text outgoing Data Key: 3F78 1D6A B654 AEAD.
  The outgoing Data Key encrypted under variant 2 of the MFK: 192E 9678 8DB2 9500.
- Data type: Unpacked ASCII hexadecimal (U)
- Length: 16 bytes

- Clear data: 1234567890ABCDEF

The command looks like this:

```
<59#0#D37730CDD619FE8A#A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8
S9T0U1V2W3X4Y5Z6A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V
2W3X##192E96788DB29500####U#16#1234567890ABCDEF#>
```

The Network Security Processor issues the following response.

```
<69#4316C2C1#1DE3###430D##4400FBB704908E15#U#16#7CAA1966B0EFF
A55#>
```

Translating Data Using ECB-Mode

The following example illustrates translating data using ECB mode, based on the following input:

- Clear-text MAC key (KMAC): 8FF4 98F1 B661 5151.
  The MAC key (KMAC) encrypted under variant 3 of the MFK: D377 30CD D619 FE8A.
- Data to be authenticated:
  1234567890ABCDEFGHIJ1234567890ABCDEFGHIJ1234567890ABCDEFGH
  IJ1234567890ABCDEFGHIJ1234567890ABCDEFGHIJ1234567890ABCDEFGH
  IJ1234567
- Clear-text incoming Data Key: D9E5 7FE9 8F83 322A.
  The incoming Data Key encrypted under variant 2 of the MFK: A437 C39D DB0A EAB5.
- Clear-text outgoing Data Key: 4029 BFE6 3720 0E98.
  The outgoing Data Key encrypted under variant 2 of the MFK: 3D83 E72F F023 EEBB.
- Encrypted data to be translated: 413E C8B2 0165 E59A

The command looks like this:

```
<59#0#D37730CDD619FE8A#1234567890ABCDEFGHIJ1234567890ABCDEF
GHIJ1234567890ABCDEFGHIJ1234567890ABCDEFGHIJ1234567890ABCDE
FGHIJ1234567890ABCDEFGHIJ1234567#A437C39DDB0AEAB5#
3D83E72FF023EEBB#413EC8B20165E59A#>
```

The Network Security Processor issues the following response.

```
<69#3B3F91F7#1DE3#2190B17248E002CA#90B2#B607#>
```

Verify and Generate MAC for VISA UKPT (Command 5C)

Command 5C verifies a MAC and generates an approval or denial MAC.

This command, by default, will generate a 1key-3DES (single-length) session key. Use option A2 to control the length of the generated session key.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<5C#E_MFK.8(Derivation Key)#Key Serial Number#Data#MAC-1#
[Session Key Length#]>
```

Response

```
<6C#Verification Flag#MACs#>[CRLF]
```

Calling Parameters

`5C`

Field 0, the command identifier.

$E_{MFK.8}$`(Derivation Key)`

Field 1, the 1key-3DES (single-length) or 2key-3DES (double-length) Derivation Key encrypted under variant 8 of the MFK. This field contains a 16 or 32 byte hexadecimal value, or a volatile table location. It can be a 1key-3DES (single-length) key only if option A2 is set to "S".

`Key Serial Number`

Field 2, the current Key Serial Number. This field contains a 10 to 20 byte hexadecimal value, leading Fs will be suppressed.

`Data`

Field 3, the data used to generate the MACs. This field contains a 16 or 32 byte hexadecimal value.

`MAC-1`

Field 4, the MAC to be verified. This field contains an eight byte hexadecimal value.

`[Session Key Length#]`

Field 5, this field is required only if option A2 is set to “B”, for all other cases this field is optional. If it exists, it should contain “S” if a 1key-3DES (single-length) session key is to be generated, or “D” if a 2key-3DES (double-length) session key is to be generated. If option A2 is set to “D”, this field cannot contain the value “S”, and if option A2 is set to “S”, this field cannot contain the value “D”.

Table 6-10. Command 5C: Verify and Generate MAC for VISA UKPT

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	5C
1	$E_{MFK.8}$(Derivation Key)*	16,[(639-s08(f0 9Ke)5.,(t). A0 Fd)]TJ9.4.0040 -1.6393 TD 00	

Responding Parameters

`6C`

Field 0, the response identifier.

`Verification Flag`

Field 1, the MAC verification flag. This field returns Y if the MAC is verified or N if the MAC is not verified.

`MACs`

Field 2, contains the MACs to return to the PIN pad. This field returns MAC-2 and MAC-3 if MAC one is verified; otherwise, it returns 00000000 and MAC three. This field contains a 16 byte hexadecimal value.

Usage Notes

Before using Command 5C, generate the Derivation Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying MAC-1 and generating an approval or denial MAC in return.

- Option A2 is set to “S”.
- Clear-text Derivation Key: 1334 1334 1334 1334.
  The DerivationKey encrypted under variant 8 of the MFK: 4A79 F2A0 E61F EECF.
- Key Serial Number: 9876 5432 10E0 0001.
- Data: 1234 1234 5678 5678.
- MAC #1: 6268 14A7.

The command looks like this:

```
<5C#4A79F2A0E61FEECF#9876543210E00001#1234123456785678#
626814A7#>
```

The Network Security Processor returns the following response:

```
<6C#Y#0C266C371DEABF85#>
```

where MAC #2 = `0C266C37`, and MAC #3 = `1DEABF85`.

This example shows the syntax when option A2 is set to “B”.

```
<5C#4A79F2A0E61FEECF#9876543210E00001#1234123456785678#626814
A7#S#>
```

2key-3DES (double-length) session key is used to verify MAC-1 and generate an approval or denial MAC in return.

- Option A2 is set to “D”.
- Clear-text Base Derivation Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The Base Derivation Key encrypted under variant 8 of the MFK:
  AAA57E4E99AE9B0328F6BA950E1664FA
- Key Serial Number: 9876 5432 10E0 0001.
- Data: 1234 1234 5678 5678.
- MAC #1: 7E37 D982.

The command looks like this:

```
<5C#AAA57E4E99AE9B0328F6BA950E1664FA#9876543210E00001#1234123
456785678#7E37D982#>
```

The Network Security Processor returns the following response:

```
<6C#Y#4D3AA91B0A0E7E12#>
```

where MAC #2 = 4D3AA91B, and MAC #3 = 0A0E7E12

This example shows the syntax when option A2 is set to “B”.

```
<5C#AAA57E4E99AE9B0328F6BA950E1664FA#9876543210E00001#1234123
456785678#7E37D982#D#>
```

The Network Security Processor returns the following response:

```
<6C#Y#4D3AA91B0A0E7E12#>
```

where MAC #2 = 4D3AA91B, and MAC #3 = 0A0E7E12.

Verify MAC and Decrypt PIN (Command 5F)

Command 5F verifies a MAC and decrypts the outer layer of an encrypted PIN Block.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy. You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

This command supports only 1key-3DES (single-length) working keys.

Command

<5F#$E_{MFK.3}$(KD)#MAC Length#[$E_{MFK.6}$(IV)]#MAC#E_{KD}(PIN Block)#
Data Type#Data Length#Data#>

Response

<6F#Verification Flag or $E_{MFK.6}$(Ending IV)#
Decrypted PIN Block#KD Check Digits#

Calling Parameters

5F

Field 0, the command identifier.

$E_{MFK.3}$(KD)

Field 1, the Data Key used for two purposes; first to verify the MAC, and if successful, to decrypt the encrypted PIN block. This field contains a 16 byte hexadecimal value.

MAC Length

Field 2, the size of the MAC to be verified. If Field 2 is set to 0, then Field 4 and 5 must be empty.

This table indicates the possible MAC sizes and the code to enter in this field for each one.

MAC Size	Numerical Code
More data expected; no MAC verified	0
32 bits	1

MAC Size	Numerical Code
48 bits	2
64 bits	3

A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A - F) and written as two groups of four digits, separated by a space. A 48 bit or 64 bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

$[E_{MFK.6}(IV)]$

Field 3, the Initialization Vector encrypted under variant 6 of the MFK. This IV is used in the verification of a MAC.

If this command contains the first block of multiple blocks of data then this field must be empty. If this command contains data subsequent to the first block in a multi-block series (that is, it contains continuation data), then this field contains the ending Initialization Vector from the previously sent data block. This field contains a16 byte hexadecimal value, or is empty.

MAC

Field 4, the MAC to be verified. This field will contain the MAC when there is no more data in a subsequent command. A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A - F) and written as two groups of four digits, separated by a space. A 48 bit or 64 bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space. This field will be empty if field 2 contains a 0.

E_{KD}(PIN Block)

Field 5, the incoming PIN Block encrypted under the Data Key (KD). This field contains a 16 byte hexadecimal value or is empty.

Data Type

Field 6, the data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

Data Length

Field 7, the data length.

This command will authenticate up to 4096 bytes of data.

If more data is being sent in the next command – indicated by the Field 2 being set to 0 – then the data length must be multiple of eight. See Authentication in Batches for additional information.

If data sent is not in batches, the Network Security Processor will right pad the data with zeros such that its length will be a multiple of eight.

This field contains a 1 to 4 byte decimal value.

`Data`

Field 8, the data to be authenticated. This field can be from 1 to 4096 bytes long. If the data is in unpacked ASCII hexadecimal format, then this field can contain the numbers 0 through 9 and the characters A through F.

Table 6-12. Command 5F: Verify MAC and Decrypt PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	5F
1	$E_{MFK.3}$(KD)*	16	0 - 9, A - F
2	MAC Length	1	0 - 3
3	[$E_{MFK.6}$(IV)]	0, 16	0 - 9, A - F
4	MAC	0, 9, 14, 19	0 - 9, A - F, “ ”
5	E_{KD}(PIN Block)	0, 16	0 - 9, A - F
6	Data Type	1	U, B
7	Data Length	1 - 4	0 - 9
8	Data to be authenticated	1 - 4096	0 - 9, A - F if unpacked ASCII

*Can be a volatile table location.

Responding Parameters

`6F`

Field 0, the response identifier.

`Verification Flag or` $E_{MFK.6}$`(Ending IV)`

Field 1, the ending Initialization Vector if command Field 2 is set to 0, or the MAC verification flag if command Field 2 is not set to 0.

If your use of this command results in the generation of an ending Initialization Vector in this field, use it as the starting Initialization Vector in subsequent MAC command to continue generating MACs.

If your use of this command results in a MAC verification flag, then this field will return Y if the MAC is verified, or N if the MAC is not verified.

This field contains either a 16 byte hexadecimal value, or a 1 byte value Y or N.

`Decrypted PIN Block`

Field 2, the decrypted PIN block. Field 2 is empty if MAC is not verified or command Field 2 is set to 0.

`KD Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the Data Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 6-13. Response 6F: Verify MAC and Decrypt PIN

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	6F
1	Verification Flag or $E_{MFK.6}$(Ending-IV)	1, 16	Y, N, 0 - 9, A - F
2	Decrypted PIN Block	0, 16	0 - 9, A - F
3	KD Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 5F, generate the communications key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a MAC and decrypting the outer layer of a PIN Block.

- Clear-text Data Key: FEDC BA98 7654 3210.
  The Data Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC length: 48 bits (2)
- Clear-text Initialization Vector: 2558 8552 2558 8552.
  The Initialization Vector encrypted under variant 6 of the MFK: 2D03 E0CE 90E4 CA46.
- MAC (48 bits): 78FA FA86 68CF
- Clear-text PIN block: 1234 0000 0000 0000.
  The PIN block encrypted under the Data Key: A931 0B88 55BC 6881.
- Data type: Binary (B)
- Data length: 3 bytes
- Data to be authenticated: 040

The command looks like this:

```
<5F#1B866280C012DD33#2#2D03E0CE90E4CA46#78FA FA86 68CF#
A9310B8855BC6881#B#3#040#>
```

The Network Security Processor issues the following response.

```
<6F#Y#1234000000000000#A68C#>
```

Generate MAC (Command 98)

Command 98 generates a MAC using Cipher Block Chaining per ANSI X9.9. This command supports 1key-3DES (single-length) or 2key-3DES (double-length) working keys.

Three types of MACs can be generated.

- Single DES CBC - uses the 1key-3DES (single-length) MAC key for all blocks of data.
- ISO 9797-1 Algorithm 1 - uses both the left and right half of the 2key-3DES (double-length) MAC key for all blocks of data.
- ISO 9797-1 Algorithm 3 - uses the left half of the 2key-3DES (double-length) MAC key for all data blocks except the last block. The last data block is processed in a true 3DES operation using both the left and right half of the MAC key.

If the MAC Type is either ISO 9797-1 Algorithm 1, or ISO 9797-1 Algorithm 3, and option 6A is enabled, this command will support a replicated 1key-3DES (single-length) key. If option 6A is disabled, which is the default, and the MAC Type is either ISO 9797-1 Algorithm 1 or ISO 9797-1 Algorithm 3, this command requires a 2key-3DES (double-length) key. All other MAC Types support only 1key-3DES (single-length) keys.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<98#$E_{MFK.V}$(KMAC)#MAC Length#MAC Type#[$E_{MFK.6}$(IV)]#Data Type#
Length#Data#[Variant#]>

Response

<A8#MAC Length#MAC or $E_{MFK.6}$(Ending IV)#KMAC Check Digits#>
[CRLF]

Calling Parameters

98

Field 0, the command identifier.

$E_{MFK.V}$(KMAC)

Field 1, the MAC Key encrypted under variant 3 or 18 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location. If field 3 contains

either a 6 or 7 then this field must contain a 32 byte hexadecimal value or a volatile table location.

`MAC Length`

Field 2, the length of the MAC to be returned. The following table indicates the possible returned MAC lengths and the code to enter in this field for each one.

Returned MAC Size	Code
More data expected; no MAC returned	0
32 bits	1
48 bits	2
64 bits	3

A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A - F) and written as two groups of four digits, separated by a space. A 48- or 64-bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

`MAC Type`

Field 3, the algorithm used to generate the MAC. The following table indicates the supported MAC types and the numerical value to enter in this field for each MAC type.

MAC Type	Value
Cipher block chaining (CBC) (single-length DES)	Empty, or 1-5
ISO - 9797-1 Algorithm 1	6
ISO - 9797-1 Algorithm 3	7

$[E_{MFK.6}(IV)]$

Field 4, the Initialization Vector encrypted under variant 6 of the MFK.

If this command contains the first block of multiple blocks of data, or if you are authenticating only one block of data, then this field must be empty; the Network Security Processor will use its default Initialization Vector of all zeros.

If this command contains data subsequent to the first block in a multi-block series (that is, it contains continuation data), then this field should contain the ending Initialization Vector from the previously sent data block. This field is either empty, or contains a 16 byte hexadecimal value.

`Data Type`

Field 5, the data types are:

Data Type	Value
Unpacked ASCII hexadecimal	U
Binary	B

See Data formats on page 1-4 for more information.

`Length`

Field 6, the data's length. This command will authenticate up to 4096 bytes of data. If more data is being sent in the next command – indicated by Field 2 being set to 0 – then the data length must be a multiple of eight for binary data, and a multiple of 16 for Unpacked ASCII data. If no more data is being sent, the Network Security Processor will right- pad the data with binary zeros (nulls, 0x00) such that the resulting data length will be a multiple of eight. This field contains a 1 to 4 byte decimal value.

`Data`

Field 7, the input data. This field can be from one to 4096 bytes long and in binary or unpacked ASCII hexadecimal format. If the data is in unpacked ASCII hexadecimal format, then this field can contain the numbers 0 through 9 and the characters A through F.

`[Variant#]`

Field 8, the variant used to encrypt the MAC Key. This field is optional; if used, it can be one or two bytes long and can contain the numbers 3 or 18. If not used, the default variant, 3, is used.

Table 6-14. Command 98: Generate MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	98
1	$E_{MFK.V}$(KMAC)*	16, 32	0 - 9, A - F
2	Response length	1	0 - 3
3	MAC Type	0, 1	Empty, or 1-7
4	$E_{MFK.6}$(IV)	0, 16	0 - 9, A - F
5	Data type	1	U, B
6	Length	1 - 4	0 - 9
7	Data	1 - 4096	0 -9, A - F (if unpacked ASCII)
8	[Variant]	0, 1, 2	3, 18

*Can be a volatile table location.

Responding Parameters

`A8`

Field 0, the response identifier.

`MAC Length`

Field 1, the length of the MAC. This field will contain the value specified in field 2 of the command.

`MAC or` $E_{MFK.6}$`(Ending IV)`

Field 2, if field 1 is set to zero this field will contain the ending Initialization Vector encrypted under variant 6 of the MFK. If Field 1 is not set to 0, this field will contain the MAC.

If your use of this command results in the generation of an ending Initialization Vector in this field, use it as the starting Initialization Vector in the subsequent MAC command to continue generating MACs. This field contains a 9, 14, 16, or 19 byte hexadecimal value as well as spaces (that is, “ ”).

`KMAC Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the MAC Key. If option 88 is enabled, this field will contain the first six digits of the result.

`[MAC Type#]`

Field 4, this field is only present if the MAC type is either 6 or 7.

Table 6-15. Response A8: Generate MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A8
1	MAC length	1	0 - 3
2	MAC or $E_{MFK.6}$(Ending IV)	9, 14, 16, 19	0 - 9, A - F, “ ”
3	KMAC Check Digits	4 or 6	0 - 9, A - F
4*	MAC Type	1	6 or 7
*Only present if MAC Type is 6 or 7			

Usage Notes

Before using Command 98 generate the MAC Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Generating a MAC using a single-length key, the Default IV, and Variant

- Clear-text MAC Key: FEDC BA98 7654 3210.
  The MAC key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.

- MAC length: 32 bits (1).
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 6 bytes.
- Data: 303430.

The command looks like this:

```
<98#1B866280C012DD33#1###U#6#303430#>
```

The Network Security Processor returns the following response:

```
<A8#1#60F0 EFDE#A68C#>
```

Authenticating binary data.

- Clear-text MAC Key: FEDC BA98 7654 3210.
  The MAC key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC length: 32 bits (1).
- Data type: Binary (B).
- Data length: 3 bytes.
- Data: 040.

The command looks like this:

```
<98#1B866280C012DD33#1###B#3#040#>
```

The Network Security Processor returns the following response:

```
<A8#1#60F0 EFDE#A68C#>
```

Generating a ISO - 9797-1 Algorithm 1 MAC using the Default IV and Variant

- Clear-text MAC Key: FEDC BA98 7654 3210 0123 4567 89AB CDEF.
  The MAC Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33 2516 6617 EC74 3AB1.
- MAC length: 32 bits (1).
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 6 bytes.
- Data: 303430.

The command looks like this:

```
<98#1B866280C012DD3325166617EC743AB1#1#6##U#6#303430#>
```

The Network Security Processor returns the following response:

```
<A8#1#AFA3 9CEF#7B83#6#>
```

Generating a ISO - 9797-1 Algorithm 3 MAC using the Default IV and Variant

- Clear-text MAC Key: FEDC BA98 7654 3210 0123 4567 89AB CDEF.
  The MAC Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33 2516 6617 EC74 3AB1.
- MAC length: 32 bits (1).
- Data type: Unpacked ASCII hexadecimal (U).
- Data length: 18 bytes.
- Data: 303430303430303430.

The command looks like this:

```
<98#1B866280C012DD3325166617EC743AB1#1#7##U#18#
303430303430303430#>
```

The Network Security Processor returns the following response:

```
<A8#1#B21A E4A4#7B83#7#>
```

Verify MAC (Command 99)

Command 99 verifies a MAC using Cipher Block Chaining per ANSI X9.9.

Three types of MACs can be verified.

- Single DES CBC - uses the 1key-3DES (single-length) MAC key for all blocks of data.
- ISO - 9797-1 Algorithm 1 - uses both the left and right half of the 2key-3DES (double-length) MAC key for all blocks of data.
- ISO - 9797-1 Algorithm 3 - uses the left half of the 2key-3DES (double-length) MAC key for all data blocks except the last block. The last data block is processed in a 3DES operation using both the left and right half of the MAC key.

If the MAC Type is either ISO - 9797-1 Algorithm 1, or ISO - 9797-1 Algorithm 3, and option 6A is enabled, this command will support a replicated single-length key. If option 6A is disabled, which is the default, and the MAC Type is either ISO - 9797-1 Algorithm 1 or ISO - 9797-1 Algorithm 3, this command requires a 2key-3DES (double-length) key. All other MAC Types support only 1key-3DES (single-length) keys.

This command is enabled in the Network Security Processor's default security policy.

Command

<99#$E_{MFK.V}$(KMAC)#MAC Type#[$E_{MFK.6}$(IV)]#MAC Length#Data Type#
Data Length#Data#[MAC]#[Variant#]>

Response

<A9#MAC Length#Verification Flag or $E_{MFK.6}$(Ending IV)#
KMAC Check Digits#>[CRLF]

Calling Parameters

99

Field 0, the command identifier.

$E_{MFK.V}$(KMAC)

Field 1, the MAC Key encrypted under variant 3 or 19 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location. If field 3 contains either a 6 or 7, then this field must contain a 32 byte hexadecimal value, or a volatile table location.

`[MAC Type]`

Field 2, the algorithm used to verify the MAC. The following table indicates the supported MAC types and the value to enter in this field for each MAC type.

MAC Type	Value
Cipher block chaining (CBC) (single-length DES)	Empty, or 1-5
ISO - 9797-1 Algorithm 1	6
ISO - 9797-1 Algorithm 3	7

`[`$E_{MFK.6}$`(IV)]`

Field 3, the Initialization Vector encrypted under variant 6 of the MFK.

If this command contains the first block of multiple blocks of data, or if you are authenticating only one block of data, then this field must be empty; the Network Security Processor will use its default Initialization Vector of all zeros.

If this command contains data subsequent to the first block in a multi-block series (that is, it contains continuation data), then this field should contain the ending Initialization Vector from the previously sent data block. This field is either empty, or contains a 16 byte hexadecimal value.

`MAC Length`

Field 4, the size of the MAC to be verified. The following table indicates the possible MAC sizes and the code to enter in this field for each one.

MAC Size	Code
More data expected; no MAC verified	0
32 bits	1
48 bits	2
64 bits	3

A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A - F) and written as two groups of four digits, separated by a space. A 48- or 64-bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

`Data Type`

Field 5, the data types are:

Data Type	Code
Unpacked ASCII hexadecimal	U
Binary	B

`Data Length`

Field 6, the data's length. This command will authenticate up to 4096 bytes of data. If more data is being sent in the next command – indicated by Field 4 being set to

zero – then the data length must be a multiple of eight for binary data, and a multiple of 16 for Unpacked ASCII data. If no more data is being sent, the Network Security Processor will right-pad the data with binary zeros (nulls, 0x00) such that the resulting data length will be a multiple of eight. This field contains a 1 to 4 byte decimal value.

`Data`

Field 7, the data to be authenticated. If the data is in unpacked ASCII hexadecimal format, this field can contain the numbers 0 through 9 and the characters A through F.

`[MAC]`

Field 8, the MAC to be verified when no more data is expected. A 32 bit MAC is expressed as eight hexadecimal digits and written as two groups of four digits, separated by a space. A 48- or 64-bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

This field must be empty if more data will be sent in a subsequent command.

`[Variant#]`

Field 9, the variant used to encrypt the MAC Key. This field is optional; if used, it can be one or two bytes long and can contain the numbers 3 or 19. If not used, the default variant, 3, is used.

Table 6-16. Command 99: Verify MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	99
1	$E_{MFK.V}$(KMAC)*	16, 32	0 - 9, A - F
2	[MAC Type]	0, 1	1 - 7
3	[$E_{MFK.6}$(IV)]	0, 16	0 - 9, A - F
4	MAC length	1	0 - 3
5	Data type	1	U, B
6	Data length	1 - 4	0 - 9
7	Data to be authenticated	1 - 4096	0 - 9, A - F (if unpacked ASCII)
8	[MAC]	0, 9, 14, 19	0 - 9, A - F
9	[Variant]	0 - 2	3, 19

*Can be a volatile table location.

Responding Parameters

`A9`

Field 0, the response identifier.

`MAC Length`

Field 1, the length of the MAC.

`Verification Flag or` $E_{MFK.6}$`(Ending IV)`

Field 2, if field 1 is set to 0 this field will contain the ending Initialization Vector. If field 1 is not set to 0, this field will contain the MAC verification flag. This field contains a 16 byte hexadecimal value, or “Y” or “N”.

If your use of this command results in the generation of an ending Initialization Vector in this field, use it as the starting Initialization Vector in subsequent MAC command to continue generating MACs.

If your use of this command results in a MAC verification flag, then this field will return Y if the MAC is verified or N if the MAC is not verified.

`KMAC Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the MAC Key. If option 88 is enabled, this field will contain the first six digits of the result.

`[MAC Type#]`

Field 4, the MAC Type. This field exists only if the MAC Type is 6 or 7.

Table 6-17. Response A9: Verify MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	A9
1	MAC length	1	0 - 3
2	Verification flag or $E_{MFK.6}$(Ending IV)	1, 16	0 - 9, A - F, Y, N
3	KMAC Check Digits	4 or 6	0 - 9, A - F
4*	[MAC Type#]	1	6, 7

* This field exists only when the MAC Type is 6 or 7.

Usage Notes

Before using Command 99, generate the MAC Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a MAC on Unpacked ASCII Data With Default IV

- Clear-text MAC Key: FEDC BA98 7654 3210.
  The MAC Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC length: 1.
- Data type: Unpacked (U).
- Data length: 6.
- Data: 303430.
- MAC: 60F0 EFDE.
- Variant: 3.

The command looks like this:

```
<99#1B866280C012DD33###1#U#6#303430#60F0 EFDE#3#>
```

The Network Security Processor returns the following response:

```
<A9#1#Y#A68C#>
```

Verifying Binary Data With Default IV

- Clear-text MAC Key: FEDC BA98 7654 3210.
  The MAC Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC length: 1.
- Data type: Binary (B).
- Data length: 3.
- Data: 040.
- MAC: 60F0 EFDE.
- Variant: 3.

The command looks like this:

```
<99#1B866280C012DD33###1#B#3#040#60F0 EFDE#3#>
```

The Network Security Processor returns the following response:

```
<A9#1#Y#A68C#>
```

Verifying a ISO - 9797-1 Algorithm 1 MAC using the Default IV and Variant

- Clear-text MAC Key: FEDC BA98 7654 3210.
  The MAC Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC Type : 6
- MAC length: 1.
- Data type: Unpacked (U).
- Data length: 6.
- Data: 303430.
- MAC: AFA3 9CEF.
- Variant: 3.

The command looks like this:

```
<99#1B866280C012DD3325166617EC743AB1#6##1#U#6#303430#
AFA3 9CEF#>
```

The Network Security Processor returns the following response:

```
<A9#1#Y#7B83#6#>
```

Verifying a ISO - 9797-1 Algorithm 3 MAC using the Default IV and Variant

- Clear-text MAC Key: FEDC BA98 7654 3210.
  The MAC Key encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- MAC Type : 7
- MAC length: 1.
- Data type: Unpacked (U).
- Data length: 18.
- Data: 3034 3030 3430 3034 30.
- MAC: B21A E4A4.
- Variant: 3.

The command looks like this:

```
<99#1B866280C012DD3325166617EC743AB1#7##1#U#18#
303430303430303430#B21A E4A4#>
```

The Network Security Processor returns the following response:

```
<A9#1#Y#7B83#7#>
```

Verify ACR (Atalla Challenge Response) Response MAC (Command 9B)

Command 9B verifies the response of the challenge number in both normal and auto mode for the ACR token.

This command supports only single-length working keys.

This command is enabled in the Network Security Processor's default security policy.

Command

<9B#$E_{MFK.7}$(KMAC$_R$)#Challenge No.#MAC#Response Format#[Mode]#>

Response

<AB#Verification Flag#Residual MAC#>[CRLF]

Calling Parameters

9B

Field 0, the command identifier.

$E_{MFK.7}$(KMAC$_R$)

Field 1, the KMACR Key encrypted under variant 7 of the MFK. This field contains a 16 byte hexadecimal value or a volatile table location.

Challenge No.

Field 2, the data, typically the challenge number that was used to compute the MAC.

This field can be four to 128 bytes long and can contain the numbers 0 through 9.

In auto mode, this field will be contain 8 characters for initialization, or 9 characters if it contains the previous response with a single challenge number.

MAC

Field 3, the MAC response to be verified. This field contains a 4 to 8 byte hexadecimal value.

Response Format

Field 4, the response format: H, D, or D :

Entering H in this field means that the response will be a hexadecimal value.

Entering D in this field means that the default decimalization table will be used to construct the response. The default decimalization table is: 0123456789222333.

To change the contents of the decimalization table, enter D where contains the numbers you want in the table. The response will be a decimal value.

`[Mode]`

Field 5, the mode flag. This field is optional.

Normal mode value is either 0, or empty.

Auto mode value is 1.

Table 6-18. Command 9B: Verify Response MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	9B
1	$E_{MFK.7}(KMAC_R)$*	16	0 - 9, A - F
2	Challenge number		
	Normal Mode Auto Mode	4 - 128 8, 9	0 - 9 0 - 9, A - F
3	MAC	4 - 8	0 - 9, A - F
4	Response format	1, 17	H, D, D
5	[Mode]	0,1	0, 1

*Can be a volatile table location.

Responding Parameters

`AB`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the MAC is verified; otherwise, it returns N.

`Residual MAC`

Field 2, a residue MAC; that is, the last 32 bits of the MAC result. This field contains an eight byte hexadecimal value.

If the MAC is not verified, this field returns XXXXXXXX, which indicates that the value was not verified.

A 16 byte number if MAC is verified in auto mode; the first 32 bits plus Residue MAC (last 32 bits). The MAC must be saved in Host DB for the next verification routine (Auto Mode keeps track of previous MAC results in continuos calculation). This response is not decimalized.

Table 6-19. Response AB: Verify Response MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AB
1	Verification flag	1	Y, N
2	Residual MAC	8, 16	0 - 9, A - F, X

Usage Notes

Before using Command 9B, generate the MAC Key.

ACR Token Auto Mode (types 2, 3, 6 and 7)

The ACR Token has six system fields.

The field lengths are 8 characters.

These fields are utilized in auto mode.

Up to six users can be assigned an ACR token.

ACR Token Auto Mode steps:

User enters the PIN followed by predefined single digit user system number (0-5). Then the user enters a predefined or single digit challenge number selected by the host, which is used to generate the Response.

The Response is generated from MAC processed data, 8 characters from system field 1, the single digit challenge number and 7 zeros.

The left 8 digits of this result are used as a Response or stored in the selected system field for the next operation.

Overview of Initialization of System Fields

When the Host Application needs to initialize or re-synchronize the auto mode operation, the host generates 8 digits for the challenge number and requires the user to enter the 8 digit challenge number instead of single digit.

The result of this 8 digit challenge (the response) is in turn stored in the system field.

The host saves the left 8 digits of result (from field 2 of response “AB”) in the data base, then is used as part of the challenge for the next operation.

ACR Sample Flow

Normal Mode.

This is a sample flow for normal mode operation for the ACR:

1. The user enters their PIN into the ACR.
2. The system prompts with a challenge number, typically 4 to 8 digits (created from Command 93).
3. The user enters the challenge number into the ACR.
4. The ACR responds with a Response MAC.
5. The user enters this Response into the system.
6. The system verifies that this Response MAC is correct and allows the user to continue logging on (Command 9B).

To calculate the Response, generate a MAC on the ASCII representation of the challenge number.

Auto (Single Digit) Mode.

This is a sample flow for INITIALIZING or RESYNCHRONIZATION of the ACR in single digit mode:

1. The user enters their PIN into the ACR (and optionally a system field number, the default system field number is 0)
2. The system prompts with an 8-digit challenge number (from Command 93). The user enters the challenge number into the ACR.
3. The ACR responds with a Response MAC. This response MAC is also stored in the selected system field for future use.
4. The user enters this Response into the system.
5. The system verifies that this Response MAC is correct and allows the user to continue logging on.

To calculate the Response, generate a MAC on the ASCII representation of the challenge number.

This is a sample flow for standard operation of the ACR in single digit mode:

1. The user enters their PIN into the ACR (and optionally a system field number; the default system field number is 0)
2. The user enters a predefined single digit challenge number into the ACR.
3. The ACR responds with a Response MAC which is calculated from the stored system field value and the entered single digit challenge. This response MAC is also stored in the selected system field for future use.
4. The user enters this Response into the system.
5. The system verifies that this Response MAC is correct and allows the user to continue logging on.

To calculate the Response, generate a MAC on the packed representation of the first 8 characters of the saved response (the response from the previous challenge/response session). Concatenate this with the single digit challenge and zero filled to 64 bits.

Examples

These detailed types show an example of Verifying with Hexadecimal MAC or Decimal MAC using either the Default Table or Custom Table.

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a Hexadecimal MAC

This example illustrates verifying a hexadecimal MAC, based on the following input:

- Clear-text MAC Key: 69EA 0A4E 73CF F9F0.
  The MAC Key encrypted under variant 7 of the MFK: 50CC BF0A A4DD 3A0A.
- Challenge number: 1487
- MAC response to be verified: 2B4F AB1A
- Response format: Hexadecimal (H)

The command looks like this:

```
<9B#50CCBF0AA4DD3A0A#1487#2B4FAB1A#H#>
```

The Network Security Processor issues the following response.

```
<AB#Y#3EAE165F#>
```

Verifying a Decimal MAC Using a Customized Table

- Clear-text MAC Key: 69EA 0A4E 73CF F9F0.
  The MAC Key encrypted under variant 7 of the MFK: 50CC BF0A A4DD 3A0A.
- Challenge number: 1618 5
- MAC response to be verified: 1111 1111
- Response format: Decimal (D) using the decimalization table: 1111 1111 1111 1111

The command looks like this:

```
<9B#50CCBF0AA4DD3A0A#16185#11111111#D1111111111111111#>
```

The Network Security Processor issues the following response.

```
<AB#Y#4DC6D4DE#>
```

Verify DUKPT MAC (Command 348)

Command 348 derives a message authentication session key using the Base Derivation Key and the key serial number, and then uses it to verify a MAC.

This command is enabled in the Network Security Processor's default security policy.

Command

<348#$E_{MFK.8}$(Base Derivation Key)#Key Serial Number#
[$E_{MFK.6}$(IV)]#Data Continuation#[MAC Type]#Data Length#
MAC Data#[MAC]#Session Key Length#>

Response

<448#Data Continuation#Verification Flag or Intermediate IV#
Base Derivation Key Check Digits#KMAC Check Digits#>

Calling Parameters

348

Field 0, the command identifier.

$E_{MFK.8}$(Base Derivation Key),MAC

Field 1, the Base Derivation Key encrypted under variant 8 of the MFK. This field contains a 32 byte value, or a volatile table location. If option A2 is set to "S" this field can contain a 16 byte value, a 1key-3DES (single-length) key.

Key Serial Number

Field 2, the 10 to 20 hexadecimal digit Key Serial Number (KSN) from the PIN entry device.

[$E_{MFK.6}$(IV)]

Field 3, the Initialization Vector (IV) encrypted under variant 6 of the MFK. If this command contains the first block of multiple blocks of data, or if you are authenticating only one block of data, then this field must be empty; the Network Security Processor will use its default Initialization Vector of all zeros. If this command contains data subsequent to the first block in a multi-block series (that is, it contains continuation data), then this field should contain the intermediate Initialization Vector from the previously sent data block. This field is either empty, or contains a 16 byte value.

`Data Continuation`

Field 4, if field 7 contains all the data to be used to verify the MAC, set this field to 1. If the amount of data to be used in the MAC verification process exceeds 4096 ASCII hexadecimal characters, multiple commands are required to process the MAC. If this is the case, set this field to 0 for all commands except the command that contains the final block of data, when processing the last block of data set this field to 1.

The value of this field can be either:

- 0 - More data is coming in a subsequent command
- 1 - This command contains all the data, or contains the last block of data

`[MAC Type]`

Field 5, the type of MAC to be calculated. The possible values for this field are:

MAC Type	Value
ISO - 9797-1 Algorithm 1 - 1key or 2key-3DES Cipher block chaining	Empty, or 1-6
ISO 9797-1 Algorithm 3 - Only the last data block is processed using 3DES, all previous blocks are processed using single DES	7
Visa DUKPT (old style) as generated by command 5C	V

`MAC Data Length`

Field 6, the number of bytes of data supplied in field 7. The minimum data length is 2, the maximum data length is 4096.

`MAC Data`

Field 7, the data in ASCII hexadecimal format that was used to generate the MAC. This field contains a 2 - 4096 hexadecimal character value. This field must contain an even number of hexadecimal characters.

`[MAC]`

Field 8, the MAC to be verified. This field must contain eight hexadecimal digits (32 bits), or must be empty if the Data Continuation flag (field 4) contains a 0 (zero).

`Session Key Length`

Field 9, the length of the generated incoming PIN Encryption and MAC session keys. The value of this field can be either:

- S - generate a 1key-3DES (single length) session key
- D - generate a 2key-3DES (double-length) session key.

If the Base Derivation Key, provided in field 1, is a 1key-3DES (single-length) key, this field must contain the letter S.

Table 6-20. Command 348: Verify DUKPT MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	348
1	$E_{MFK.8}$(Base Derivation Key)	16, 32	0 - 9, A - F
2	Key Serial Number	10 - 20	0 - 9, A - F
3	[$E_{MFK.6}$(IV)]	0, 16	0 - 9, A - F
4	Data Continuation	1	0, 1
5	[MAC Type]	0, 1	empty, 1 - 7, V
6	MAC Data Length	1 - 4	2 - 4096
7	MAC Data	2 - 4096	0 - 9, A - F
8	[MAC]	0, 8	empty, 0 - 9, A - F
9	Session Key Length	1	S or D

Responding Parameters

`448`

Field 0, the response identifier.

`Data Continuation`

Field 1, the value specified in field 4 of the command.

`Verification Flag or Intermediate IV`

Field 2, This field contains one of following values:

- Y – This value will be present only if field 4 of the command contains a 1, and the MAC verified.
- N – This value will be present only if field 4 of the command contains a 1, and the MAC did not verify.
- Intermediate IV - This value will be present only if field 4 of the command contains a zero. If present this field will contain the 16 hexadecimal character cryptogram of the intermediate IV.

`Base Derivation Key Check Digits`

Field 3, check digits of the base derivation key. Check digits are the first six digits that result from encrypting zeros using the base derivation key.

`KMAC Check Digits`

Field 4, check digits of the generated Message Authentication Key (KMAC). Check digits are the first six digits that result from encrypting zeros using the KMAC.

Table 6-21. Response 448: Verify DUKPT MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	448
1	Data Continuation	1	0, 1
2	Verification Flag or IV	1, 16	Y, N, or 0 - 9, A - F
3	Base Derivation Key Check Digits	6	0 - 9, A - F
4	KMAC Check Digits	6	0 - 9, A - F

Usage Notes

- Generate the cryptogram for the base derivation key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Base Derivation Key: 0123456789ABCDEF FEDCBA9876543210
  The Base Derivation Key encrypted under variant 8 of the MFK:
  AAA57E4E99AE9B03 28F6BA950E1664FA
- Key serial number: 9876543210E00012
- The command contains all the data: there is no IV
- The MAC Type: ISO 9797-1 Algorithm 1 - 3DES CBC
- The MAC data: 0123456789ABCEF
- The MAC to be verified: 6FCEDEBD
- The DUKPT session key length: 2key-3DES (double-length)

The command looks like this:

```
<348#AAA57E4E99AE9B0328F6BA950E1664FA#9876543210E00012##1##16
#0123456789ABCDEF#6FCEDEBD#D#>
```

The Network Security Processor returns the following response.

```
<448#1#Y#08D7B4#B97051#>
```

Generate DUKPT MAC (Command 386)

Command 386 derives a message authentication session key using the Base Derivation Key and the key serial number, and then uses it to generate a message authentication code (MAC).

This command is not enabled in the Network Security Processor's default security policy. To use this command you must add it to the Network Security Processor's security policy.

Command

<386#$E_{MFK.8}$(Base Derivation Key)#Key Serial Number#
[$E_{MFK.6}$(IV)]#Data Continuation#[MAC Type]#Data Length#
MAC Data#Session Key Length#>

Response

<486#Data Continuation#MAC or Intermediate IV#
Base Derivation Key Check Digits#KMAC Check Digits#>

Calling Parameters

386

Field 0, the command identifier.

$E_{MFK.8}$(Base Derivation Key)

Field 1, the Base Derivation Key encrypted under variant 8 of the MFK. This field contains a 32 byte value, or a volatile table location. If option A2 is set to "S" this field can contain a 16 byte value, a 1key-3DES (single-length) key.

Key Serial Number

Field 2, the 10 to 20 hexadecimal digit Key Serial Number (KSN) from the PIN entry device.

[$E_{MFK.6}$(IV)]

Field 3, the Initialization Vector (IV) encrypted under variant 6 of the MFK. If this command contains the first block of multiple blocks of data, or if you are authenticating only one block of data, then this field must be empty; the Network Security Processor will use its default Initialization Vector of all zeros. If this command contains data subsequent to the first block in a multi-block series (that is, it contains continuation data), then this field should contain the intermediate

Initialization Vector from the previously sent data block. This field is either empty, or contains a 16 byte value.

`Data Continuation`

Field 4, If field 7 contains all the data to be used to verify the MAC, set this field to 1. If the amount of data to be used in the MAC verification process exceeds 4096 ASCII hexadecimal characters multiple commands are required to process the MAC. If this is the case, set this field to 0 for all commands except the command that contains the final block of data, when processing the last block of data set this field to 1.

The value of this field can be either:

- 0 - More data is coming in a subsequent command
- 1 - This command contains all the data, or contains the last block of data

`[MAC Type]`

Field 5, the type of MAC to be calculated. The possible values for this field are:

MAC Type	Value
ISO - 9797-1 Algorithm 1 - 1key or 2key-3DES Cipher block chaining	Empty, or 1-6
ISO 9797-1 Algorithm 3 - Only the last data block is processed using 3DES, all previous blocks are processed using single DES	7
Visa DUKPT (old style) as generated by command 5C. The NSP will return the left half of the MAC if this field contains “VL”, it will return the right half of the MAC if this field contains “VR”.	V, VL or VR

`MAC Data Length`

Field 6, the number of bytes of data supplied in field 7. The minimum data length is 2, the maximum data length is 4096.

`MAC Data`

Field 7, the data in ASCII hexadecimal format that was used to generate the MAC. This field contains a 2 - 4096 hexadecimal character value. This field must contain an even number of hexadecimal characters.

`Session Key Length`

Field 8, the length of the generated incoming PIN Encryption and MAC session keys. The value of this field can be either:

- S - generate a 1key-3DES (single length) session key
- D - generate a 2key-3DES (double-length) session key.

If the Base Derivation Key, provided in field 1, is a 1key-3DES (single-length) key, this field must contain the letter S.

Table 6-22. Command 386: Generate DUKPT MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	386
1	$E_{MFK.8}$(Base Derivation Key)	16, 32	0 - 9, A - F
2	Key Serial Number	10 - 20	0 - 9, A - F
3	[$E_{MFK.6}$(IV)]	0, 16	0 - 9, A - F
4	Data Continuation	1	0, 1
5	[MAC Type]	0, 1	empty, 1 - 7, V
6	MAC Data Length	1 - 4	2 - 4096
7	MAC Data	2 - 4096	0 - 9, A - F
8	Session Key Length	1	S or D

Responding Parameters

`486`

Field 0, the response identifier.

`Data Continuation`

Field 1, the value specified in field 4 of the command.

`MAC or Intermediate IV`

Field 2, if field 4 of the command contains a 1, this field contains the 32 bit MAC represented as 8 hexadecimal characters. If field 4 of the command contains a 0 (zero) this field will contain the 16 hexadecimal character cryptogram of the intermediate IV.

`Base Derivation Key Check Digits`

Field 3, check digits of the base derivation key. Check digits are the first six digits that result from encrypting zeros using the base derivation key.

`KMAC Check Digits`

Field 4, check digits of the generated Message Authentication Key (KMAC). Check digits are the first six digits that result from encrypting zeros using the KMAC.

Table 6-23. Response 486: Generate DUKPT MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	486
1	Data Continuation	1	0, 1
2	MAC	8	0 - 9, A - F
	or	or	or
	Intermediate IV	16	0 - 9, A - F
3	Base Derivation Key Check Digits	6	0 - 9, A - F
4	KMAC Check Digits	6	0 - 9, A - F

Usage Notes

- Generate the cryptogram for the Base Derivation Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Base Derivation Key: 0123456789ABCDEF FEDCBA9876543210
  The Base Derivation Key encrypted under variant 8 of the MFK:
  AAA57E4E99AE9B03 28F6BA950E1664FA
- Key serial number: 9876543210E00012
- The command contains all the data: there is no IV
- The MAC Type: ISO 9797-1 Algorithm 1 - 3DES CBC
- The MAC data: 0123456789ABCEF
- The DUKPT session key length: 2key-3DES (double-length)

The command looks like this:

```
<386#AAA57E4E99AE9B0328F6BA950E1664FA#9876543210E00012##1##16
#0123456789ABCDEF#D#>
```

The Network Security Processor returns the following response.

```
<486#1#6FCEDEBD#08D7B4#B97051#>
```

7 Authorizing VISA, MasterCard, American Express, and Discover Cards

VISA and MasterCard magnetic stripe card transactions are authorized using the same algorithm. When the algorithm is used for VISA transactions, it is called Card Verification Value (CVV). When the algorithm is used for MasterCard transactions, it is called Card Validation Code. American Express uses a different algorithm called a Card Security Code (CSC).

Mastercard PayPass transactions are protected using a value called the CVC3, which is generated by the PayPass chip for each transaction. VISA uses a dynamic Card Verification Value (dCVV) generated by the smartcard to protect contactless smartcard transactions.

This section explains the purpose of CVVs, dCVVs, CVCs, CVC3s, and CSCs, and describes the commands that are used to implement support for CVV/CVC/CSCs.

To skip this introduction go to Table 7-1, CVV, dCVV, CVC, CVC3 and CSC Commands for a list of commands.

About CVVs, CVCs, and CSCs

VISA Card Verification Values (CVVs), MasterCard Card Validation Codes (CVCs), and American Express Card Security Code (CSCs) are check-values that confirm the validity of a bankcard's magnetic stripe. Confirming the magnetic stripe's validity protects against the production of counterfeit cards that have account numbers which have been generated in sequential order based on the account number taken from a valid card.

The CVV/CVC algorithm takes as its input the primary account number, expiration date, and service code. These values are on the magnetic stripe's first two tracks. The input is operated on by keys, referred to as $KCVV_A$ and $KCVV_B$. The result – the CVV/CVC – is added to the card's magnetic stripe. For calculating the encoded CVC1, use the primary account number, card expiration date and the service code. For calculating the indent CVC2, use the primary account number, card expiration date, and "zero fill" the service code.

A static CVC3 uses the same algorithm as CVC1 and CVC2, the data inputs are the primary account number, card expiration date, and a service code value of 502. A static CVC3 can be generated or verified using commands 5D and 5E, respectively.

A dynamic CVC3 uses a different algorithm, the data inputs are the primary account number, card expiration date, service code, unpredictable number, and application transaction counter, value. Use command 359 to verify a dynamic CVC3.

Note. For specific applications Visa refers to the CVV by other similar names. The Cardholder Authentication Verification Value (CAVV) uses the same algorithm and data values as those used to generate and verify a CVV. The Integrated Card Verification Value (iCVV) also uses the CVV algorithm with a service code of '999'.

The CSC algorithm takes as its input the primary account number and expiration date. These values are on the magnetic stripe's first two tracks. The input is operated on by a 2key-3DES (double-length) key, referred to as KCSC. The result – the CSC– is added to the card's magnetic stripe.

The Discover algorithm is unique to Discover smartcards. Use command 35F to verify a Discover dynamic CVV.

CVV, dCVV, CVC, CVC3, and CSC Commands

The remainder of this section contains the command and response syntax for the VISA CVV, MasterCard CVC, and American Express CSC commands.

Quick Reference

Table 7-1 identifies each command by number, name, and purpose.

Table 7-1. CVV, dCVV, CVC, CVC3 and CSC Commands

Command #	Name	Purpose
5D	Generate CVV/CVC	Generates a Card Verification Value/Card Validation Code
5E	Verify CVV/CVC	Verifies a Card Verification Value/Card Verification Code
357	Verify dCVV	Verifies a VISA dynamic Card Verification Value
359	Verify CVC3	Verifies a MasterCard CVC3
35A	Verify CSC	Verifies a Card Security Codes
35B	Generate CSC	Generates Card Security Codes
35F	Verify DCVV	Verifies a Discover Dynamic Card Verification Value
36A	Verify AMEX Expresspay - Magstripe	Verifies an AMEX Expresspay value using the Magstripe mode

Generate CVV/CVC (Command 5D)

Command 5D generates a Visa Card Verification Value (CVV) or a MasterCard Card Validation Code (CVC). Visa and MasterCard use the same algorithm to generate their CVV or CVC value. Whenever the terms Card Verification Value or CVV are used in this manual, they also refer to Card Validation Code and CVC.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

This command supports only 1key-3DES (single-length) working keys.

Command

<5D#Algorithm Identifier#$E_{MFK.3}$(KCVV$_A$)#$E_{MFK.3}$(KCVV$_B$)#Data#>

Response

<6D#CVV#KCVV$_A$ Check Digits#KCVV$_B$ Check Digits#>[CRLF]

Calling Parameters

5D

Field 0, the command identifier.

Algorithm Identifier

Field 1, the algorithm identifier. This field may contain either 2 or 3. The standard algorithm for CVV is 3. Algorithm 2 is no longer recommended. A three alphanumeric character CVV is returned in field one of the response when the algorithm identifier is set to 2. An eight digit CVV is returned in field one of the response when the algorithm identifier is set to 3.

$E_{MFK.3}$(KCVV$_A$)

Field 2, the Card Verification Value Key A encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.3}$(KCVV$_B$)

Field 3, the Card Verification Value Key B encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Data

Field 4, the data used to generate the Card Verification Value. The data in this field should be the primary account number, the card expiration date, and the service code. This field contains a 1 to 32 byte decimal value.

Table 7-2. Command 5D: Generate CVV/CVC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	5D
1	Algorithm identifier	1	2, 3
2	$E_{MFK.3}(KCVV_A)$*	16	0 - 9, A - F
3	$E_{MFK.3}(KCVV_B)$*	16	0 - 9, A - F
4	Data	1 - 32	0 - 9

*Can be a volatile table location.

Responding Parameters

`6D`

Field 0, the response identifier.

`CVV`

Field 1, the generated Card Verification Value. When the algorithm identifier (specified in field one of the command) is two, this field will contain three alphanumeric characters. When the algorithm identifier is three, this field will contain 8 decimal digits.

`KCVV`$_A$ `Check Digits`

Field 2, the Card Verification Value Key A check digits; the first four digits that result from encrypting zeros using the Card Verification Value Key A. If option 88 is enabled, this field will contain the first six digits of the result.

`KCVV`$_B$ `Check Digits`

Field 3, the Card Verification Value Key B check digits; the first four digits that result from encrypting zeros using the Card Verification Value Key B. If option 88 is enabled, this field will contain the first six digits of the result.

Table 7-3. Response 6D: Generate CVV/CVC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	6D
1	CVV	varies	0 - 9, A - Z
2	$KCVV_A$ Check Digits	4 or 6	0 - 9, A - F
3	$KCVV_B$ Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Before using Command 5D generate the two Card Verification Value Keys.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Card Verification Value Key A: 0123 4567 89AB CDEF.
  The Card Verification Value Key A encrypted under variant 3 of the MFK: 2516 6617 EC74 3AB1.
- Clear-text Card Verification Value Key B: FEDC BA98 7654 3210.
  The Card Verification Value Key A encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- The data used to generate the Card Verification Value is 4123 4567 8901 2345 8701 101. This value includes the following information:
- Primary account number: 4123 4567 8901 2345.
- Card expiration date: 8701.
- Service code: 101.

The command looks like this:

```
<5D#3#25166617EC743AB1#1B866280C012DD33#
4123456789012345870110 1#>
```

The Network Security Processor returns the following response:

```
<6D#56149820#D5D4#A68C#>
```

Verify CVV/CVC (Command 5E)

Command 5E verifies a Visa Card Verification Value (CVV) or a MasterCard Card Validation Code (CVC). Visa and MasterCard use the same algorithm to verify their CVV or CVC value. Whenever the terms Card Verification Value or CVV are used in this manual, they also refer to Card Validation Code and CVC.

This command is enabled in the Network Security Processor's default security policy.

Command

<5E#Algorithm#$E_{MFK.3}$(KCVV$_A$)#$E_{MFK.3}$(KCVV$_B$)#Data#CVV#>

Response

<6E#Verification Flag#KCVV$_A$ Check Digits#KCVV$_B$ Check Digits#>

[CRLF]

Calling Parameters

5E

Field 0, the command identifier.

Algorithm

Field 1, the algorithm identifier. This field may contain either 2 or 3. The standard algorithm is 3. Algorithm 2 is no longer recommended.

$E_{MFK.3}$(KCVV$_A$)

Field 2, the Card Verification Value Key A encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{MFK.3}$(KCVV$_B$)

Field 3, the Card Verification Value Key B encrypted under variant 3 of the MFK. This field contains a 16 byte hexadecimal value, or a volatile table location.

Data

Field 4, the data used to verify the Card Verification Value. The data in this field should be the primary account number, the card expiration date, and the service code. This field contains a 1 to 32 byte decimal value.

CVV

Field 5, the Card Verification Value to be verified. When the algorithm identifier (field 1) is set to two, this field must contain a three alphanumeric character value.

When the algorithm identifier (field 1) is set to three and option 4D is enabled, this field must contain a 3 to 8 digit value.

Table 7-4. Command 5E: Verify CVV/CVC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	5E
1	Algorithm identifier	1	2, 3
2	$E_{MFK.3}(KCVV_A)$*	16	0 - 9, A - F
3	$E_{MFK.3}(KCVV_B)$*	16	0 - 9, A - F
4	Data	1 - 32	0 - 9
5	CVV	varies	0 - 9, A - Z

*Can be a volatile table location.

Responding Parameters

6E

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the CVV verified; otherwise, it returns N.

`KCVV`$_A$ `Check Digits`

Field 2, the Card Verification Value Key A check digits; the first four digits that result from encrypting zeros using the Card Verification Value Key A. If option 88 is enabled, this field will contain the first six digits of the result.

`KCVV`$_B$ `Check Digits`

Field 3, the Card Verification Value Key B check digits; the first four digits that result from encrypting zeros using the Card Verification Value Key B. If option 88 is enabled, this field will contain the first six digits of the result.

Table 7-5. Response 6E: Verify CVV/CVC

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	6E
1	Verification flag	1	Y, N
2	$KCVV_A$ Check Digits	4 or 6	0 - 9, A - F
3	$KCVV_B$ Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Generate the Card Verification Value key pair.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Card Verification Value Key A: 0123 4567 89AB CDEF.
  The Card Verification Value Key A encrypted under variant 3 of the MFK: 2516 6617 EC74 3AB1.
- Clear-text Card Verification Value Key B: FEDC BA98 7654 3210.
  The Card Verification Value Key B encrypted under variant 3 of the MFK: 1B86 6280 C012 DD33.
- The data used to generate the Card Verification Value is 4123 4567 8901 2345 8701 101. This value includes the following information:
- Primary account number: 4123 4567 8901 2345.
- Card expiration date: 8701.
- Service code: 101.
- The Card Verification Value to be verified: 56149820

The command looks like this:

```
<5E#3#25166617EC743AB1#1B866280C012DD33#
4123456789012345870110 1#56149820#>
```

The Network Security Processor returns the following response:

```
<6E#Y#D5D4#A68C#>
```

Verify dCVV (Command 357)

Command 357 verifies a VISA dynamic Card Verification Value generated by a contactless smartcard. This command is enabled in the Network Security Processor's default security policy.

Command

<357#$E_{MFK.9}$(IMK$_{CVV}$)#PAN#PAN Sequence Number#Expiration Date#
Service Code#ATC#dCVV#>

Response

<457#Verification Flag#UDK Check Digits#>[CRLF]

Calling Parameters

357

Field 0, the command identifier.

$E_{MFK.9}$(IMK$_{CVV}$)

Field 1, the double-length Issuer Master Key encrypted under variant 9 of the MFK. This field contains a 32 byte hexadecimal value. A replicated single-length Issuer Master Key is supported only if option 6A is enabled.

PAN

FIeld 2, Primary Account Number. This field contains a 3 through 19 digit value.

PAN Sequence Number

Field 3, the two digit sequence number which is appended to the PAN.

Expiration Date

Field 4, the four digit expiration date.

Service Code

Field 5, the three digit service code.

ATC

Field 6, the three or four digit Application Transaction Counter.

dCVV

Field 7, the three digit dynamic Card Verification Value to be verified.

Table 7-6. Command 357: Verify dCVV

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	357
1	$E_{MFK.9}(IMK_{CVV})$	32	0 - 9, A - F
2	PAN	3 - 19	0 - 9
3	PAN Sequence Number	2	0 - 9
4	Expiration Date	4	0 - 9
5	Service Code	3	0 - 9
6	ATC	3 - 4	0 - 9
7	dCVV	3	0 - 9

Responding Parameters

`457`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the dCVV is verified; otherwise, it returns N.

`UDK Check Digits`

Field 2, the unique derived key check digits; the first four digits that result from encrypting zeros using the unique derived key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 7-7. Response 457: Verify dCVV

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	457
1	Verification flag	1	Y, N
2	UDK Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 357 generate the Issuer Master Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Issuer Master Key: 0123456789ABCDEF FEDCBA9876543210, check digits 08D7. The Issuer Master Key encrypted under variant 9 of the MFK: 94E1BA8235D38B089AC5BBD4F34C67E8
- Primary account number: 0123456789
- PAN sequence number: 00
- Card expiration date: 1204
- Service code: 555
- ATC: 666
- The dCVV to be authenticated: 505

The command looks like this:

```
<357#94E1BA8235D38B089AC5BBD4F34C67E8#0123456789#00#1204#555#
666#505#>
```

The Network Security Processor returns the following response:

```
<457#Y#677E9A#>
```

Verify dynamic CVC3 (Command 359)

Command 359 verifies a MasterCard dynamic Card Verification Code 3 (CVC3) generated by a PayPass smartcard. This command is enabled in the Network Security Processor's default security policy.

Command

<359#$E_{MFK.9}$(IMK$_{CVC3}$)#PAN#PAN Sequence Number#Track 1/2 Data#
Unpredictable Number#ATC#dynamic CVC3#>

Response

<459#Verification Flag#UDK Check Digits#>[CRLF]

Calling Parameters

359

Field 0, the command identifier.

$E_{MFK.9}$(IMK$_{CVC3}$)

Field 1, the double-length Issuer Master Key for dynamic CVC3 encrypted under variant 9 of the MFK. This field contains a 32 byte hexadecimal value. A replicated single-length Issuer Master Key is supported only if option 6A is enabled.

PAN

Field 2, Primary Account Number. This field contains a 1 through 19 digit value.

PAN Sequence Number

Field 3, the two hexadecimal digit sequence number which is appended to the PAN.

Track 1/2 Data

Field 4, the track 1 or track 2 data used to generate the dynamic CVC3. Track 1 data must be supplied as the hexadecimal representation of ASCII characters. For example, the number '5' is converted to 0x35, the letter 'A' is converted to 0x41. Track 2 data must be supplied as hexadecimal characters. If the track 2 data length is not an even number append a hexadecimal 'F'. The maximum length of this field is 160 hexadecimal characters. The length of this field must be a multiple of 16. The track data must be padded per these steps:

1. If the track length is a multiple of 16, add these 16 pad digits 8000000000000000, then go to step 4. If not go to step 2.

2. If the track length is not a multiple of 16, add these two pad digits ‘80’, then go to step 3.
3. If the padded track data is a multiple of 16 go to step 4. If not, it is right-padded with hexadecimal zeroes until it is a multiple of 16. Go to step 4.
4. The padding is complete.

`Unpredictable Number`

Field 5, the 8 digit unpredictable number.

`ATC`

Field 6, the 4 hexadecimal digit Application Transaction Counter.

`dynamic CVC3`

Field 7, the three to five digit dynamic Card Validation Code 3 value to be verified.

Table 7-8. Command 359: Verify dynamic CVC3

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	359
1	$E_{MFK.9}(IMK_{CVC3})$	32	0 - 9, A - F
2	PAN	1 - 19	0 - 9
3	PAN Sequence Number	2	0 - 9, A - F
4	Track1/2 Data	16 - 160	0 - 9, A - F
5	Unpredictable Number	8	0 - 9
6	ATC	4	0 - 9, A - F
7	dynamic CVC3	3 - 5	0 - 9

Responding Parameters

`459`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the dynamic CVC3 is verified; otherwise, it returns N.

`UDK Check Digits`

Field 2, the unique derived key check digits; the first four digits that result from encrypting zeros using the unique derived key A. If option 88 is enabled, this field will contain the first six digits of the result.

Table 7-9. Response 459: Verify dynamic CVC3

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	459
1	Verification flag	1	Y, N
2	UDK Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 359 generate the Issuer Master Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Issuer Master Key: 0123456789987654 3210012345678998, check digits 48F2 . The Issuer Master Key encrypted under variant 9 of the MFK: 55D3D62D30591D7F1A6E62FD623A4CD0

Track 1 example

- Primary account number: 5413123456784808
- PAN sequence number: 00
- Track 1 data: B5413123456784808^SUPPLIED/NOT^0906101330003330002222200011110

  Track 1 data in hexadecimal: 42 35 34 31 33 31 32 33 34 35 36 37 38 34 38 30 38 5E 53 55 50 50 4C 49 45 44 2F 4E 4F 54 5E 30 39 30 36 31 30 31 33 33 30 30 30 33 33 33 30 30 30 32 32 32 32 32 30 30 30 31 31 31 31 30
- Unpredictable Number: 00000899
- ATC: 005E
- The dynamic CVC3 to be verified: 587

The command looks like this:

```
<359#55D3D62D30591D7F1A6E62FD623A4CD0#5413123456784808#00#423
534313331323334353637383430385E535550504C4945442F4E4F545E30
393036313031333330303033333330303032323232303030313131308
000#00000899#005E#587#>
```

The Network Security Processor returns the following response:

```
<459#Y#AF59#>
```

Track 2 example

- Primary account number: 5413123456784808
- PAN sequence number: 00
- Track 2 data in hexadecimal: 54 13 12 34 56 78 48 08 D0 90 61 01 90 00 99 00 00 00 0F
- Unpredictable Number: 00000899
- ATC: 005E
- The dynamic CVC3 to be verified: 572

The command looks like this:

```
<359#55D3D62D30591D7F1A6E62FD623A4CD0#5413123456784808#00#541
3123456784808D0906101900099000000F8000000000#00000899#005E#5
72#>
```

The Network Security Processor returns the following response:

```
<459#Y#AF59#>
```

Verify AMEX CSC (Command 35A)

Command 35A verifies the American Express Card Security Codes (CSC). This command supports any combination of 3 digit, 4 digit and 5 digit CSC values.

This command supports a 2key-3DES (double-length) KCSC. Option 6A determines if the 2key-3DES (double-length) key is allowed to have identical halves. If option 6A is disabled this command will require a true 2key-3DES (double-length) key that has different values for each half. If option 6A is enabled, no checks will be performed and a 1key-3DES (single-length) key can be replicated to make a 2key-3DES (double-length) key.

In version 1.30 and above this command can be used to verify either a version 1.0 or version 2.0 CSC.

This command is enabled in the Network Security Processor's default security policy.

Command

<35A#Data#[Expiration Date]#$E_{MFK.3}$(KCSC)#[CSC-5]#[CSC-4]#
[CSC-3]#>

Response

<45A#[VF5]#[VF4]#VF3]#KCSC Check Digits#>[CRLF]

Calling Parameters

35A

Field 0, the command identifier.

Data

Field 1, the contents of this field depends on the version of CSC to be verified.

To verify a CSC version 1.0 value the 15 digit Primary Account Number (PAN) is entered in this field. The leftmost two digits must be either 34 or 37.

To verify a CSC version 2.0 value this field must contain two 16 digit account blocks (32 total digits). Account block 1 is the 4 digit expiration date followed by digits 3 through 14 of the PAN. Account block 2 is the 3 digit service coded right-padded with zeros.

Example: Expiration date = 9912, PAN = 375987654321001, Service Code = 992.

Account block 1 = 9912598765432100
Account block 2 = 9920000000000000

Field 1 Data = 9912598765432109920000000000000

`[Expiration Date]`

Field 2, the expiration date is entered in the YYMM format. This field contains a 4 byte decimal value. This field is ignored if field 1 contains a 32 byte decimal value.

$E_{MFK.3}$(KCSC)

Field 3, the KCSC encrypted under variant 3 of the MFK. This field contains a 32 byte hexadecimal value, or a key table index.

`[CSC-5]`

Field 4, contains the 5 digit CSC, or this field may be empty.

`[CSC-4]`

Field 5, contains the 4 digit CSC, or this field may be empty.

`[CSC-3]`

Field 6, contains the 3 digit CSC, or this field may be empty.

Table 7-10. Command 35A: Verify AMEX CSC

Field	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	35A
1	Data	15, 32	0 - 9
2	[Expiration Date]	4,0	0 - 9
3	$E_{MFK.3}$(KCSC)	32*	0 - 9, A - F
4	[CSC-5]	0, 5	0 - 9
5	[CSC-4]	0, 4	0 - 9
6	[CSC-3]	0. 3	0 - 9

*Can be a volatile table location.

Responding Parameters

`45A`

Field 0, the response identifier.

`[VF5]`

Field 1, the Verify Flag result for the CSC-5 value. This field contains a Y if the CSC-5 is verified, or a N if the CSC-5 is not verified. The field is empty if field 4 of the command was empty.

`[VF4]`

Field 2, the Verify Flag result for the CSC-4 value. This field contains a Y if the CSC-4 is verified or a N if the CSC-4 is not verified. The field is empty if field 5 of the command was empty.

`[VF3]`

Field 3, the Verify Flag result for the CSC-3 value. This field contains a Y if the CSC-3 is verified or a N if the CSC-5 is not verified. The field is empty if field 6 of the command was empty.

`KCSC Check Digits`

Field 4, check digits; the first four digits that result from encrypting zeros using the KCSC. If option 88 is enabled, this field will contain the first six digits of the result.

Table 7-11. Response 45A: Verify AMEX CSC

Field	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	45A
1	[VF5]	0, 1	Y, N
2	[VF4]	0, 1	Y, N
3	[VF3]	0, 1	Y, N
4	KCSC Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 35A generate Card Security Code Key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Card Security Code Key (KCSC) is:
  0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The Card Security Code Key (KCSC) encrypted under variant 3 of the MFK:
  25166617EC743AB11B866280C012DD33

This example illustrates verifying version 1.0 CSCs.

- Account number: 371234567890123
- Card expiration date: 9912
- CSC-5 = 61247, CSC-4 = 8720, CSC-3 = 552

The command looks like this:

```
<35A#371234567890123#9912#25166617EC743AB11B866280C012DD33#61
247#8720#552#>
```

The Network Security Processor returns the following response:

```
<45A#Y#Y#Y#08D7#>
```

This example illustrates verifying version 2.0 CSCs.

- Account number: 375987654321001
- Card expiration date: 9912
- Service Code: 992
- 5-digit CSC = 72417, 4-digit CSC = 7998, 3-digit CSC = 746

The command looks like this:

```
<35A#991259876543210099200000000000000##25166617EC743AB11B8662
80C012DD33#72417#7998#746#>
```

The Network Security Processor returns the following response:

```
<45A#Y#Y#Y#08D7#>
```

Generate AMEX CSC (Command 35B)

Command 35B generates the American Express Card Security Codes (CSC). The CSC algorithm produces three codes; a 5-digit CSC, a 4-digit CSC, and 3-digit CSC.

This command supports a 2key-3DES (double-length) KCSC. Option 6A determines if the 2key-3DES (double-length) key is allowed to have identical halves. If option 6A is disabled this command will require a true 2key-3DES (double-length) key that has different value for each half. If option 6A is enabled, no checks will be performed and a 1key-3DES (single-length) key can be replicated to make a 2key-3DES (double-length) key.

In version 1.30 and above this command can be used to generate either version 1.0 or version 2.0 CSC values.

This command has a high security exposure, it is not enabled in the Network Security Processor's default security policy.

Command

<35B#Data#[Expiration Date]#$E_{MFK.3}$(KCSC)#>

Response

<45B#CSC-5#CSC-4#CSC-3#KCSC Check Digits#>[CRLF]

Calling Parameters

35B

Field 0, the command identifier.

Data

Field 1, the contents of this field depends on the version of CSC to be verified.

To generate CSC version 1.0 values the 15 digit Primary Account Number (PAN) is entered in this field. The leftmost two digits must be either 34 or 37.

To generate CSC version 2.0 values this field must contain two 16 digit account blocks (32 total digits). Account block 1 is the 4 digit expiration date followed by digits 3 through 14 of the PAN. Account block 2 is the 3 digit service coded right-padded with zeros.

Example: Expiration date = 9912, PAN = 375987654321001, Service Code = 992.

Account block 1 = 9912598765432100
Account block 2 = 9920000000000000

Field 1 Data = 9912598765432109920000000000000

`[Expiration Date]`

Field 2, the expiration date is entered in the YYMM format. This field contains a 4 byte decimal value. This field is ignored if field 1 contains a 32 byte decimal value.

$E_{MFK.3}$(KCSC)

Field 3, the KCSC encrypted under variant 3 of the MFK. This field contains a 32 byte hexadecimal value, or a volatile table location.

Table 7-12. Command 35B: Generate AMEX CSC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	35B
1	Data	15, 32	0 - 9
2	[Expiration Date]	4, 0	0 - 9
3	$E_{MFK.3}$(KCSC)*	32	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`45B`

Field 0, the response identifier.

`CSC-5`

Field 1, the generated 5 digit CSC. This field contains a 5 byte decimal value.

`CSC-4`

Field 2, the generated 4 digit CSC. This field contains a 4 byte decimal value.

`CSC-3`

Field 3, the generated 3 digit CSC. This field contains a 3 byte decimal value.

`KCSC Check Digits`

Field 4, check digits; the first four digits that result from encrypting zeros using the KCSC. If option 88 is enabled, this field will contain the first six digits of the result.

.

Table 7-13. Response 45B: Generate AMEX CSC (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	45B
1	5 Digit CSC	5	0-9

Table 7-13. Response 45B: Generate AMEX CSC (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
2	4 Digit CSC	4	0-9
3	3 Digit CSC	3	0-9
4	KCSC Check Digits	4 or 6	0 - 9, A - F

Usage Notes

- Before using this command, generate the KCSC key.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

- Clear-text Card Security Code Key (KCSC) is:
  0123 4567 89AB CDEF FEDC BA98 7654 3210.
  The Card Security Code Key (KCSC) encrypted under variant 3 of the MFK:
  25166617EC743AB11B866280C012DD33

This example illustrates generating version 1.0 CSCs.

- Account number: 371234567890123
- Card expiration date: 9912

The command looks like this:

```
<35B#371234567890123#9912#25166617EC743AB11B866280C012DD33#>
```

The Network Security Processor returns the following response:

```
<45B#61247#8720#552#08D7#>
```

This example illustrates generating version 2.0 CSCs.

- Account number: 375987654321001
- Card expiration date: 9912
- Service Code: 992

The command looks like this:

```
<35B#991259876543210099200000000000000##25166617EC743AB11B8662
80C012DD33#>
```

The Network Security Processor returns the following response:

```
<45B#72417#7998#746#08D7#>
```

Verify Discover DCVV (Command 35F)

This command verifies a Discover Dynamic Card Verification Value (DCVV) generated by a contactless smartcard.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<35F#E_MFK.9(IMK-DCVV)#PAN#Expiration Date#UN#ATC#DCVV#>
```

Response

```
<45F#Verification Flag#IMK-DCVV Check Digits#
AUK-DCVV Check Digits#>[CRLF]
```

Calling Parameters

`35F`

Field 0, the command identifier.

$E_{MFK.9}$`(IMK-DCVV)`

Field 1, the Issuer Master Key for DCVV encrypted under variant 9 of the MFK. The IMK-DCVV is a 2key-3DES (double-length) key. Option 6A determines if the 2key-3DES (double-length) key is allowed to have identical halves. If option 6A is disabled this command will require a true 2key-3DES (double-length) key that has different value for each half. If option 6A is enabled, no checks will be performed and a 1key-3DES (single-length) key can be replicated to make a 2key-3DES (double-length) key. This field contains a 32 byte hexadecimal value.

`PAN`

Field 2, the Primary Account Number. This field contains a 14, 16, or 18 byte decimal value.

`Expiration Date`

Field 3, the expiration date is entered in the YYMM format. This field contains a 4 byte decimal value.

`UN`

Field 4, the unpredictable number. This field contains a 2 byte decimal value.

ATC

Field 5, the application transaction counter. This field contains a 4 byte decimal value.

DCVV

Field 6, the dynamic card verification value. This field contains a 3 byte decimal value.

Table 7-14. Command 35F: Verify Discover DCVV

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	35F
1	$E_{MFK.9}$(IMK-DCVV),MAC	32	0 - 9, A - F
2	PAN	14, 16 or 18	0 - 9
3	Expiration Date	4	0 - 9
4	UN	2	0 - 9
5	ATC	4	0 - 9
6	DCVV	3	0 - 9

Responding Parameters

45F

Field 0, the response identifier.

Verification Flag

Field 1, the verification flag. This field returns Y if the DCVV is verified; otherwise, it returns N.

IMK-DCVV Check Digits

Field 2, the first four digits that result from encrypting zeros using the Issuer Master Key-DCVV. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the Issuer Master Key-DCVV.

AUK-DCVV Check Digits

Field 3, the first four digits that result from encrypting zeros using the derived Account Unique Key-DCVV (AUK-DCVV). If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the AUK-DCVV.

.

Table 7-15. Response 45F: Verify Discover DCVV

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	45F
1	Verification Flag	1	Y or N
2	IMK-DCVV Check Digits	4, 6	0 - 9, A - F
3	AUK-DCVV Check Digits	4, 6	0 - 9, A - F

Usage Notes

- Before using this command, generate the IMK-DCVV key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Using a 2key-3DES IMK-DCVV to verify a Discover DCVV

- Clear-text Issuer Master Key-DCVV (IMK-DCVV) is:
  0123456789ABCDEF FEDCBA9876543210, check digits = 08D7
  IMK-DCVV encrypted under variant 3 of the MFK:
  94E1BA8235D38B08 9AC5BBD4F34C67E8
- PAN: 6011111111111117
- Expiration date: 0801
- Unpredictable number: 56
- Application transaction counter: 1234
- DCVV: 204

The command looks like this:

```
<35F#94E1BA8235D38B089AC5BBD4F34C67E8#6011111111111117#0801#
56#1234#204#>
```

The Network Security Processor returns the following response:

```
<45F#Y#08D7#A522#>
```

Verify AMEX Expresspay value - Magstrip Mode (Command 36A)

This command verifies an American Express Expresspay value using the Magstripe mode.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<36A#Derivation Type#E_MFK.9(IMK-AMEX)#PAN#PAN Sequence Number#
Reserved#Partial AC#AC Padded Data Block#>
```

Response

```
<46A#Verification Flag#Session Key Check Digits#
IMK-AMEX Check Digits#>[CRLF]
```

Calling Parameters

`36A`

Field 0, the command identifier.

`Derivation Type`

Field 1, the AMEX derivation algorithm. This field must contain the number 3.

$E_{MFK.9}$(IMK-AMEX)

Field 2, the Issuer Master Key for AMEX encrypted under variant 9 of the MFK. The IMK-AMEX is a 2key-3DES (double-length) key. Option 6A determines if the 2key-3DES (double-length) key is allowed to have identical halves. If option 6A is disabled this command will require a true 2key-3DES (double-length) key that has different value for each half. If option 6A is enabled, no checks will be performed and a 1key-3DES (single-length) key can be replicated to make a 2key-3DES (double-length) key. This field contains a 32 byte hexadecimal value.

`PAN`

Field 3, the Primary Account Number. This field contains a 1-20 byte value. This field is also used to indicate the Master Key derivation method. If this field contains the letter "B" followed by 17 to 19 digits, method B will be used; otherwise method A will be used.

`PAN Sequence Number`

Field 4, the application PAN sequence number. This field contains a 2 byte decimal value. Applications that do not have a valid PAN sequence number should set this field to 00.

`Reserved`

Field 5, this field must be empty.

`Partial AC`

Field 6, the partial Application Cryptogram to be verified. It is formed by decimalizing the rightmost 3 bytes of the standard Application Cryptogram. This field contains a 5 byte decimal value.

`AC Padded Data Block`

Field 7, the data used to generate the Application Cryptogram version 02. The data block elements are the Unpredictable Number and the Application Transaction Counter (ATC). The data block is right pad with zeros such that the length of this field is a multiple of 16 characters. It is the host application's responsibility to collect all necessary data and format it for processing. The Network Security Processor does not pad the data.

Table 7-16. Command 36A: Verify AMEX Express pay value - Magstripe Mode

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	36A
1	Derivation Type	1	3
2	$E_{MFK.9}$(IMK-AMEX)	32	0 - 9, A - F
3	PAN	1-20	0 - 9, B
4	PAN Sequence Number	2	0 - 9
5	Reserved	0	none
6	Partial AC	5	0 - 9
7	AC Padded Data Block	16-1024	0 - 9, A - F

Responding Parameters

`46A`

Field 0, the response identifier.

`Verification Flag`

Field 1, the verification flag. This field returns Y if the partial AC is verified; otherwise, it returns N.

`Session Key Check Digits`

Field 2, the first four digits that result from encrypting zeros using the session key. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the session key.

`IMK-AMEX Check Digits`

Field 3, the first four digits that result from encrypting zeros using the Issuer Master Key-AMEX (IMK-AMEX). If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the IMK-AMEX.

.

Table 7-17. Response 46A: Verify AMEX Express pay value - Magstripe Mode

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	46A
1	Verification Flag	1	Y or N
2	Session Key Check Digits	4, 6	0 - 9, A - F
3	IMK-AMEX Check Digits	4, 6	0 - 9, A - F

Usage Notes

- Before using this command, generate the IMK-AMEX key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-6 for component values.

- Clear-text Issuer Master Key-AMEX (IMK-AMEX) is:
  BA20C2FB2A57EF9D F8D65B7623DA73C4, check digits = 925F
  IMK-AMEX encrypted under variant 9 of the MFK:
  18C4E73EE168921B15D479FB5B07C1ED
- PAN: 374245455400001
- PAN Sequence Number: 01
- Partial AC: 52195
- Unpredictable Number: 00004912
- Application Transaction Counter: 001803A00000
- AC Padded Data Block: 00004912001803A00000000000000000

The command looks like this:

```
<36A#3#18C4E73EE168921B15D479FB5B07C1ED#374245455400001#01##
52195#00004912001803A00000000000000000#>
```

The Network Security Processor returns the following response:

```
<46A#Y#8C6B#925F#>
```

8 Processing EMV and Visa Stored Value Cards

Europay, Mastercard, and Visa (EMV) have established a series of specifications for integrated circuit cards used in payment systems. These specifications are available at the following website: www.emvco.com. The Network Security Processor provides the ability to verify an Application Request Cryptogram (ARQC), and if successful return an Application Response Cryptogram (ARPC); generate a Message Authentication code; and generate the integrated circuit card master key.

The VISA Stored Value Card (VSVC) is VISA International's implementation of an electronic cash card application. This implementation uses a chip card to store cash value that can be spent with merchants who have the hardware to read and receive money from the chip card. The holder of a chip card can use an Automated Teller Machine (ATM) to reload cash into the card. The Network Security Processor is used in conjunction with an ATM to reload a VISA Stored Value card. The Network Security Processor does not support the initial personalization of the VSVC.

To skip this introduction, go to Table 8-2 for a list of commands.

EMV Master Key Derivation

Annex A of the EMV Integrated Circuit Card Specifications for Payment Systems, Book 2, Security and Key Management, Version 4.1, May 2004, documents an additional method "Option B" for generating the 16-byte ICC Master Key used for Application Cryptogram generation, issuer authentication, and secure messaging. Smartcards that are Common Core Definitions (CCD) compliant may require that option B be used to generate the ICC Master key.

VSVC Signatures

Three signatures, S1, S2, and S3, are used in the process of reloading a VSVC chip card. When a card reload transaction is requested at an ATM, an S1 Signature is generated by the chip card and sent through the ATM and host application to the Network Security Processor for verification. Upon verification of the S1 Signature, the Network Security Processor generates an S2 Signature which becomes part of the host authorization response to the transaction.

The S2 is sent through the ATM to the VSVC for verification. If the S2 is verified, the card adjusts its fund balance and calculates an S3 Signature to indicate completion of the transaction. The S3 Signature and related data are archived by the host and may be used in the future for non-repudiation. If a customer dispute occurs, the S3 Signature is used as proof of the transaction and may be sent to the Network Security Processor for verification.

VSVC signatures are generated using Data Encryption Standard (DES) encryption. For detailed descriptions of the VSVC signature generation algorithm, see DES Key Management for VSVC and VSVC Data Elements on page 8-3 below.

DES Key Management for VSVC

The DES key used to generate signatures is either a 1key-3DES (single-length) or 2key-3DES (double-length) VSVC Session Key. This key is calculated by encrypting the card-specific data, such as expiration-date and card transaction number, with a 1key-3DES (single-length) VSVC Diversified Key which is loaded in the card and is unique to each card.

The Diversified Key can be generated in the Network Security Processor by encrypting the bank-and-card-specific data, such as bank identification number and card serial number, using a VSVC Master Key. The VSVC Master Key is a 2key-3DES (double-length) DES key that is encrypted under variant 9 of the Network Security Processor Master File Key (MFK). The cryptogram of the VSVC Master Key is stored on the host. When a VSVC transaction is requested, the VSVC Master Key is sent to the Network Security Processor, along with other data that are required to generate the Diversified Key, Session Key, and the signatures. See VSVC Data Elements on page 8-3 for information about generating the Diversified Key and Session Key.

VSVC Data Elements

Table 8-1 lists the data elements and their token names used in Commands BE and BF.

Table 8-1. VSVC Data Elements

Data Element	Token	Type	Length (bytes)
VSVC Issuer BIN (Purse Provider ID)	PPiep	binary	3
Card Serial Number	IEPid	binary	5
Card Expiration Date	DEXPiep	binary	3
Transaction Number of IEP	NTiep	binary	2
Load Request Dollar Amount	Mlda	binary	4
Currency Code	CURRlda	binary	2
Currency Exponent	CEXPlda	binary	1
Balance of the IEP (chip card)	BALiep	binary	4
Acquirer BIN	PPSAMID	binary	4
ATM Date and Time	R	binary	4
Transaction Completion Code	CCiep	binary	2
Data used to generate S1 Signature	S1 Signature Data / S1 Data	binary	19
Data used to generate S2 Signature	S2 Signature Data / S2 Data	binary	7
Data used to generate S3 Signature	S3 Signature Data / S3 Data	binary	10
S1 Signature	S1 or S1 Signature	binary	8
S2 Signature	S2 or S2 Signature	binary	8
S3 Signature	S3 or S3 Signature	binary	8
Key Version	VKLiep	binary	1

Quick Reference

Table 8-2 identifies each command by number, name, and purpose.

Table 8-2. VSVC Signature and EMV Commands

Command #	Name	Purpose
BE	Verify S1 and Generate S2 Signatures	To verify a card reload request.
BF	Verify S3 Signature	To verify a S3 Signature.
350	Verify ARQC and ARPC	This command will verify an Application Request Cryptogram (ARQC), and if successful return an Application Response Cryptogram (ARPC), in accordance with Europay, MasterCard, and Visa standards.
351	EMV PIN Change	Facilitates the functions required when performing an EMV PIN Change with or without using the current PIN.
352	Generate EMV MAC	Generates a Message Authentication code in accordance with Europay, MasterCard, and Visa standards.
354	Generate ICC MK	Returns the ICC Master Key encrypted under the Key Exchange Key.
356	Validate CAP Token	Verifies an application cryptogram (AC) or signs transaction data.

Verify VSVC S1 Signature and Generate VSVC S2 Signature (Command BE)

Command BE is used to verify the S1 Signature and generate the S2 Signature. The S1 Signature was generated by the VSVC and sent to the ATM as a result of requesting a card reload transaction at an ATM.

This command is not enabled in the Network Security Processor's default factory security policy. You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

Command

```
<BE#E_MFK.9(VSVCMK)#PPiep#IEPid#DEXPiep#NTiep#S1 Data#
S1 Signature#S2 Data#>
```

Response

```
<CE#Verification Indicator#S2#Diversified Key Check Digits#
Session Key Check Digits#>
```

Calling Parameters

`BE`

Field 0, the command identifier.

$E_{MFK.9}$(VSVCMK)

Field 1, the VSVC Master Key encrypted under variant 9 of the MFK. This field contains a 32 byte hexadecimal value, or a volatile table location.

`PPiep`

Field 2, the Purse Provider Identifier. This field is used in the generation of the VSVC Diversified Key. This field contains a 3 byte binary value that has been converted to 6 ASCII hexadecimal characters.

`IEPid`

Field 3, the IEP (Intersector Electronic Purse) Identifier. This field is used in the generation of the VSVC Diversified Key. This field contains a 5 byte binary value that has been converted to 10 ASCII hexadecimal characters.

`DEXPiep`

Field 4, the expiration date of the IEP. This field is used in the generation of the VSVC Session Key. This field contains a 3 byte binary value that has been converted to 6 ASCII hexadecimal characters.

`NTiep`

Field 5, the transaction number of the IEP. This field is used in the generation of the VSVC Session Key. This field contains a 2 byte binary value that has been converted to 4 ASCII hexadecimal characters.

`S1 Data`

Field 6, the S1 Signature data. This field represents the six concatenated data elements, used to generate the S1 Signature. It contains 38 ASCII hexadecimal characters. The data elements are:

- Mlda, Load Request Dollar amount, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.
- CURRlda, Currency Code, a 2 byte binary value that has been converted to 4 ASCII hexadecimal characters.
- CEXPlda, Currency Exponent, a 1 byte binary value that has been converted to 2 ASCII hexadecimal characters.
- BALiep, Balance of the IEP, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.
- PPSAMID, Acquirer BIN, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.
- R, ATM Date and Time, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.

`S1`

Field 7, the S1 Signature. This value is compared with the S1 Signature that is generated by the Network Security Processor. This field contains a 16 byte hexadecimal value.

`S2 Data`

Field 8, the S2 Signature data. This field represents the concatenated data elements, used to generate the S2 Signature. It contains 14 ASCII hexadecimal characters. The data elements are:

- Mlda, Load Request Dollar amount, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.
- CURRlda, Currency Code, a 2 byte binary value that has been converted to 4 ASCII hexadecimal characters.

- CEXPlda, Currency Exponent, a 1 byte binary value that has been converted to 2 ASCII hexadecimal characters.

Table 8-3. Command BE: Verify VSVC S1 Signature

Field #	Contents	Length (bytes)	Legal Characters
0	Command Identifier	2	BE
1	$E_{MFK.9}$(VSVCMK)*	32	0-9, A-F
2	PPiep	6	0-9, A-F
3	IEPid	10	0-9, A-F
4	DEXPiep	6	0-9, A-F
5	NTiep	4	0-9, A-F
6	S1 Signature Data	38	0-9, A-F
7	S1 Signature	16	0-9, A-F
8	S2 Signature Data	14	0-9, A-F

* Can be a key table index.

Responding Parameters

`CE`

Field 0, the response identifier.

`Verification Indicator`

Field 1, the S1 Signature verification indicator. This field contains ‘Y’ if the S1 Signature is verified; otherwise ‘N’ is returned.

`S2 Signature`

Field 2, the S2 Signature. This field contains a 16 byte hexadecimal value. This field is empty if the S1 Signature is not verified.

`Diversified Key Check Digits`

Field 3, the Diversified Key check digits; the first four digits that result from encrypting zeros using the Diversified Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Session Key Check Digits`

Field 4, the Session Key check digits; the first four digits that result from encrypting zeros using the Session Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 8-4. Response CE: Verify VSVC S1 Signature

Field #	Contents	Length (bytes)	Legal Characters
0	Response Identifier	2	CE
1	Verification Indicator	1	Y or N
2	S2 Signature	0 or 16	0-9, A-F
3	Diversified Key Check Digits	4 or 6	0-9, A-F
4	Session Key Check Digits	4 or 6	0-9, A-F

Usage Notes

Before using Command BE generate the VSVC Master Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a S1 signature and generating a S2 signature.

- Clear-text VSVC Master Key: 7007 C1D5 EA19 0B98 BA75 E50B 89D0 2601. The VSVC Master Key encrypted under variant 9 of the MFK: C35C 04FD 418A 0829 1FF3 77A7 E440 41F6.
- PPiep: 451861.
- IEPid: 0000000011.
- DEXPied: 970731.
- NTiep: 000E.
- Mlda: 00000001.
- CURRlda: 0840.
- CEXPlda: 02.
- BALiep: 00002C0C.
- PPSAMID: 0000002E.
- R: 0000015A.
- S1 Signature: BD50 9E29 0EDC BCDA.

The command looks like this:

```
<BE#C35C04FD418A08291FF377A7E44041F6#451861#0000000011#
970731#000E#0000000108400200002C0C0000002E0000015A#
BD509E290EDCBCDA#00000001084002#>
```

The Network Security Processor returns the following response:

```
<CE#Y#0A4CA804206DD91C#2CF5#C78C#>
```

Verify VSVC S3 Signature (Command BF)

Command BF is used to verify the S3 Signature that is calculated by the VSVC after the S2 Signature is verified.

This command is not enabled in the Network Security Processor's default factory security policy. You must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

Command

<BF#$E_{MFK.9}$(VSVCMK)#PPiep#IEPid#DEXPiep#NTiep#S3 Data#S3#>

Response

<CF#Verification Indicator#Diversified Key Check Digits#
Session Key Check Digits#>

Calling Parameters

BF

Field 0, the command identifier.

$E_{MFK.9}$(VSVCMK)

Field 1, the VSVC Master Key encrypted under variant 9 of the MFK. This field contains a 32 byte hexadecimal value, or a volatile table location.

PPiep

Field 2, the Purse Provider Identifier. This field is used in the generation of the VSVC Diversified Key. This field contains a 3 byte binary value that has been converted to 6 ASCII hexadecimal characters.

IEPid

Field 3, the IEP (Intersector Electronic Purse) Identifier. This field is used in the generation of the VSVC Diversified Key. This field contains a 5 byte binary value that has been converted to 10 ASCII hexadecimal characters.

DEXPiep

Field 4, the expiration date of the IEP. This field is used in the generation of the VSVC Session Key. This field contains a 3 byte binary value that has been converted to 6 ASCII hexadecimal characters.

`NTiep`

Field 5, the transaction number of the IEP. This field is used in the generation of the VSVC Session Key. This field contains a 2 byte binary value that has been converted to 4 ASCII hexadecimal characters.

`S3 Signature Data`

Field 6, the S3 Signature data. This field represents the concatenated data elements, used to generate the S3 Signature. It contains 20 ASCII hexadecimal characters. The data elements are:

- PPSAMID, Acquirer BIN, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.
- R, ATM Date and Time, a 4 byte binary value that has been converted to 8 ASCII hexadecimal characters.
- CCiep, Transaction Completion Code, a 2 byte binary value that has been converted to 4 ASCII hexadecimal characters.

`S3 Signature`

Field 7, the S3 Signature. This field contains a 16 byte hexadecimal value.

Table 8-5. Command BF: Verify VSVC S3 Signature

Field #	Contents	Length (bytes)	Legal Characters
0	Command Identifier	2	BF
1	$E_{MFK.9}$(VSVCMK)*	32	0-9, A-F
2	PPiep	6	0-9, A-F
3	IEPid	10	0-9, A-F
4	DEXPiep	6	0-9, A-F
5	NTiep	4	0-9, A-F
6	S3 Signature Data	20	0-9, A-F
7	S3 Signature	16	0-9, A-F

* Can be a key table index.

Responding Parameters

`CF`

Field 0, the response identifier.

`Verification Indicator`

Field 1, the S3 Signature verification indicator. This field contains ‘Y’ if S3 Signature is verified; otherwise ‘N’ is returned.

`Diversified Key Check Digits`

Field 2, the Diversified Key check digits; the first four digits that result from encrypting zeros using the Diversified Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Session Key Check Digits`

Field 3, the Session Key check digits; the first four digits that result from encrypting zeros using the Session Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 8-6. Response CF: Verify VSVC S3 Signature

Field #	Contents	Length (bytes)	Legal Characters
0	Response Identifier	2	CF
1	Verification Indicator	1	Y or N
2	Diversified Key Check Digits	4 or 6	0-9, A-F
3	Session Key Check Digits	4 or 6	0-9, A-F

Usage Notes

Before using Command BF generate the VSVC Master Key.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Verifying a S3 signature.

- Clear-text VSVC Master Key: 7007 C1D5 EA19 0B98 BA75 E50B 89D0 2601. The VSVC Master Key encrypted under variant 9 of the MFK: C35C 04FD 418A 0829 1FF3 77A7 E440 41F6.
- PPiep: 4518 61.
- IEPid: 0000 0000 11.
- DEXPied: 9707 31.
- NTiep: 000E.
- PPSAMID: 0000 002E.
- R: 0000 015A.
- CCiep: 9000.
- S3 Signature: 1B48 ED0A F1BA 1A98.

The command looks like this:

```
<BF#C35C04FD418A08291FF377A7E44041F6#451861#0000000011#
970731#000E#0000002E0000015A9000#1B48ED0AF1BA1A98#>
```

The Network Security Processor returns the following response:

```
<CF#Y#2CF5#C78C#>
```

Verify EMV ARQC (Command 350)

Command 350 will generate an EMV Authorization Request Cryptogram (ARQC) and compare it with an ARQC that is supplied in the command. If they match, an Authorization Response Cryptogram (ARPC) will be returned.

This command requires a 2key-3DES (double-length) Issuer Master Key. If option 6A is enabled, this command will accept a replicated 1key-3DES (single-length) key. If option 6A is disabled, which is the default, this command requires a 2key-3DES (double-length) Issuer MasterKey.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<350#EMV Derivation Type#E_MFK.9(IMK)#Application PAN#
[Application PAN Sequence Number]#[Diversification Data]#
Authorization Request Cryptogram#Padded Data Block#
Authorization Response Code#[H#IV#Index#]
[Failure Response Code#]>
```

Response

```
<450#Authorization Response Cryptogram#
Session Key Check Digits#Issuer Master Key Check Digits#
[Verification Indictor#]>[CRLF]
```

Calling Parameters

`350`

Field 0, the command identifier.

`EMV Derivation Type`

Field 1, the derivation technique used to generate the ARQC and ARPC. This field contains a 1 byte decimal value defined as follows:

Standard Derivation	Value
Common Session (per EMV 4.1 and Specification Update Bulletin 46)	2

The following additional values are supported for inter operation with legacy applications.

Legacy Derivation	Value
Europay/Mastercard (ICC MK for ARPC calculation)	0
VISA (ICC MK for ARPC calculation)	1
VISA (Derived session key for ARPC calculation)	3
EMV2000-Tree (Derived session key for ARPC calculation)	8
EMV-Tree (ICC MK for ARPC calculation)	9

Derivation types 0, 1, and 9 use the derived session key to verify the ARQC and the ICC master key to generate the ARPC. Derivation types 2, 3 and 8 use the session key for both calculations.

$E_{MFK.9}$(IMK)

Field 2, the Issuer Master Key encrypted under variant 9 of the MFK. This field contains a 32 byte hexadecimal value.

Application PAN

Field 3, the application Primary Account Number. This field is also used to indicate the Master Key derivation method. If this field contains the letter “B” followed by 17 to 19 decimal digits, method B will be used, otherwise method A will be used.

[Application PAN Sequence Number]

Field 4, the optional application PAN sequence number. When present, this field contains a 2 hexadecimal character value. If not present, a PAN Sequence Number of 00 will be used.

[Diversification Data]

Field 5, the value of this field depends on the derivation type specified in field 1.

For the common session derivation algorithm (if the derivation type, Field 1, is 2) this field contains a 16 byte hexadecimal value consisting of the following two items:

- 2 byte Application Transaction Counter (ATC). This binary value is expressed as 4 hexadecimal characters.
- 6 byte fixed value. This binary value ‘000000000000’ is expressed as 12 hexadecimal characters.

For the EMV-Tree derivation algorithm (if the derivation type, Field 1, is 8 or 9) this field contains the four hexadecimal characters (2 bytes) of the Application Transaction Counter (ATC).

For the legacy Visa derivation algorithm (if the derivation type, Field 1, is 1 or 3) this field must be empty.

For the legacy Europay/Mastercard derivation algorithm (if the derivation type, Field 1, is 0) this field will contain either the same fields as the common session algorithm, or the four character ATC concatenated with 4 zero characters ‘0000’, followed by 4 bytes of hexadecimal characters (the unpredictable number).

`Authorization Request Cryptogram`

Field 6, the incoming Authorization Request Cryptogram (ARQC) to be validated. This field contains a 16 byte hexadecimal value.

`Padded Data Block`

Field 7, the padded data block. The length of this field is 16 to 1024 bytes.

For derivation types 0, 2, 8, and 9, the data should be right-padded with a one 80 byte (expressed as two hexadecimal characters ‘80’), followed by a variable number of binary zeros bytes (expressed as two hexadecimal characters ‘00’) to make the total data length a multiple of 8 bytes (16 hexadecimal characters). If the data length is a multiple of 8, the data is padded with a single byte 80 (expressed as two hexadecimal characters ‘80’) followed by 7 bytes of binary zeros (expressed as 00000000000000).

For example, assume 37 bytes of data (expressed as 74 hexadecimal characters).

```
0000000010000000000000000826000000800000
56000912002975E7015C00001600AB0975
```

The padding would contain 1 byte of hex 80 followed by 2 bytes of binary zero.

```
800000
```

The padded data block would be 40 bytes (expressed as 80 hexadecimal characters):

```
0000000010000000000000000826000000800000
56000912002975E7015C00001600AB0975800000
```

For derivation types 1 and 3 the data should be padded with a variable number of binary zeros bytes (expressed as 00 hex) to make the total data length a multiple of 8. If the data length is a multiple of 8, the data is padded with a 8 bytes of binary zeros (expressed as 0000000000000000).

For example, assume 30 bytes of data (expressed as 60 hexadecimal characters).

```
0000000010000000000000000826000000800000
56000912002975E7015C
```

The padded data block would be 32 bytes (expressed as 64 hexadecimal characters):

```
0000000010000000000000000826000000800000
56000912002975E7015C0000
```

The Network Security Processor does not enforce these data formats it only requires that the length of data is a multiple of 16 hexadecimal characters.

`Authorization Response Code`

Field 8, the Authorization Response Code (ARC) used to calculate the ARPC if the ARQC verified. See [Failure Response Code#] if the ARQC does not verify.

If this field contains a 2 byte (4 hexadecimal characters) value method 1 will be used to calculate the ARPC.

If method 2 should be used to calculate the ARPC this field must contain the 4 byte (8 hexadecimal characters) Card Status Update value. Proprietary Authentication Data is optional, if present, it must be concatenated to the right of the Card Status Update value. The maximum size of the Proprietary Authentication Data is 8 bytes (16 hexadecimal characters). The ARPC is the leftmost 4 bytes (8 hexadecimal characters) of the MAC (ISO/IEC 9797-1 Algorithm 3). The data used in the MAC calculation is as follows:

`ARQC||Card Status Update||Proprietary Authentication Data`

`[H#IV#Index#]`

These next three fields are used only if the derivation type is 8 or 9.

`[H#`

Field 9, the height value used for EMV-Tree derivation. This field is present only if the derivation type is 8 or 9. This field contains the value 8 or 16, or it can be empty. If this field is empty and the derivation type is 8 or 9, the height value of 8 will be used.

`IV#`

Field 10, the clear Initialization Vector used for EMV-Tree derivation. This field is present only if the derivation type is 8 or 9. This field contains a 32-byte hexadecimal value, or it can be empty. If this field is empty and the derivation type is 8 or 9, 32-bytes of 0 will be used.

`Index#]`

Field 11, the index value used for EMV-Tree derivation. The index specifies the byte location of the key that will be exclusive Or'd with the ATC coefficient. An index value of zero indicates the leftmost byte of the key will be exclusive Or'd with the ATC coefficient. An index value of 7 indicates the rightmost byte of the key will be exclusive Or'd with the ATC coefficient. If the key is double-length the index value is applied to both halves of the double length key.

This field is present only if the derivation type is 8 or 9. This field contains a 1 digit decimal value between 0-7, or it can be empty. If this field is empty and the derivation type is 8 or 9, the index value of 7 will be used.

`[Failure Response Code#]`

Field 12, the Failure Response Code (FRC) is used to calculate the ARPC if the ARQC verification fails. This field contains a 4 byte hexadecimal value. This is only permitted when the value in Field 1 is 2, 3, or 8.

Table 8-7. Command 350: Verify EMV ARQC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	350
1	EMV Derivation Type	1	0, 1, 2, 3, 8, 9
2	$E_{MFK.9}$(IMK)	32	0 - 9, A - F
3	Application PAN	1 - 20	0 - 9, B
4	[Application PAN Sequence Number]	0, 2	0 - 9, A - F
5	[Diversification Data]*	0, 16	0 - 9, A - F
6	Authorization Request Cryptogram	16	0 - 9, A - F
7	Padded Data Block**	16 - 1024	0 - 9, A - F
8	Authorization Response Code		
	ARPC Method 1	4	0 - 9, A - F
	ARPC Method 2	8 - 24	0 - 9, A - F
9	[H#	0-2	8, 16
10	IV#	0, 32	0 - 9, A - F
11	Index#]	0, 1	0 - 7
12	[Failure Response Code#]	4	0 - 9, A - F

*Empty if Field 1 contains a 1.

**Length must be a multiple of 16

Responding Parameters

`450`

Field 0, the response identifier.

`[Authorization Response Cryptogram]`

Field 1, the Authorization Response Cryptogram. The length of this field depends upon the ARPC method; for method 1 this field contains an 8 byte (16 hexadecimal character) value, for method 2 this field contains a 4 byte (8 hexadecimal character) value. This field is empty if ARQC did not verify and [Failure Response Code#] was not included in the command input. This field will not be empty when the Verification Indicator is present.

`Session Key Check Digits`

Field 2, the first four digits of the result from encrypting zeros using the generated Session Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Issuer Master Key Check Digits`

Field 3, the first four digits of the result from encrypting zeros using the Issuer Master Key. If option 88 is enabled, this field will contain the first six digits of the result from encrypting zeros using the Issuer Master Key.

`[Verification Indicator]`

Field 4, only signifies success or failure of the ARQC verification. This field will be present when the Failure Response Code (field 12 of the command input) is present. This field will be omitted when the [Failure Response Code#] is not present. When the Verification Indicator is present, the ARPC (Authorization Response Cryptogram) field will not be empty. This field contains a 1 byte character defined as follows:

- Y ARQC verification passed and ARPC is calculated using the ARC (Field 8)
- N ARQC verification failed and ARPC is calculated using the [Failure Response Code#]

Table 8-8. Response 450: Verify EMV ARQC

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	450
1	[Authorization Response Cryptogram]	0, 8, 16	0-9, A-F
2	Session Key Check Digits	4 or 6	0-9, A-F
3	Issuer Master Key Check Digits	4 or 6	0-9, A-F
4	[Verification Indicator]	1	Y, N

Usage Notes

- The Issuer Master Key must be encrypted under variant 9 of the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Europay/MasterCard ARQC Verification, ARPC Method 1, Option A Master Key Generation.

- Clear-text Issuer Master Key: 16AD 6E16 0226 2AF2 B383 8564 2C13 E66E. The Issuer Master Key encrypted under variant 9 of the MFK: 0620 AC0C 70AE EB93 39D7 8B69 4BCB DD5E.
- PAN: 5396 0300 0000 0669
- Sequence Number: 00
- ATC: 0016
- UN: 2975 E701
- ARQC: 92C7 D145 78E2 6E8D
- DATA: 0000000010000000000000000826000000800000
- 56000912002975E7015C00001600AB0975000000
- ARC: 0000

The command looks like this:

```
<350#0#0620AC0C70AEEB9339D78B694BCBDD5E#5396030000000669#00#
001600002975E701#92C7D14578E26E8D#0000000010000000000000000008
260000008000005600091200 2975E7015C00001600AB0975800000#0000#>
```

The Network Security Processor returns the following response:

```
<450#6936F437C5BB00BC#B952#C697#>
```

Visa ARQC Verification, ARPC Method 1, Option A Master Key Generation.

- Clear-text Issuer Master Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210. The Issuer Master Key encrypted under variant 9 of the MFK: 94E1 BA82 35D3 8B08 9AC5 BBD4 F34C 67E8.
- PAN: 3110 4999 9100 34
- Sequence Number: 01
- ARQC: 30E4 D3FC CC38 A565
- DATA: 0000000111070000000000000826000000000008260006300010422741 4C00004903A00000000000
- ARC: 0000

The command looks like this:

```
<350#1#94E1BA8235D38B089AC5BBD4F34C67E8#31104999910034#01##
30E4D3FCCC38A565#0000000111070000000000000826000000000008 2
60006300010422741 4C00004903A00000000000#0000#>
```

The Network Security Processor returns the following response:

```
<450#124D4BF4AC90D06D#4C63#08D7#>
```

EMV Tree Derivation ARQC Verification using MK, ARPC Method 1, Option A Master Key Generation.

- Clear-text Issuer Master Key: `589CA02B6BAC5BDD 97238A7EDAF71298`
  The Issuer Master Key encrypted under variant 9 of the MFK:
  E2214AE745E7077F 98C9B405B102F9BB
- PAN = `9901234567890123`
- Sequence Number = `45`
- ATC = `293A`
- ARQC = `4F5413D5EAB69B18`
- Data = `0123456789ABCDEF0123456789ABCDEF`
- ARC = `EF12`
- Height = `8`
- IV = `Null`
- Index = `7`

The command looks like this:

```
<350#9#E2214AE745E7077F98C9B405B102F9BB#9901234567890123#45#2
93A#4F5413D5EAB69B18#0123456789ABCDEF0123456789ABCDEF#EF12#8#
#7#>
```

The Network Security Processor returns the following response:

```
<450#74C88CDE14FFF289#7FA0#FDD1#>
```

EMV Tree Derivation ARQC Verification using derived Session Key ARPC Method 1, Option A Master Key Generation.

- Clear-text Issuer Master Key: 589CA02B6BAC5BDD 97238A7EDAF71298
  The Issuer Master Key encrypted under variant 9 of the MFK:
  E2214AE745E7077F 98C9B405B102F9BB
- PAN = `9901234567890123`
- Sequence Number = `45`
- ATC = `293A`
- ARQC = 4F5413D5EAB69B18 (match)
- Data = `0123456789ABCDEF0123456789ABCDEF`
- ARC = `EF12`

- Height = 8
- IV = Null
- Index = 7

The command looks like this:

```
<350#8#E2214AE745E7077F98C9B405B102F9BB#9901234567890123#45#2
93A#4F5413D5EAB69B18#0123456789ABCDEF0123456789ABCDEF#EF12#8#
#7#>
```

The Network Security Processor returns the following response:

```
<450#BA6EC017FBE0AF8D#7FA0#FDD1#>
```

EMV Tree Derivation ARQC Verification using Derived Session Key, Option B Master Key Derivation, ARPC Method 2

- Clear-text Issuer Master Key: 589CA02B6BAC5BDD 97238A7EDAF71298
  The Issuer Master Key encrypted under variant 9 of the MFK:
  E2214AE745E7077F 98C9B405B102F9BB
- PAN = B990123456789012300
- Sequence Number = 45
- ATC = 293A
- ARQC = EBAC702CAF7E57EF
- Data = 0123456789ABCDEF0123456789ABCDEF
- Card Status Update = EF123456
- Proprietary Authentication Data = ABCDEF123456ABCD
- Height = 8
- IV = Null
- Index = 7

The command looks like this:

```
<350#8#E2214AE745E7077F98C9B405B102F9BB#B990123456789012300#4
5#293A#EBAC702CAF7E57EF#0123456789ABCDEF0123456789ABCDEF#EF12
3456ABCDEF123456ABCD#8##7#>
```

The Network Security Processor returns the following response:

```
<450#DC1DC779#11EA#FDD1#>
```

EMV PIN Change (Command 351)

Command 351 – Facilitates the functions required when performing EMV PIN Change with or without using the current (old) PIN.

This command supports two EMV-specific types of PIN blocks. These PIN blocks are constructed like an ANSI PIN block. However, instead of XORing with the account number, these blocks XOR with the derived AC ICC MK. The EMV VISA8 block encrypts the resulting block directly. The EMV VISA PIN block prepends the length of the PIN block, and then pads the result. Thus, the EMV VISA PIN block will be 32 characters long.

This command requires a 2key-3DES (double-length) Issuer Master Key. If option 6A is enabled, this command will accept a replicated 1key-3DES (single-length) key. If option 6A is disabled, which is the default, this command requires a 2key-3DES (double-length) Issuer MasterKey.

To enable this command you must purchase this command in the form of a command 105, and enable it in the Network Security Processor's security policy.

Command

<351#Derivation Type#Incoming PIN Block type#KPE Variant#
$E_{MFK.V}$(KPE)#$E_{MFK.14}$(IMK_{ENC})#$E_{MFK.13}$(IMK_{MAC})#[$E_{MFK.9}$(IMK_{AC})]#
E_{KPE}(new PIN Block)#[PIN Issue Number]#Application PAN#
PAN Sequence Number#Diversification Data#Application data#
[PIN Block Data]#[E_{KPE}(old PIN Block)]#[H#IV#Index#]>

Response

<451#Sanity Check#[Encrypted PIN block]#[MAC]#KPE Check
Digits# IMK_{ENC} Check Digits#IMK_{MAC} Check Digits#[IMK_{AC} Check
Digits] #[SK_{ENC} Check Digits]#[SK_{MAC} Check Digits]#>

Calling Parameters

351

Field 0, the command identifier.

`Derivation Type`

Field 1, the derivation type. This field contains a 1 byte decimal value that describes both the session key derivation method and the type of outgoing PIN block to generate. Valid values are defined as follows:

Derivation Type	Numerical Code
Common Session (EMV Version 4.1 and Specification Update Bulletin 46) derivation with ISO format 2 PIN block.	0
Legacy VISA derivation technique with VISA PIN block	1
Legacy VISA derivation technique with VISA8 PIN block	2
EMV2000 (Tree-based technique) with ISO format 2 PIN block	3
EMV2000 (Tree-based technique) with VISA PIN block	4
EMV2000 (Tree-based technique) with VISA8 PIN block	5

`Incoming PIN Block type`

Field 2, specifies the incoming PIN block type. This field is 1 byte, and can contain the values 0, 1, or L. The following table identifies the numerical code for each PIN block type.

PIN Block Type	Numerical Code
ISO format 1	0
ANSI	1
Lloyds	L

`KPE Variant`

Field 3, the variant of the KPE - must be 1 or 14. The variant must be 14 if Field 2 is 'L'.

$E_{MFK.V}$(KPE)

Field 4, the PIN Encryption Key encrypted under the MFK. KPE can be 1key-3DES (single-length) or 2key-3DES (double-length). This field contains a 16 or 32 byte value.

$E_{MFK.14}(IMK_{ENC})$

Field 5, the Issuer Master Key encrypted under the MFK. IMK_{ENC} must be 2key-3DES (double-length). This field contains a 32 byte value.

$E_{MFK.13}(IMK_{MAC})$

Field 6, the MAC of the Issuer Master Key encrypted under the MFK. IMK_{MAC} must be 2key-3DES (double-length). This field contains a 32 byte value.

$[E_{MFK.9}(IMK_{AC})]$

Field 7, the Issuer Master Key encrypted under the MFK. This field is optional. This field is empty if field 1 is 0. IMK_{AC} must be 2key-3DES (double-length), if present. This field contains a 32 byte value.

E_{KPE}(new PIN Block)

Field 8, the encrypted PIN block. An error is returned if this PIN block fails the sanity check. This field contains a 16 byte hexadecimal value.

[PIN Issue Number]

Field 9, This field is optional. This field is empty if field 2 is not 'L'.

Application PAN

Field 10, the Primary Account Number for the application. This field is also used to indicate the Master Key derivation method. If this field contains the letter "B" followed by 17 to 19 decimal digits, method B will be used, otherwise method A will be used.

PAN Sequence Number

Field 11, the Primary Account Number sequence number. This field contains a 2 digit decimal value.

Diversification Data

Field 12, the value of this field depends on the derivation type specified in field 1.

For the common session derivation algorithm (if the derivation type, Field 1, is 0) this field contains a 16 byte hexadecimal value as defined in EMV SU-46.

For all other derivation types, this field contains the four hexadecimal characters (2 bytes) of the Application Transaction Counter.

[Application Data]

Field 13, the APP Data field may contain the 5-byte EMV command message header (CLA, INS, P1, P2, and Lc) followed by other optional items such as the Application Transaction Counter (ATC), or the Application Cryptogram (ARQC). If the optional ATC and ARQC are included in the calculation of MAC, it is the application's responsibility to pre-attach them in the Application data that is provided in the command. The Application data will be concatenated with the encrypted PIN block (i.e., Application data || Encrypted PIN block) to form a script message to calculate the MAC. The content of Lc byte is not validated or manipulated by the Network Security Processor but it is important that it contains the appropriate value per EMV specification. It must be an even number of ascii-hexadecimal characters. The maximum amount of data in this field is 3600 bytes.

`[PIN Block Data]`

Field 14, PIN block data. This field is optional. Its contents depend on the PIN block type used. See PIN Block Types on page 4-4. Empty if Field 2 is ‘0’ or '`L`'; 12-digit PAN for ANSI PIN Block.

`[`E_{KPE}`(old PIN Block)]`

Field 15, the encrypted PIN block for the old PIN. This field is optional. This field is provided only if the old PIN is needed to XOR with the new PIN. This field should be empty if the Derivation Type is not Visa or if there is no old PIN. If used, this field contains a 16 byte hexadecimal value.

`[H#`

Field 16, the height value used for EMV-Tree derivation. This field is present only if the EMV-Tree derivation type is used (field 1 contains a value of 3-5). This field contains the value 8 or 16, or it can be empty. If this field is empty and field 1 contains a value of 3-5 the height value of 8 will be used.

`IV#`

Field 17, the clear Initialization Vector used for EMV-Tree derivation. This field is present only if the EMV-Tree derivation type is used (field 1 contains a value of 3-5)This field contains a 32-byte hexadecimal value, or it can be empty. If this field is empty and field 1 contains a value of 3-5 an IV of 32-bytes of 0 will be used.

`Index#]`

Field 18, the index value used for EMV-Tree derivation. The index specifies the byte location of the key that will be exclusive Or'd with the ATC coefficient. An index value of zero indicates the leftmost byte of the key will be exclusive Or'd with the ATC coefficient. An index value of 7 indicates the rightmost byte of the key will be exclusive Or'd with the ATC coefficient. If the key is double-length the index value is applied to both halves of the double length key.

This field is present only if the EMV-Tree derivation type is used (field 1 contains a value of 3-5). This field contains a 1 digit decimal value between 0-7, or it can be empty. If this field is empty and field 1 contains a value of 3-5 the index value of 7 will be used.

Table 8-9. Command 351: PIN Change – EMV (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	351
1	Derivation type	1	0, 1, 2, 3, 4, or 5
2	Incoming PIN block type	1	0, 1, or L
3	KPE Variant	1 - 2	1, 14

Table 8-9. Command 351: PIN Change – EMV (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	$E_{MFK.V}$(KPE)*	16 or 32	0 - 9, A - F
5	$E_{MFK.14}$(IMK_{ENC})*	32	0 - 9, A - F
6	$E_{MFK.13}$(IMK_{MAC})*	32	0 - 9, A - F
7	[$E_{MFK.9}$(IMK_{AC})*]	0 or 32	Empty; or 0 - 9, A - F
8	E_{KPE}(new PIN Block)	16	0 - 9, A - F
9	[PIN issue number]	0 or 3	Empty, or 000-255
10	Application PAN	13 - 19	0 - 9, B
11	PAN sequence number	2	0 - 9
12	Diversification Data	4 or 16	0 - 9, A - F
13	[Application Data]	0 - 3600	Empty; or 0 - 9, A - F
14	[PIN block data]	0 or 12	Empty; or 0 - 9
15	[E_{KPE}(old PIN Block)]	0 or 16	Empty; or 0 - 9, A - F
16	[H#	0-2	8, 16
17	IV#	0, 32	0 - 9, A - F
18	Index#]	0, 1	0 - 7

* Can be a volatile table location.

Responding Parameters

`451`

Field 0, the response identifier.

`Sanity Check`

Field 1, the sanity check status. This field will contain either:

Y = OLD PIN verified successfully.
LR = Indicated PIN length is less than the minimum PIN length
I = Incorrect rightmost padding characters
SR = Incorrect control field, indicated PIN length is greater than 12, or non-numeric PIN digits

`[Encrypted PIN block]`

Field 2, the PIN block encrypted by SK_{ENC}. This field is 0, 16, or 32 bytes. The PIN Block format will default to Format-2 (ANSI) when the Derivation Type is 0 or 3 (Europay/MasterCard), and Format-0 when Derivation Type is 1or 4 (Visa), or 2 or 5 (VISA8).

[MAC]

Field 3, the MAC of the issuer script message. This field is 0 or 16 bytes.

KPE Check Digits

Field 4, the check digits of the key for PIN encryption. This field is 6 bytes.

IMK_{ENC} Check Digits

Field 5, the check digits of the Issuer Master Key for message confidentiality (IMK_{ENC}). This field is 6 bytes.

IMK_{MAC} Check Digits

Field 6, the check digits of the Issuer Master Key for message integrity (IMK_{MAC}). This field is 6 bytes.

[IMK_{AC} Check Digits]

Field 7, the check digits of the Issuer Master Key for Application Cryptogram (IMK_{AC}). This field is empty if field 7 in the command is empty, otherwise this field is 6 bytes.

3F.5h3 Tc(AC)Tj Master Ke 124.8 498 Tm .0081 Tc(Check Digits])Tj/TT4 1 Tf

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Europay/Mastercard, Master Key Derivation Option A

- cleartext KPE: 0123456789ABCDEF FEDCBA9876543210
  The KPE encrypted under variant 1 of the MFK:
  AE86D417E64E07E0 BC62A2AD72516EA1
- cleartext IMK for confidentiality: 1234 1234 5678 5678 8765 8765 4321 4321
  The IMK for confidentiality encrypted under variant 14 of the MFK:
  883B5A5B5A040688 B439ECB1F37595AA

  cleartext IMK for integrity: ABCD ABCD EF01 EF01 10FE 10FE DCBA DCBA
  The IMK for integrity encrypted under variant 13 of the MFK:
  02C6A6A79BC70719 AC4D013F8E566492

Data

- PIN = 654321
- Application PAN = 5555557890123456
- PAN Sequence Number = 73
- Unpredictable Number = 5093000087654321
- APP Data = 8424000210
- ISO Format 1 PIN Block = 16654321FFFFFFFF
- EKPEI(ISO Format 1 PIN Block) = 30E96734FD6501AB(incoming encrypted PIN block)

The command looks like this:

```
<351#0#0#1#AE86D417E64E07E0BC62A2AD72516EA1#883B5A5B5A040688B
439ECB1F37595AA#02C6A6A79BC70719AC4D013F8E566492##30E96734FD6
501AB##5555557890123456#73#5093000087654321#8424000210##>
```

The Network Security Processor response is:

```
<451#Y#B5660CC137F464AF#ED8A944FA0DC75F0#08D7B4#61DEBE#718B4C
##7356D5#34EA9F#>
```

Europay/Mastercard Tree Derivation, Master Key Derivation Option B

- ISO format 1 PIN block
- cleartext KPE: 0123456789ABCDEF FEDCBA9876543210
  The KPE encrypted under variant 1 of the MFK:
  AE86D417E64E07E0 BC62A2AD72516EA1
- cleartext IMK for confidentiality: 1234 1234 5678 5678 8765 8765 4321 4321
  The IMK for confidentiality encrypted under variant 14 of the MFK:
  883B5A5B5A040688 B439ECB1F37595AA
- cleartext IMK for integrity: ABCD ABCD EF01 EF01 10FE 10FE DCBA DCBA
  The IMK for integrity encrypted under variant 13 of the MFK:
  02C6A6A79BC70719 AC4D013F8E566492
- The encrypted new PIN block: 30E96734FD6501AB
- Application PAN: B55555578901234567
- Sequence Number: 73
- ATC: FFFF
- Application Data: 8424000210
- Height: 8
- IV: all zeros
- Index: 7

The command looks like this:

```
<351#3#0#1#AE86D417E64E07E0BC62A2AD72516EA1#883B5A5B5A040688B
439ECB1F37595AA#02C6A6A79BC70719AC4D013F8E566492##30E96734FD6
501AB##B555555789012345 67#73#FFFF#8424000210###8##7#>
```

The Network Security Processor returns the following response:

```
<451#Y#2B8C8BDC3DCF181E#47753CEAC2E81620#08D7B4#61DEBE#718B4C
##561A31#19CBD0#>
```

Generate EMV MAC (Command 352)

Command 352 generates an EMV MAC.

This command requires a 2key-3DES (double-length) Issuer Master Key. If option 6A is enabled, this command will accept a replicated 1key-3DES (single-length) key. If option 6A is disabled, which is the default, this command requires a 2key-3DES (double-length) Issuer MasterKey.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<352#EMV Derivation Type#E_MFK.13(Issuer Master Key)#
[Application PAN]#Application PAN Sequence Number#
Diversification Data#MAC Length#
[E_MFK.6(Continuation-IV)]#Padded Data#
[H#IV#Index#]>
```

Response

```
<452#MAC Length#MAC or E_MFK.6(Continuation-IV)#
KMAC Check Digits#Issuer Master Key Check Digits#>[CRLF]
```

Calling Parameters

`352`

Field 0, the command identifier.

`EMV Derivation Type`

Field 1, the derivation type. This field contains a 1 byte decimal value defined as follows:

Derivation Type	Numerical Code
Common Session (per EMV 4.1 and Specification Update Bulletin 46)	0
Legacy VISA technique	1
EMV2000 (Tree-based technique)	9

$E_{MFK.13}$(IMK)

Field 2, the Issuer Master Key encrypted under variant 13 of the MFK. This field contains a 32 byte hexadecimal value.

`Application PAN`

Field 3, the Application Primary Account Number. This field is also used to indicate the Master Key derivation method. If this field contains the letter “B” followed by 17 to 19 decimal digits, method B will be used, otherwise method A will be used.

`Application PAN Sequence Number`

Field 4, the sequence number. This field contains a 2 digit decimal value.

`Diversification Data`

Field 5, the value of this field depends on the derivation type specified in field 1.

For the common session derivation algorithm (if the derivation type, Field 1, is 0) this field contains a 16 byte hexadecimal value as defined in EMV SU-46.

For all other derivation types, this field contains the four hexadecimal characters (2 bytes) of the Application Transaction Counter.

`MAC Length`

Field 6, the length of the MAC.

The following table indicates the possible MAC sizes and the codes to enter in this field.

MAC Size	Code
More data expected; no MAC verified	0
32 bits	1
48 bits	2
64 bits	3

A 32 bit MAC is expressed as eight hexadecimal digits (0-9, A-F) and written as two groups of four digits, separated by a space. A 48 bit or 64 bit MAC is expressed as three or four groups of four hexadecimal digits, separated by a space.

This field can contain a 1 byte decimal value.

`[`$E_{MFK.6}$`(Continuation-IV)]`

Field 7, contains the continuation-IV, only if the MAC calculation is continued from a previous command. It must not be present in the first command of a multiple command sequence. This field contains either a 16 byte hexadecimal value, or is empty.

`Padded Data`

Field 8, is the data used to calculate the MAC. Per the EMV specification, the data should be right-padded with a single byte (expressed as two hexadecimal characters “80”), followed by a variable number of binary zeros bytes (expressed as two hexadecimal characters “00”) to make the total data length a multiple of 8

bytes (16 hexadecimal characters). If the data length is a multiple of 8 bytes (expressed as 16 hexadecimal characters), the data is padded with a single byte (expressed as two hexadecimal characters “80”) followed by 7 bytes of binary zeros (expressed as 00000000000000).

For example, assume 37 bytes of data (expressed as 74 hexadecimal characters).
```
00000000100000000000000000826000000800000
56000912002975E7015C00001600AB0975
```

The padding would contain 1 byte of hex 80 followed by 2 bytes of binary zero.
```
800000
```

The padded data block would be 40 bytes (expressed as 80 hexadecimal characters):
```
00000000100000000000000000826000000800000
56000912002975E7015C00001600AB0975800000
```

The Network Security Processor does not enforce this data format it only requires that the length of data is a multiple of 16 hexadecimal bytes.

`[H#`

Field 9, the height value used for EMV-Tree derivation. This field is present only if the EMV-Tree derivation type is used (field 1 contains a value of 9). This field contains the value 8 or 16, or it can be empty. If this field is empty and field 1 contains a value of 9 the height value of 8 will be used.

`IV#`

Field 10, the clear Initialization Vector used for EMV-Tree derivation. This field is present only if the EMV-Tree derivation type is used (field 1 contains a value of 9).This field contains a 32-byte hexadecimal value, or it can be empty. If this field is empty and field 1 contains a value of 9 an IV of 32-bytes of 0 will be used.

`Index#]`

Field 11, the index value used for EMV-Tree derivation. The index specifies the byte location of the key that will be exclusive Or'd with the ATC coefficient. An index value of zero indicates the leftmost byte of the key will be exclusive Or'd with the ATC coefficient. An index value of 7 indicates the rightmost byte of the key will be exclusive Or'd with the ATC coefficient. If the key is double-length the index value is applied to both halves of the double length key.

This field is present only if the EMV-Tree derivation type is used (field 1 contains a value of 9). This field contains a 1 digit decimal value between 0-7, or it can be empty. If this field is empty and field 1 contains a value of 9 the index value of 7 will be used.

Table 8-11. Command 352: Generate EMV MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	352
1	EMV Derivation Type	1	0, 1, or 9
2	$E_{MFK.13}$(IMK)	32	0 - 9, A - F
3	Application PAN	1-19	0 - 9, B
4	Application PAN Sequence Number	2	0 - 9, A - F
5	Diversification Data	4, 16	0 - 9, A - F
6	MAC Length	1	0 - 3
7	[$E_{MFK.6}$(Continuation-IV)]*	0, 16	0 - 9, A - F
8	Padded Data**	16-4096	0 - 9, A - F
9	[H#	0-2	8, 16
10	IV#	0, 32	0 - 9, A - F
11	Index#]	0, 1	0 - 7

*Contains data only if the MAC calculation is continued from a previous command. It is empty in the first command of a multiple command sequence.

**Length must be a multiple of 16

Responding Parameters

`452`

Field 0, the response identifier.

`MAC Length`

Field 1, the length of the MAC. This field contains the value of field 6 in the command.

`MAC` **`or`** $E_{MFK.6}$`(Continuation-IV)`

Field 2, if Field 1 of the response is set to 0, this field will contain the Continuation-Initialization Vector encrypted under variant 6 of the MFK. If Field 1 of the response is not 0, this field will contain MAC verification flag.

If your use of this command results in the generation of an Continuation- IV in this field, input this value in subsequent MAC commands used to continue generating the MAC.

If using this command results in a MAC verification flag, then this field will return Y if the MAC is verified, or N if the MAC is not verified.

This field contains a 16 byte hexadecimal value, or a one byte value, either “Y” or “N”.

`KMAC Check Digits`

Field 3, the first four digits of the result from encrypting zeros using the derived Message Authentication Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Issuer Master Key Check Digits`

Field 4, the first four digits of the result from encrypting zeros using the Issuer Master Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 8-12. Response 452: Generate EMV MAC

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	452
1	MAC Length	1	0-3
2	MAC	9, 14, 19	0-9, A-F; Space
	or		
	$E_{MFK.6}$(Continued-IV)	16	0-9, A-F
3	KMAC Check Digits	4 or 6	0-9, A-F
4	Issuer Master Key Check Digits	4 or 6	0-9, A-F

Usage Notes

- The Issuer Master Key must be encrypted under variant 13 of the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Visa MAC Generation

- Clear-text Issuer Master Key: 160E 5EA2 D670 8083 DA13 1332 7051 62DF. The Issuer Master Key encrypted under variant 13 of the MFK: FCE6 FC9C 73B1 A34A FB22 4B43 13A8 15F2.
- PAN: 4921 8292 6875 1914
- Sequence Number: 01
- ATC: 0007
- MAC Type: 3

- DATA: 841600000800077519ED6C6606E8E180

The command looks like this:

```
<352#1#FCE6FC9C73B1A34AFB224B4313A815F2#49218292687519 14#01#
0007#3##841600000800077519ED6C6606E8E180#>
```

The Network Security Processor response is:

```
<452#3#D220 2504 3A29 CA00#BA13#5128#>
```

EMV-Tree Example

- Clear-text Issuer Master Key: 589C A02B 6BAC 5BDD 9723 8A7E DAF7 1298
  The Issuer Master Key encrypted under variant 13 of the MFK: 6641 CE3D D053 FBA4 5C45 A570 53AC 533E.
- Application PAN: 9901234567890123
- Sequence Number: 45
- ATC: 293A
- MAC Length: 3
- Data: `0123456789ABCDEF0123456789ABCDEF`
- Height: 8
- IV: all zeros
- Index: 7

The command looks like this:

```
<352#9#6641CE3DD053FBA45C45A57053AC533E#9901234567890123#45#2
93A#3##0123456789ABCDEF0123456789ABCDEF#8##7#>
```

The Network Security Processor returns the following response:

```
<452#3#4F54 13D5 EAB6 9B18#7FA0#FDD1#>
```

Europay/MasterCard MAC Generation, Master Key Generation Option B

- Clear-text Issuer Master Key: 160E 5EA2 D670 8083 DA13 1332 7051 62DF.
  The Issuer Master Key encrypted under variant 13 of the MFK: FCE6 FC9C 73B1 A34A FB22 4B43 13A8 15F2.
- PAN: B49215678901234567
- Sequence Number: 01
- Random Number: 7F3D0000275A210B
- MAC Type: 3
- DATA: 841600000800077519ED6C6606E8E180

The command looks like this:

```
<352#0#FCE6FC9C73B1A34AFB224B4313A815F2#B4921567890123456 7#01
#7F3D0000275A210B#3##8416000008000775 19ED6C6606E8E180#>
```

The Network Security Processor response is:

```
<452#3#B99E 3F64 6449 6CF0#E25A#5128#>
```

Generate EMV ICC Master Key (Command 354)

This command generates the Integrated Circuit Card Master Key and returns it encrypted under a Key Exchange Key.

This command requires a 2key-3DES (double-length) Issuer Master Key. If option 6A is enabled, this command will accept a replicated 1key-3DES (single-length) key. If option 6A is disabled, which is the default, this command requires a 2key-3DES (double-length) Issuer MasterKey.

This command is enabled in the Network Security Processor's default security policy.

Command

<354#$E_{MFK.9}$(IMK)#Application PAN#
Application PAN Sequence Number#$E_{MFK.31}$(KEK)#>

Response

<454#E_{KEK}(ICC Master Key)#ICC Master Key Check Digits#
Issuer Master Key Check Digits#
Key Exchange Key Check Digits#>[CRLF]

Calling Parameters

354

Field 0, the command identifier.

$E_{MFK.9}$(IMK)

Field 1, the Issuer Master Key encrypted under variant 9 of the MFK. This field contains a 32 byte hexadecimal value.

Application PAN

Field 2, the Application Primary Account Number. This field is also used to indicate the Master Key derivation method. If this field contains the letter "B" followed by 17 to 19 decimal digits, method B will be used, otherwise method A will be used.

Application PAN Sequence Number

Field 3, the sequence number. This field contains a 2 digit decimal value.

$E_{MFK.31}$(KEK)

Field 4, is the Key Exchange Key encrypted under variant 31 of the MFK. This field contains a 32 byte hexadecimal value.

Table 8-13. Command 354: Generate ICC Master Key

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	354
1	$E_{MFK.9}$(IMK)	32	0 - 9, A - F
2	Application PAN	1-19	0 - 9, B
3	Application PAN Sequence Number	2	0 - 9, A - F
4	$E_{MFK.31}$(KEK)	32	0 - 9, A - F

Responding Parameters

`454`

Field 0, the response identifier.

E_{KEK}`(ICC Master Key)`

Field 1, the length of the MAC. This field contains the value of field 6 in the command.

`ICC Master Key Check Digits`

Field 2, the first four digits of the result from encrypting zeros using the derived ICC Master Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Issuer Master Key Check Digits`

Field 3, the first four digits of the result from encrypting zeros using the Issuer Master Key. If option 88 is enabled, this field will contain the first six digits of the result.

`Key Exchange Key Check Digits`

Field 4, the first four digits of the result from encrypting zeros using the Key Exchange Key. If option 88 is enabled, this field will contain the first six digits of the result.

Table 8-14. Response 454: Generate ICC Master Key (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	454
1	E_{KEK}(ICC Master Key)	32	0 -9, A - F

Table 8-14. Response 454: Generate ICC Master Key (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
2	ICC Master Key Check Digits	4 or 6	0-9, A-F
3	Issuer Master Key Check Digits	4 or 6	0-9, A-F
4	Key Exchange Key Check Digits	4 or 6	0-9, A-F

Usage Notes

- The Issuer Master Key must be encrypted under variant 9 of the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Option A Master Key Generation

- Clear-text Issuer Master Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210. The Issuer Master Key encrypted under variant 9 of the MFK: 94E1BA8235D38B089AC5BBD4F34C67E8.
- PAN: 31104999910034
- Sequence Number: 01
- Clear-text Key Exchange Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210. The Key Exchange Key encrypted under variant 31 of the MFK: 49E612E060F2DC1765D7BD60335B95B5.

The command looks like this:

```
<354#94E1BA8235D38B089AC5BBD4F34C67E8#31104999910034#01#
49E612E060F2DC1765D7BD60335B95B5#>
```

The Network Security Processor response is:

```
<454#18C70B43939B5C0C1EEFEF782AB4397B#4C63#08D7#08D7#>
```

Option B Master Key Generation

- Clear-text Issuer Master Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210. The Issuer Master Key encrypted under variant 9 of the MFK: 94E1BA8235D38B089AC5BBD4F34C67E8.
- PAN: B31104999910034567
- Sequence Number: 01
- Clear-text Key Exchange Key: 0123 4567 89AB CDEF FEDC BA98 7654 3210. The Key Exchange Key encrypted under variant 31 of the MFK: 49E612E060F2DC1765D7BD60335B95B5.

The command looks like this:

```
<354#94E1BA8235D38B089AC5BBD4F34C67E8#31104999910034567#01#
49E612E060F2DC1765D7BD60335B95B5#>
```

The Network Security Processor response is:

```
<454#3DC183ECB9D12F7E11B26480E7735700#7F15C0#08D7#08D7#>
```

Validate CAP Token (Command 356)

Command 356 supports both partial application cryptogram (AC) validation and transaction data signing (TDS). For partial AC validation, the Network Security Processor generates an EMV application cryptogram, selects a subset of the bits according to a supplied Issuer Proprietary Bitmap (IPB), and compares the selected bits to the partial AC. If transaction data signing is selected instead, the Network Security Processor generates the EMV AC, and then uses the AC as a key to single-DES CBC MAC the transaction data. The Network Security Processor then selects a subset of the bits from the MAC result according to the IPB and compares the result to the input partial MAC.

This command requires a 2key-3DES (double-length) Issuer Master Key. If option 6A is enabled, this command will accept a replicated 1key-3DES (single-length) key. If option 6A is disabled, which is the default, this command requires a 2key-3DES (double-length) Issuer MasterKey.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<356#EMV Derivation Type#E_MFK.9(IMK)#Application PAN#
[PAN Sequence Number]#[Diversification Data]#
Partial AC or MAC#AC Padded Data Block#[Partial IPB]#
[H#IV#Index#][TDS Data Block#]>
```

Response

```
<456#Verification Indicator#Session Key Check Digits#
Issuer Master Key Check Digits#>[CRLF]
```

Calling Parameters

`356`

Field 0, the command identifier.

`Derivation Type`

Field 1, the derivation type. This field contains a 1 byte decimal value defined as follows:

Derivation Type	Numerical Code
Common Session (per EMV 4.1 and Specification Update Bulletin 46)	2
Legacy VISA technique	3
EMV2000 (Tree-based technique)	8

$E_{MFK.9}$`(IMK),MAC`

Field 2, the Issuer Master Key encrypted under variant 9 of the MFK. This field contains a 32 byte value. This key must be either a double-length key.

`Application PAN`

Field 3, the Application Primary Account Number. This field is also used to indicate the Master Key derivation method. If this field contains the letter “B” followed by 17 to 19 digits, method B will be used, otherwise method A will be used.

`[PAN Sequence Number]`

Field 4, the optional application PAN sequence number. When present, this field contains a 2 digit decimal value. If not present a PAN Sequence Number of 00 will be used.

`[Diversification Data]`

Field 5, the value of this field depends on the derivation type specified in field 1.

For the common session derivation algorithm (if the derivation type, Field 1, is 2) this field contains a 16 byte hexadecimal value consisting of the following two items:

- 2 byte Application Transaction Counter (ATC). This binary value is expressed as 4 hexadecimal characters.
- 6 byte fixed value. This binary value “000000000000” is expressed as 12 hexadecimal characters.

For the EMV-Tree derivation algorithm (if the derivation type, Field 1, is 8) this field contains the four hexadecimal characters (2 bytes) of the Application Transaction Counter (ATC).

For the legacy Visa derivation algorithm (if the derivation type, Field 1, is 3) this field must be empty.

For the legacy Europay/Mastercard derivation algorithm (if the derivation type, Field 1, is 2) this field will contain either the same fields as the common session algorithm, or the four character ATC concatenated with 4 zero characters ‘0000’, followed by 4 bytes of hexadecimal characters (the unpredictable number).

`Partial AC or MAC`

Field 6, the value to be verified from the CAP token. This field must contain 4 to 16 hexadecimal characters; its length must be a multiple of 2. The value of this field depends on the content of field 12.

The effective IPB length (the number of 1-bits in the IPB) determines the maximum number of partial AC or MAC hexadecimal characters to supply in the command. The supplied partial AC or MAC must be zero padded on the right when total number of 1 bits of the IPB, divided by 4, is not an even number. For example an IPB of FFFFF00000000000 has twenty bits that have a value of 1. Twenty divided by 4 is 5, which is not an even number, therefore the partial AC or MAC must be right padded with a zero so its length will be 6. If more than the maximum number of partial AC or MAC characters are provided in the command, field 1of the response will be “N”, indicating that the token did not verify.

When the application is using CAP MODE 1 or MODE 2 without TDS, field 12 must be empty, and this field must contain the partial AC. When the default Partial IPB is used (field 8 is empty or not present), the partial AC must be 4 hexadecimal characters.

When the application is using CAP MODE 2 with TDS, field 12 is not empty, and this field must contain a MAC value.

`AC Padded Data Block`

Field 7, the data to be MACed to generate the Application Cryptogram. The length of this field is 16 to 1024 hexadecimal characters. The length of this field must be a multiple of 16 characters.

It is the responsibility of the host application to collect all necessary data and format it for processing. The Network Security Processor does not uncompress the CAP token to recover any portion of this data.

`[Partial IPB]`

Field 8, the 8 bytes from the Issuer Proprietary Bitmap (IPB) that indicates which bits of the calculated AC should be compared to the input in field 6. The shaded area in the table below highlights the location of the appropriate bytes.

PSN	CID	ATC	AC	IAD
1 byte	1 byte	2 bytes	8 bytes	0 - 32 bytes

If this field is empty, a default value of FFFF0000 00000000 is assumed. The length of this field must be zero or 16 ASCII-hex characters.

`[H#IV#Index#]`

These next three fields are present only if the derivation type is 8.

`[H#`

Field 9, the height value used for EMV-Tree derivation. This field contains the value 8 or 16, or it can be empty. If this field is empty and field 1 contains a value of 8 the height value of 8 will be used.

`IV#`

Field 10, the clear Initialization Vector used for EMV-Tree derivation. This field contains a 32-byte hexadecimal value, or it can be empty. If this field is empty and field 1 contains a value of 8 an IV of 32-bytes of binary zeros (nulls) will be used.

`Index#]`

Field 11, the index value used for EMV-Tree derivation.

This field contains a 1 digit decimal value between 0-7, or it can be empty. If this field is empty and the derivation type is 8 the index value of 7 will be used.

The index specifies the byte location of the key that will be exclusive Or'd with the ATC coefficient. An index value of zero indicates the leftmost byte of the key will be exclusive Or'd with the ATC coefficient. An index value of 7 indicates the rightmost byte of the key will be exclusive Or'd with the ATC coefficient. If the key is double-length the index value is applied to both halves of the double length key.

`[TDS data block]`

Field 12, if this field is present, the command will perform validation for CAP MODE 2 with TDS. The TDS data block consists of the transaction data that is to be MACed in this mode. The length of this field must be a multiple of 16 characters.

Table 8-15. Command 356: Validate CAP Token

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	3	356
1	Derivation Type	1	2, 3, 8
2	$E_{MFK.9}$(IMK),MAC	32	0 - 9, A - F
3	Application PAN	1-20	0 - 9, B
4	[PAN Sequence Number]	0, 2	0 - 9
5	[Diversification Data]	0, 4, 16	0 - 9, A - F
6	Partial AC or MAC	4 - 16	0 - 9, A - F
7	AC Padded Data Block	16 - 1024	0 - 9, A - F
8	[Partial IPB]	0, 16	0 - 9, A - F
9	[H#	0 - 2	8, 16

Table 8-15. Command 356: Validate CAP Token

Field #	Contents	Length (bytes)	Legal Characters
10	IV#	0, 32	0 - 9, A - F
11	Index#]	0, 1	0 - 7
12	[TDS Data Block]	0 - 1024	0 - 9, A - F

Responding Parameters

`456`

Field 0, the response identifier.

`Verification Indicator`

Field 1, signifies success or failure of the AC or MAC verification. This field contains 1 byte character either 'Y' (verification pass) or 'N' (verification fail).

`Session Key Check Digits`

Field 2, the first four digits of the result from encrypting zeros using the generated session key. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the session key.

`Issuer Master Key Check Digits`

Field 3, the first four digits of the result from encrypting zeros using the Issuer Master Key. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the Issuer Master Key.

Table 8-16. Response 456: Validate CAP Token

Field #	Contents	Length (bytes)	Legal Characters
0	Response indicator	3	456
1	Verification Indicator	1	Y or N
2	Session Key Check Digits	4 or 6	0-9, A-F
3	Issuer Master Key Check Digits	4 or 6	0-9, A-F

Usage Notes

- The Issuer Master Key must be encrypted under the MFK.

Examples

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Option A, CAP MODE 1 (no TDS field)

- EMV Derivation Type = 2 (EPI/MCI)
- Clear-text Issuer Master Key: 0123456789ABCDEF FEDCBA9876543210
  The Issuer Master Key encrypted under variant 9 of the MFK:
  94E1BA8235D38B08C7C77430C93D925A
- Application PAN = 9901234567890123
- Application PAN Sequence Number = 45
- [ATC or Random Number] = 1234567890123456
- partial AC or MAC = 9309
- Padded Data Block = 0123456789ABCDEF0123456789ABCDEF
- partial IPB = 8181818181818181

The command looks like this:

```
<356#2#94E1BA8235D38B08C7C77430C93D925A#9901234567890123#45#1
234567890123456#9309#0123456789ABCDEF0123456789ABCDEF#8181818
181818181#>
```

The Network Security Processor returns the following response:

```
<456#Y#0995#08D7#>
```

Example 2: Option A, CAP MODE 2 (with TDS)

- EMV Derivation Type = 2 (EPI/MCI)
- Clear-text Issuer Master Key: 165441472D13CED3 CFC7CB6ADF63C31A
  The Issuer Master Key encrypted under variant 9 of the MFK:
  34148C2307AC78DE413A9C3E9078D6B7
- Application PAN = 71372600550304
- Application PAN Sequence Number = 67
- [ATC or Random Number] = 6F1197963F72BBAD
- partial AC or MAC = D091
- Padded Data Block = DD3144D8C92138C5
- partial IPB = FFFF123400FFABCD
- TDS Data Block = DD3144D8C92138C5

The command looks like this:

```
<356#2#34148C2307AC78DE413A9C3E9078D6B7#71372600550304#67#6F1
197963F72BBAD#D091#DD3144D8C92138C5#FFFF123400FFABCD#DD3144D8
C92138C5#>
```

The Network Security Processor returns the following response:

```
<456#Y#7F90#2BD1#>
```

9 Storing Values in the Volatile Table

The volatile table is an area of Network Security Processor memory where you can temporarily store DES working keys, conversion tables, and Diebold Number Tables. This section describes the volatile table commands.

To skip this introduction go to Table 9-1 for a list of commands.

About the Volatile Table

The volatile table memory is erased when the Network Security Processor experiences a power outage, is reset to factory state, or when the Master File Key is promoted via command 9F. The Master File Key and the Pending Master File Key are stored in a separate non-volatile key table and are not erased in these situations.

The volatile table holds up to 9,999 1key-3DES (single-length) keys or 4,999 2key-3DES (double-length) keys, as they are stored in two consecutive volatile table locations.

Referencing a location

Instead of providing the 16 or 32 hexadecimal character value of the DES key in a command, the host application specifies the location in the volatile table were the value is stored. The location is specified in the following manner:

T*n*

where *n* is the location where the value has been stored.

For example, assume that volatile table location 75 contained a Key Exchange Key. The syntax for command 10 to generate a 2key-3DES PIN Encryption Key would be as follows:

```
<10#1#T75#>
```

Volatile Table Tasks

Using the volatile table typically involves the following tasks:

- Loading values into the table.
- Verifying the existence of values within the table.
- Deleting values when they are no longer needed.

Loading the Volatile Table

To load a DES key, conversion table, or a row of a Diebold Number Table into the volatile table, you must first generate the cryptogram. Once the cryptogram has been

created, you pass its value as a parameter in one of the key-table loading commands, 70,74, or 7F.

Keep a record of the Atalla Key Blocks you store in the volatile table. Reconstructing the table can be difficult unless you have a record of its contents.

Verifying Values in the Volatile Table

When the Network Security Processor loses power, is reset to factory state, or the Master File Key is promoted via command 9F, the volatile table is erased. With this in mind, you should periodically verify the table, using command 72, to be sure that it contains the correct values.

Deleting Values from the Volatile Table

On occasion you may need to erase values that are no longer being used. Use command 73 to delete the entire volatile table or Command 71 to delete a single table location. Command 74, which is used to load a row of the Diebold Number Table, will overwrite any value in the specified location.

Volatile Table Commands

The remainder of this section contains the command and response syntax for the Network Security Processor volatile table commands.

Quick Reference

Table 9-1 identifies each command by number, name, and purpose. Table 9-1 lists the commands in numerical order.

Table 9-1. Volatile Table Commands

Command #	Name	Purpose
70	Load Volatile Table Value	Loads a DES key or conversion table into the next available location in the table.
71	Delete Volatile Table Value	Deletes a value stored in a specific location.
72	Verify Volatile Table Value	Retrieves the check digits of a value stored in a specific location.
73	Clear Volatile Table	Clears the volatile table.
74	Load Diebold Number Table Row	Loads a row of the Diebold number table.
7F	Load Value to a Specific Location	Loads a DES key or conversion table into a specified location.

Load Volatile Table Value (Command 70)

Command 70 is used to load either a DES key or conversion table into the **first available** location of the volatile table. This command is enabled in the Network Security Processor's default security policy. Alternately you can use command 7F to load a DES key or conversion table into a specific location.

When loading a conversion table, the Network Security Processor does not check that the clear-text value of a conversion table contains all numeric digits.

This command is enabled in the Network Security Processor's default security policy.

Command

<70#Variant#$E_{MFK.V}$(Working Key)#>

Response

<80#Location#Remaining Locations#Check Digits#>[CRLF]

Calling Parameters

70

Field 0, the command identifier.

Variant

Field 1, the MFK variant (V) under which the key will be encrypted, establishing the key's function. This field can be one or two bytes long and can contain the numbers 0 - 31. See Key variants on page 2-2 for information on variants.

$E_{MFK.V}$(Working Key)

Field 2, the cryptogram of the key being loaded. The working key is encrypted using the MFK variant specified in Field 1. This field contains a 16 or 32 byte hexadecimal value.

Table 9-2. Command 70: Load Volatile Table Value

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	70
1	Variant (V)	1, 2	0 - 31
2	$E_{MFK.V}$(Working Key)	16, 32	0 - 9, A - F

Responding Parameters

`80`

Field 0, the response identifier.

`Location`

Field 1, the location where the DES working key or conversion table is stored. This field contains a number between 0000 and 9999.

`Remaining Locations`

Field 2, the number of locations available after this command is executed. This field contains a number in the range of 0000 through 9999.

`Check Digits`

Field 3, check digits; the first four digits that result from encrypting zeros using the DES working key or conversion table. If option 88 is enabled, this field will contain the first six digits of the result from encrypting zeros using the DES working key or conversion table.

Table 9-3. Response 80: Load Volatile Table Value

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	80
1	Location	4	0 - 9
2	Remaining Locations	4	0 - 9
3	Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 70, generate the cryptogram of the key to be loaded into the volatile table. If loading a conversion table, used in the IBM 3624 or NCR algorithms, it must be encrypted under variant 6 of the MFK.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Loading a key into the volatile table.

This example assumes that the volatile table is empty.

- Variant (V): 0.
- Clear-text key to be loaded: 0123 4567 89AB CDEF.
  The key to be loaded encrypted under variant 0 of the MFK: 9007 B875 1BB7 AB4E.

The command looks like this:

```
<70#0#9007B8751BB7AB4E#>
```

The Network Security Processor returns a response similar to this:

```
<80#0000#9998#D5D4#>
```

Delete Volatile Table Value (Command 71)

Command 71 deletes a value stored in a specific location within the volatile table. This command is enabled in the Network Security Processor's default security policy.

Command

```
<71#Location#>
```

Response

```
<81#Available Locations#>[CRLF]
```

Calling Parameters

`71`

Field 0, the command identifier.

`Location`

Field 1, the volatile table location that contains the value to be deleted. This field contains a 1 to 4 byte decimal value.

For 2key-3DES (double-length) keys, which occupy two adjacent key slot locations, you must delete the first key slot of the pair. For example, assume a 2key-3DES (double-length) key is loaded in key slots 1 and 2. If you attempt to delete key slot 2 you will receive an error 07, you must delete key slot 1, which will also delete key slot 2.

Table 9-4. Command 71: Delete Volatile Table Value

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	71
1	Location	1 - 4	0 - 9

Responding Parameters

`81`

Field 0, the response identifier.

`Available Locations`

Field 1, the number of available locations available after the command has been executed. This field contains a value in the range of 0000 through 9999.

Table 9-5. Response 81: Delete Volatile Table Value

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	81
1	Available Locations	4	0 - 9

Example

The following example illustrates deleting a value stored in location 35.

The command looks like this:

```
<71#35#>
```

The Network Security Processor returns a response similar to this:

```
<81#9999#>
```

Verify Volatile Table Value (Command 72)

Command 72 retrieves the check digits for the value currently stored at the specified location. This command supports both 1key-3DES (single-length) and 2key-3DES (double-length) working keys.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<72#Location#>
```

Response

```
<82#Available Locations#Check Digits#>[CRLF]
```

Calling Parameters

`72`

Field 0, the command identifier.

`Location`

Field 1, the location that contains the value you want to verify. This field must contain a number in the range of 0000 through 9999.

2key-3DES (double-length) keys are stored in two consecutive table locations, the check digits are stored in the first table location. To obtain the check digits of a 2key-3DES (double-length) key that is stored in table locations 3 and 4, specify table location 3 to obtain the check digits.

Table 9-6. Command 72: Verify Volatile Table Value

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	72
1	Location	1 - 4	0 - 9

Responding Parameters

`82`

Field 0, the response identifier.

`Available Locations`

Field 1, the number of locations available after the command has been executed. This field contains a value in the range of 0000 through 9999.

`Check Digits`

Field 2, the check digits for the value stored in the specified location. Check digits are the first 4 digits that result from encrypting zeros using the value. If option 88 is enabled, this field will contain the first six digits of the result.

Table 9-7. Response 82: Verify Volatile Table Value

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	82
1	Available Locations	4	0 - 9
2	Check Digits	4 or 6	0 - 9, A - F

Usage Notes

The Network Security Processor returns an error message <00#0701...> when the specified location is empty.

Example

Verifying a DES key in location 17.

The command looks like this:

```
<72#17#>
```

The Network Security Processor returns a response similar to this:

```
<82#1950#D5D4#>
```

Clear Volatile Table (Command 73)

Command 73 deletes all values stored in the volatile table.

▲ **WARNING.** Before you send this command, make sure no other host application is using the volatile table.

This command is enabled in the Network Security Processor's default security policy.

Command

```
<73#>
```

Response

```
<83#OK#>[CRLF]
```

Calling Parameters

`73`

Field 0, the command identifier.

Table 9-8. Command 73: Clear Volatile Table

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	73

Responding Parameters

`83`

Field 0, the response identifier.

`OK`

Field 1, the indicator that the command has been executed.

Table 9-9. Response 83: Clear Volatile Table

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	83
1	OK	2	OK

Usage Notes

Use Command 73 prior to loading the table to ensure that values are loaded in the appropriate order and are loaded into the correct locations.

Example

The following example illustrates clearing the entire table:

The command looks like this:

```
<73#>
```

The Network Security Processor returns the following response:

```
<83#OK#>
```

Load Diebold Number Table (Command 74)

Command 74 loads one row of the Diebold number table (DNT) into the volatile table. A Diebold Number Table contains 512 characters. It is organized as 32 rows, each of which is 16 hexadecimal characters. To load a Diebold Number Table you must load each row using a separate command 74, and specify consecutive volatile table locations. This command is enabled in the Network Security Processor's default security policy.

Command

<74#Location#$E_{MFK.0}$(Intermediate Key)#$E_{Intermediate\ Key.5}$(DNT1)#>

Response

<84#Location#>[CRLF]

Calling Parameters

74

Field 0, the command identifier.

Location

Field 1, the volatile table location that is being loaded. (The Diebold Number Table must be loaded into 32 contiguous volatile table locations.) You must specify the location in the successive command executions. The specified location will be loaded regardless of its current contents; therefore, be sure to coordinate the use of this command with other uses of the volatile table to avoid conflict and key overlay. This field contains a number between 0000 and 9999.

$E_{MFK.0}$(Intermediate Key)

Field 2, the intermediate key encrypted under variant 0 of the MFK. The intermediate key decrypts the cryptogram of the Diebold Number Table when the row is loaded into the volatile table, thus allowing it to exist in protected form on the database, but in clear form in the Network Security Processor. This field contains a 16 byte hexadecimal value, or a volatile table location.

$E_{Intermediate\ Key.5}(DNT_n)$

Field 3, one row of the Diebold number table, encrypted under variant 5 of the intermediate key. This field contains a 16 byte hexadecimal value.

Table 9-10. Command 74: Load Diebold Number Table

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	74
1	Location	1 - 4	0 - 9
2	$E_{MFK.0}$(Intermediate Key)*	16	0 - 9, A - F
3	$E_{Intermediate\ Key.5}$(DNT_n)	16	0 - 9, A - F

*Can be a volatile table location.

Responding Parameters

`84`

Field 0, the response indicator.

`Location`

Field 1, the volatile table location for the just-loaded row of the DNT. This field contains a number between 0000 and 9999.

Table 9-11. Response 84: Load Diebold Number Table

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	84
1	Location	4	0 - 9

Usage Notes

The encrypted rows of the Diebold Number Table are generated using the Secure Configuration Assistant-3 (SCA-3). The following instructions briefly describe the process using the SCA-3 and Network Security Processor to generate these cryptograms:

1. Use the SCA-3's Calculate Crypto function to input a 1key-3DES (single-length) key. Select Key for the Cryptogram Type, then select 0 for the variant.
   Complete transaction and record the encrypted value, it will be used as field 2 in command 11 in step 3 below, and in field 2 in command 74 in step 4 below.
2. Use the SCA-3's Calculate Crypto function to input a 1key-3DES (single-length) key. This should be row 1of the DNT.
   Select Diebold Number Table for the cryptogram type.
   Complete the transaction and record the encrypted value, it will be used in field 3 of command 11 in step 3 below.
   Repeat this step for the remaining 31 rows of the DNT.
3. Create a command 11 for each row of the DNT as follows:
   <11#5#result from step 1#result from step 2#>

Send this command to the Network Security Processor, record field 1 of the response, this value will be used as field 3 in command 74.

4. Create a command 74 for each row of the DNT as follows:
   <74#location#result from step 1#result from step 3#>

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Loading a row of the Diebold Number Table.

- Volatile Table Location number: 350.
- Clear-text Intermediate Key: 0123 4567 89AB CDEF.
  The Intermediate Key encrypted under variant 0 of the MFK: 9007B8751BB7AB4E.
- Clear-text row of the Diebold Number Table: C860 2A41 4D38 2C5B.
  The row of the Diebold Number Table encrypted under variant 5 of the Intermediate Key: 2144ADC8498E6920.

The command looks like this:

```
<74#350#9007B8751BB7AB4E#2144ADC8498E6920#>
```

The Network Security Processor returns the following response:

```
<84#0350#>
```

Load Value to a Specific Volatile Table Location (Command 7F)

Command 7F is used to load either a DES key or conversion table into an empty specified location in the volatile table. When the location referenced in the command already contains a DES key or conversion table, an error message <00#0603xx#> is returned. This command is enabled in the Network Security Processor's default security policy.

When loading a conversion table, the Network Security Processor does not check that the clear-text value of a conversion table contains all numeric digits.

Command

<7F#Variant#$E_{MFK.V}$(WK)#Location#>

Response

<8F#Location#Available Locations#Check Digits#>[CRLF]

Calling Parameters

7F

Field 0, the command identifier.

Variant

Field 1, the variant (V) of the MFK under which the value is encrypted. This field is one or two bytes long and contains the decimal values 0 to 31. See Key variants on page 2-2 for information on variants.

$E_{MFK.V}$(WK)

Field 2, the DES key or conversion table encrypted under the variant, specified in field 1, of the MFK. This field contains a 16 or 32 byte hexadecimal value.

Location

Field 3, the volatile table location. This value indicates where the value will be stored in the volatile table. This location must be empty. When loading a 2key-3DES (double-length) key, this location and subsequent location must both be empty. This field contains a number between 0000 and 9999.

Table 9-12. Command 7F: Load Value to a Specific Volatile Table Location

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	7F
1	Variant	1, 2	0 - 31
2	$E_{MFK.V}$(WK)	16, 32	0 - 9, A - F
3	Location	1 - 4	0 - 9

Responding Parameters

`8F`

Field 0, the response identifier.

`Location`

Field 1, the location where the value has been stored. This field contains a number in the range of 0000 through 9999.

`Available Locations`

Field 2, the number of locations available to store values. This field contains a number in the range of 0000 through 9999.

`Working Key Check Digits`

Field 3, the check digits for the value stored in the specified location. Check digits are the first 4 digits that result from encrypting zeros using the value. If option 88 is enabled, this field will contain the first six digits of the result.

Table 9-13. Response 8F: Load Value to a Specific Volatile Table Location

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	8F
1	Location	4	0 - 9
2	Available Locations	4	0 - 9
3	Check Digits	4 or 6	0 - 9, A - F

Usage Notes

Before using Command 7F, generate the key to be loaded into the volatile table. If loading a conversion table, used in the IBM 3624 or NCR algorithms, it must be encrypted under variant 6 of the MFK.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-5 for component values.

Loading a key into a predefined location of the volatile table.

- Variant: 1.
- Clear-text working key: 3333 3333 3333 3333.
  The working key encrypted under variant 1 of the MFK: 3219 92E9 44B0 F423.
- Location: 0008.

The command looks like this:

```
<7F#1#321992E944B0F423#0008#>
```

The Network Security Processor returns a response similar to this:

```
<8F#0008#9993#ADC6#>
```

10 Printing Commands

In version 1.35 the following commands have been added to support printing of letters that contain either a cleartext PIN or key component:

- Combine Key Components (Command 15E)
- Generate PIN Printing Key (Command 160)
- Print PIN Letter (Command 161)
- PIN Issuance: IBM 3624 Method (Command 162)
- PIN Issuance: Visa Method (Command 163)
- Divide a Key into Components (Command 16E)
- Print Component Letter (Command 16F)

These commands are disabled in the Ax160 NSP's default security policy.

Note. These commands must be enabled in the NSP's security policy prior to use, refer to Printing command configuration in section 4 of the SCA-3 user guide. These commands are disabled when the NSP is powered off.

Note. These commands are only allowed on the NIC1 Print Command Port.

Note. Commands Print PIN Letter (Command 161) and Print Component Letter (Command 16F) require the option 87 (enable NIC2) to be enabled, and these keywords must be present in the config.prm file: PORT_PRTCMD, PRINTER_ADDR_2, and PRINTER_PORT_2. For information on how to configure these values see section 4 of the

.

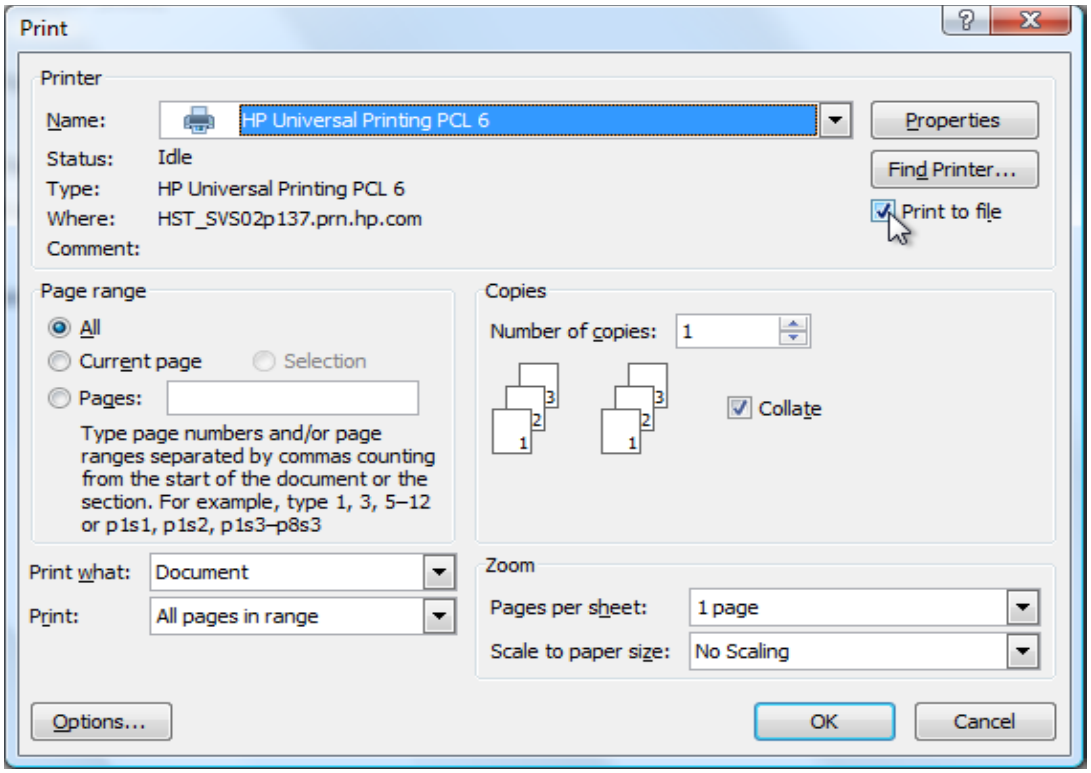
Letter template file

To print a PIN or key component letter the host application must first create a letter template file. Once this file has been created the host application can send it as binary data to the Ax160 NSP in either the Print PIN Letter (Command 161) or Print Component Letter (Command 16F) command. The Ax160 NSP will process the letter template file and create a print job which it will then send to the printer.

For maximum performance and efficiency the letter template file should be a simple ASCII text file. You can use a text editor such as Windows Notepad to create the letter template file. The printer's default values for font and size will be used to print the letter.

You can use Microsoft Word to create a complex letter template file. Typically they are large files that require multiple commands to send the entire letter template file to the Ax160 NSP. If you use Microsoft Word to create a letter template file all of these restrictions apply:

- The HP Universal Print Driver for Windows PCL6 is required on the PC/laptop that creates the letter template file. It is available for download from the HP.com website; search for HP Universal Print Driver for WIndows PCL6.
- The font for these marker strings must be either "`Courier`" or "`Courier New`", the remainder of the document can utilize any font.
- The entire PIN, component, check digit, or reference marker string must be input without using the copy/paste features and cannot be modified once it has been input.
- All marker strings must have a leading and trailing space.
- The data encoding method specified in field 10 of the print letter command must contain the letter W.
- To create the letter template file you must print it to the HP Universal PCL 6 driver and specify the **Print to File** option as shown in the screen shot below.



You must specify the filename to save the output file. The binary data of this output file must be supplied in the Data Block fields of the Print PIN Letter (Command 161) or Print Component Letter (Command 16F).

Marker strings

The letter template file must contain a marker string which is a unique value within the letter template file. The numbers 0-9 and the letters A-Z and a-z are allowed in a marker string.

There are four types of marker strings.

Marker Type	Command	Required	Length in characters
PIN	161	Yes	12
Component	16F	Yes	19
Check Digits	16F	Yes	varies based on the check digit method
Reference	16F	No	19

The Ax160 NSP will replace the marker string with the PIN, component, check digits, or reference value prior to sending the print job to the printer. For a PIN letter template sent to the Ax160 NSP in the Print PIN Letter (Command 161), the PIN marker string must be 12 characters. For a component letter template sent to the Ax160 NSP in the Print Component Letter (Command 16F), there are two required marker strings and one optional marker string. The component marker string must be 19 characters. For components that are longer than 16 characters the marker string must be repeated. For example, the component marker string must be present two times for a 2key-3DES key component. A maximum of four component marker strings may be present in a letter template file. If there are more component marker strings present in the letter template file than are needed (for example, when a letter template file contains 4 component marker strings but is printing only a 2-key 3DES component), the unused component marker strings will be filled with spaces. The length of the check digits marker string is based on the check digit method. The optional reference marker string, if present, must be 19 characters.

Printing large letter template files

The maximum size of the letter template file is 1,048,576 bytes (1 megabyte). If the letter template file is larger than 30,000 bytes the host application must split it into separate data blocks and send each data block as a separate command to the Ax160 NSP. The maximum size of a data block is 30,000 bytes. When a letter template file is split into multiple data blocks, information about the PIN or component (i.e. PIN block type, PIN Encryption Key, encrypted PIN block, etc.) must be included in only the final command.

The Ax160 NSP can receive a maximum of four concurrent multi-command letter template files. The Ax160 NSP's response to the first command in a multi-command sequence will include a continuation index that must be provided in the subsequent intermediate and final commands required to send the remainder of the letter template file. When the Ax160 NSP receives the final data block of the letter template file, it will replace the marker strings with the clear PIN or component and check digit values and then send the complete letter print job to the printer.

Printing an encrypted PIN

The PIN printing command requires that the PIN be encrypted under a PIN Printing Key. A special variant of the Master File Key is used to encrypt a PIN Printing Key. An ANSI or ISO-3 PIN block that is encrypted under a PIN Encryption Key must be re-

encrypted under a PIN Printing Key. To obtain an encrypted PIN that can be printed perform these steps:

1. Use Generate PIN Printing Key (Command 160) to generate a PIN Printing Key encrypted under a special variant of the Master File Key. The resulting value will be used as field 6 in the commands listed in step 2, and as field 7 in the command listed in step 3.
2. Use either PIN Issuance: IBM 3624 Method (Command 162) or PIN Issuance: Visa Method (Command 163) to produce an encrypted PIN block encrypted under the PIN Printing Key.
3. Use Print PIN Letter (Command 161) to print the encrypted PIN from step 2.

Printing a key component

The key component printing command requires that the component be encrypted under a special variant of the Master File Key. There are two ways to obtain key components that can be printed.

Divide an existing key into key components

To obtain an encrypted key component that can be printed perform these steps:

1. Use Divide a Key into Components (Command 16E) to create encrypted key components from an existing key that is encrypted under a variant of the Master File Key.
2. Use Print Component Letter (Command 16F) to print the encrypted key component from step 1.

Create new key components and combine them into a key

1. Use Print Component Letter (Command 16F) to generate a random key component and print it. Repeat this step to create the desired number of key components.
2. Use Combine Key Components (Command 15E) to combine the key components into a key that is encrypted under specified variant of the Master File key which can then be stored on the host application's key database. The response to the command also returns the key encrypted under the specified variant of the Key Exchange Key.

Printing a test page

Before attempting to print a batch of PIN or component letters it is highly recommended that the host application print a test page to ensure that the printer is online and operating properly. The test page feature can also be used to print operator instructions, job identifiers, start of job, and end of job pages.

Below are example commands that will print the text "This is a test page!!" on a page.

The command looks like this:

```
<161#0#0########A#B#21#21#This is a test page!!#>
```

The Network Security Processor returns the following response:

```
<261####>
```

The command looks like this:

```
<16F#0#0########A#B#21#21#This is a test page!!#>
```

The Network Security Processor returns the following response:

```
<26F####>
```

HP Printers

Only HP printers that support Printer Command Language version 6 are supported.

Managing printer sockets

The Ax160 NSP opens a socket on the printer after it has received all the print job data for a letter. After the socket is established, the Ax160 NSP sends the print job to the printer. The printer will acknowledge receipt of the print job and print the letter. The Ax160 NSP will then close the socket connection. The Ax160 NSP will open one socket for each complete print job that it will send to the printer. The Ax160 NSP can open a maximum of 16 sockets on the printer. The Ax160 NSP will wait for 75 seconds to establish a socket connection on the printer. If it cannot establish the socket connection within this time it will return an error 11 to the host application. The detailed error 11xx will indicate the cause of the error (see Detailed Errors for the specific detailed error values).

Printing errors

When the Ax160 NSP receives an error from the printer, it will return an error 11 to the host application. The detailed error 11xx will indicate the cause of the error (see Detailed Errors for the specific detailed error values).

The Ax160 NSP does not support status reporting from the printer, it only checks that the printer has received the print job. Once the printer acknowledges receipt of the print job the Ax160 NSP will return the response to the host application. If the printer is out of paper, the printer will buffer the job and print it once the operator loads the printer with paper.

Clearing the printer’s buffer

Printers store print jobs in their memory. After printing a job of PIN or component letters it is highly recommended that the print job be erased from the printer’s memory. The Ax160 NSP does not perform this function, it must be performed by an operator.

Combine Key Components (Command 15E)

Command 15E combines 3DES key components, which are encrypted under a special variant of the Master File Key, and then returns the key encrypted under the specified variant of the Key Exchange Key (KEK) and Master File Key (MFK). The minimum number of key components is two, and the maximum number of key components is four.

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

<15E#$E_{MFK.0}$(KEK)#$E_{MFK.VC}$(Comp-1)#$E_{MFK.VC}$(Comp-2)#
[$E_{MFK.VC}$(Comp-3)]#[$E_{MFK.VC}$(Comp-4)#Variant#Reserved#>

Response

<25E#$E_{KEK.V}$(WK)#$E_{MFK.V}$(WK)#Working Key Check Digits#>[CRLF]

Calling Parameters

15E

Field 0, the command identifier.

$E_{MFK.0}$(KEK)

Field 1, the Key Exchange Key (KEK) encrypted under variant 0 of the MFK. After combining the key components into a key, the Ax160 NSP uses the KEK to encrypt the key. This field contains a 32 hexadecimal value; the KEK must be a 2key-3DES key. The length of the KEK must be equal to or greater than the length of the key components.

$E_{MFK.VC}$(Comp-1)

Field 2, the first key component encrypted under the variant supplied in field 6 and special variant applied to the second byte of the MFK. This field contains either 16 or 32 hexadecimal character value.

$E_{MFK.VC}$(Comp-2)

Field 3, the second key component encrypted under the variant supplied in field 6 and special variant applied to the second byte of the MFK. This field contains either 16 or 32 hexadecimal character value.

$[E_{MFK.VC}(Comp-3)]$

Field 4, the third key component encrypted under the variant supplied in field 6 and special variant applied to the second byte of the MFK. If present, this field contains either 16 or 32 hexadecimal character value.

$[E_{MFK.VC}(Comp-4)]$

Field 5, the fourth key component encrypted under the variant supplied in field 6 and special variant applied to the second byte of the MFK. If present, this field contains either 16 or 32 hexadecimal character value.

Variant

Field 6, the variant applied to the KEK and MFK that will be used to encrypt the key.

Reserved

Field 7, this field must be empty.

Table 10-1. Command 15E: Combine Key Components

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	15E
1	$E_{MFK.0}$(KEK)	32	0 - 9, A - F
2	$E_{MFK.VC}$(Comp-1)	16, 32	0 - 9, A - F
3	$E_{MFK.VC}$(Comp-2)	16, 32	0 - 9, A - F
4	[$E_{MFK.VC}$(Comp-3)]	0, 16, 32	0 - 9, A - F
5	[$E_{MFK.VC}$(Comp-4)]	0, 16, 32	0 - 9, A - F
6	Variant	1-2	0 - 31
7	Reserved	0	empty

Responding Parameters

25E

Field 0, the response identifier.

$E_{KEK.V}$(Working Key)

Field 1, the working key encrypted under the variant of the KEK supplied in field 6 of the command. This field contains either 16 or 32 hexadecimal character value.

$E_{MFK.V}$(Working Key)

Field 2, the working key encrypted under the variant of the MFK supplied in field 6 of the command. This field contains either 16 or 32 hexadecimal character value.

`Working Key Check Digits`

Field 3, the check digits of the working key. The check digits are the first four digits that result from encrypting zeros using the working key. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the working key.

Table 10-2. Response 25E: Combine Key Components

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	25E
1	$E_{KEK.V}$(Working Key)	16, 32	0 - 9, A - F
2	$E_{MFK.V}$(Working Key)	16, 32	0 - 9, A - F
3	Working Key Check Digits	4, 6	0 - 9, A - F

Usage Notes

- Generate the KEK cryptogram.
- All key components must be the same length. An error <03xx...#> will be returned which points to the field in the command that contains a component whose length is not equal to the length of component 1. The detailed error code is 209.
- The length of the KEK must be equal to or greater than the length of the key components. If the length of any of the key components is greater than the length of the KEK an error <00#0301....#> will be returned. The detailed error code is 209.
- If the combination of the key components produces a weak or semi-weak key (for a list of these keys, see Table A-1, Weak and Semi-weak Keys), an error<00#0600....#> will be returned. The detailed error is 513.
- The resulting key will not be adjusted to odd parity.

Example

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Combine two 2key-3DES key components into a CVV key

The 2key-3DES KEK:0123 4567 89AB CDEF FEDC BA98 7654 3210, check digits = 08D7.

The KEK encrypted under MFK.0: 9007B8751BB7AB4E0B176C3EBEED18AF

Key component 1: 13B55EEA2083B658 E34F61BCABF119C2, check digits = 09D9.
Key component 1 encrypted under the special variant of the MFK:
8C000382F8593B90EAFB7D1D2AEE6025

Key component 2: 8045623E3D70E3DA 75E3E61F8F01DC7F, check digits = 077E.
Key component 2 encrypted under the special variant of the MFK:
07B986EF749264D276DFEA945E549CDA

Variant: 3

The command looks like this:

```
<15E#9007B8751BB7AB4E0B176C3EBEED18AF#8C000382F8593B90EAFB7D1
D2AEE6025#07B986EF749264D276DFEA945E549CDA###3##>
```

The Network Security Processor returns the following response:

```
<25E#B663B8EB7AE5FDC9745912E369C8EBD5#8C3303B887AC9E338742E75
0DA7DCB27#C99D#>
```

Generate PIN Printing Key (Command 160)

This command is used to generate a PIN printing key. This key is encrypted under a special variant of the Master File Key and can only be used in these three commands:

- Print PIN Letter (Command 161)
- PIN Issuance: IBM 3624 Method (Command 162)
- PIN Issuance: Visa Method (Command 163)

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

```
<160#Variant#Key Length#>
```

Response

<260#$E_{MFK.VP}$(PIN Printing Key)#Check Digits#>[CRLF]

Calling Parameters

`160`

Field 0, the command identifier.

`Variant`

Field 1, the variant applied to the Master File Key that will encrypt the PIN Printing Key. This field must contain this value "1p".

`Key Length`

Field 2, the length of the PIN Printing Key to be generated. This field must contain the letter "D"

Table 10-3. Command 160: Generate PIN Printing Key

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	160
1	Variant	1	1p
2	Key Length	1	D

Responding Parameters

`260`

Field 0, the response identifier.

$E_{MFK.VP}$`(PIN Printing Key)`

Field 1, the PIN Printing Key encrypted under the printing variant of the Master File Key. This field will contain 32 hexadecimal characters.

`Check Digits`

Field 2, the PIN Printing Key check digits. The first four digits that result from encrypting zeros using this key. If option 88 is enabled this field will contain the first six digits of the result.

Table 10-4. Response 260: Generate PIN Printing Key

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	260
1	$E_{MFK.VP}$(PIN Printing Key)	32	0 - 9, A - F
2	Check Digits	4,6	0 - 9, A - F

Example

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

The command looks like this.

```
<160#1p#D#>
```

The Network Security Processor generates a random key and returns a response similar to this:

```
<260#749022577868A133E747FD1A9B4F90BB#78B7#>
```

Print PIN Letter (Command 161)

This command is used to send a PIN letter print job to the printer.

▲ **WARNING.** The print job will contain the cleartext PIN. Appropriate security measures are required to ensure that only authorized personnel have access to the printer, and that communications between the Ax160 NSP and the printer are not monitored.

The host application creates a PIN letter template (as a standard ASCII text file or Microsoft Word document). The PIN letter template must contain a unique 12 character PIN marker string value, for example "123456789012" or "xxxxxxxxxxxx"). The PIN marker string indicates where the cleartext PIN will be inserted into the PIN letter template; it must be present only once.

The host application uses this command to send the PIN letter template to the Ax160 NSP along with the encrypted PIN which is encrypted under a PIN Printing Key and also the PIN Printing Key encrypted under a special variant of the MFK. The Ax160 NSP decrypts the PIN, searches the PIN letter template for the PIN marker string and then replaces the PIN marker string with the cleartext PIN value (right padded with spaces if necessary). The Ax160 NSP then sends the PIN letter print job to the printer.

The maximum size of the PIN letter template is 1,048,576 bytes (1 megabyte). If the PIN letter template is larger than 30,000 bytes the host application must split it into separate data blocks and send each data block as a separate command to the Ax160 NSP. When a PIN letter template is split into multiple data blocks, information about the PIN (i.e. PIN block type, PIN Encryption Key, encrypted PIN block, etc.) and the PIN marker string must be included in only the final command.

The Ax160 NSP can receive a maximum of four concurrent multi-command PIN letter templates. The Ax160 NSP's response to the first command in a multi-command sequence will include a continuation index that must be provided in the subsequent intermediate and final commands required to send the remainder of the PIN letter template to the Ax160 NSP. When the Ax160 NSP receives the final data block of the PIN letter template, it will replace the PIN marker string with the clear PIN and then send the complete PIN letter print job to the printer.

To reduce the PIN letter template size, company logos and other graphics should be preprinted on the paper that is loaded into the printer.

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

```
<161#Letter Type#Continuation Flag#[Continuation Index]#
[PIN Block Type]#[E_KPP(PIN Block)]#Variant#
[E_MFK.VP(PIN Printing Key)]#[PIN Block Digits]#
[PIN Marker String]#Data Encoding#Data Type#
Letter Template Size#Data Block Length#Data Block#>
```

Response

```
<261#[Continuation Index]#[KPP Check Digits]#
[PIN Sanity Error]#>[CRLF]
```

Calling Parameters

`161`

Field 0, the command identifier.

`Letter Type`

Field 1, this field is specifies the type of letter to be printed.

Specify a letter type of 0 (zero) to print a test page. The following restrictions apply to printing a test page: field 2 must be contain the number 0 (zero), fields 3 through 9 must be empty, and field 10 must be contain the letter A.

To print a PIN letter specify a letter type value of 1.

`Continuation Flag`

Field 2, the continuation flag. The table below defines the allowed values.

Value	Description
0	Entire PIN letter template is included in this command.
1	The command contains the first block of a multi-block PIN letter template.
2	The command contains an intermediate block of a multi-block PIN letter template.
3	The command contains the final block of a multi-block PIN letter template.
4	Cancel current print job; removes a partial PIN letter template from Ax160 NSP's memory.

`[Continuation Index]`

Field 3, this index specifies which of the four internal memory storage locations the

Ax160 NSP used to store the first PIN letter template data block.This field must be empty when the continuation flag (field 2) is set to a value of 0 or 1. This field must be empty if the command is used to send the first data block of the PIN letter template. For subsequent commands used to send intermediate and final data blocks the value of this field must match the value returned in field 1 of the response to the command that was used to send the first data block of the PIN letter template. When the continuation flag (field 2) is set to a value of 2, 3, or 4, this field can contain the values 0, 1, 2, or 3.

`[PIN Block Type]`

Field 4, the incoming PIN block format. This field must contain one of these values:

Value	Description
1	ANSI (ISO-0) Format PIN Block.
8	ISO-3 Format PIN Block.

This field should be empty when the continuation flag (field 2) is set to a value of 1, 2, or 4.

$[E_{PPK}(PIN\ Block)]$

Field 5, the ANSI PIN Block or ISO-3 PIN Block encrypted under a PIN Printing Key. This field contains a 16 hexadecimal digit value. This field should be empty when the continuation flag (field 2) is set to a value of 1, 2, or 4.

`Variant`

Field 6, this field must contain the value 1p.

$[E_{MFK.VP}(KPP),MAC]$

Field 7, the PIN Printing Key encrypted under the PIN printing variant of the MFK. When option 6C is enabled, this field can contain a 1key-3DES (single-length) key; otherwise it must contain a 2key-3DES key. This field should be empty when the continuation flag (field 2) is set to a value of 1, 2, or 4.

`[PIN Block Digits]`

Field 8, the account number digits used to format the ANSI or ISO-3 PIN block. This field contains 12 numeric digits. This field should be empty when the continuation flag (field 2) is set to a value of 1, 2, or 4.

`[PIN Marker String]`

Field 9, the 12 character PIN marker string in the print letter template that identifies the location where the cleartext PIN will be printed. This field can contain upper and lower case letters (A-Z, a-z) and numeric digits (0-9). When printed in the letter, the PIN will be left justified and space filled. For example a five digit PIN will

print in the leftmost 5 positions followed by 7 spaces. This field should be empty when the continuation flag (field 2) is set to a value of 1, 2, or 4.

`Data Encoding`

Field 10, the encoding used for the PIN marker string in the letter template file. This field can contain one of these values:

Value	Description
A	ASCII encoding, where 1234 = 0x31323334.
W	Windows encoding (16-char, little endian) where 1234 = 0x3100320033003400.

`Data Type`

Field 11, only binary is supported. This field must contain the letter B.

`Letter Template Size`

Field 12, the size of the complete PIN letter template. The maximum size of the PIN letter template is 1,048,576 bytes (1 megabyte).

`Data Block Length`

Field 13, the number of bytes of the data sent in this data block. The maximum value is 30000.

`Data Block`

Field 14, the binary data of the PIN letter template. The maximum amount of binary data is 30000 bytes.

Table 10-5. Command 161: Print PIN Letter (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	161
1	Letter Type	1	0,1
2	Continuation Flag	0,1	0-4
3	[Continuation Index]	0,1	0-3
4	[PIN Block Type]	0,1	1,8
5	[E_{KPP}(PIN Block)]	0, 16	0-9, A-F
6	Variant	2	1p
7	[$E_{MFK.VP}$(KPP)]	0, 16, 32	0-9, A-F
8	[PIN Block Digits]	0, 12	0-9
9	[PIN Marker String]	0, 12	0-9, A-Z, a-z
10	Data Encoding	1	A, W
11	Data Type	1	B

Table 10-5. Command 161: Print PIN Letter (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
12	Letter Template Size	1-7	0-1048576
13	Data Block Length	1-5	0-30000
14	Data Block	1-30000	binary

Responding Parameters

`261`

Field 0, the response identifier.

`[Continuation Index]`

Field 1, this field will contain a value in the range of 0 through 3 if the continuation flag (command-field 2) contains the number 1. It will match field 3 of the command if the continuation flag (command-field 2) is 2, 3, or 4. It will be empty if the continuation flag is 0.

`[KPP Check Digits]`

Field 2, the PIN Printing Key check digits. The first four digits that result from encrypting zeros using the KPE. If option 88 is enabled this field will contain the first six digits of the result. This field will be empty when the continuation flag (command-field 2) is 1, 2, or 4.

`[PIN Sanity Error]`

Field 3, the PIN sanity error. This field will be empty when the continuation flag (command-field 2) is 1, 2, or 4. If the Ax160 NSP is able to successfully decrypt the encrypted PIN block (command-field 5) this field will be empty. If the Ax160 NSP is unable to correctly decrypt the encrypted PIN block, or if the length of the decrypted PIN is less than the value defined in option A0 or greater than 12, this field will contain either the letter S or L depending upon how option A1 is configured.

Table 10-6. Response 261: Print PIN Letter

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	261
1	[Continuation Index]	0,1	0-3
2	[KPP Check Digits]	0,4,6	0 - 9, A - F
3	[PIN Sanity Error]	0,1	S,L

Usage Notes

When a letter template file requires multiple commands to process the entire letter template file, the Ax160 NSP will clear the entire letter template file from its memory on any of the error conditions listed below. In this case correct the error and send all of the commands required to process the entire letter template file again.

- Invalid letter template length - the total number of bytes received is not equal to the total number of bytes specified in the Letter Template Size (field 12) of the command.
- Invalid PIN Printing Key specified in field 7.The Ax160 NSP will return an error code 07.
- TCP/IP connection or send/receive error is detected. The Ax160 NSP will return an error code 11.
- Ax160 NSP execution error. The Ax160 NSP will return an error code 08.
- Cannot find the marker strings in the template file. The Ax160 NSP will return an error code 12.
- The decrypted PIN block fails the sanity test. The Ax160 NSP will return either an S or L in field 3 of the response.

Command syntax errors, such as invalid number of fields in a command or invalid character in a command, will not cause the Ax160 NSP to erase the letter template file from its internal memory slot. In this case correct the syntax error and send the command again to print the PIN letter.

Example

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

- 2key-3DES KPP:0123 4567 89AB CDEF FEDC BA98 7654 3210,
  check digits = 08D7
  The KPP encrypted under the special variant of the MFK:
  CEA90AE716BB3929D5EA8BD331846B35
- Clear PIN to be printed: 1234
- Clear ANSI PIN Block: 041262876FEDCBA9
- Encrypted ANSI PIN Block: E08962A98076BF5C
- PIN Marker String: xxxxxxxxxxxx

ASCII PIN Letter Template Text

```
Mr John Smith
1234 Main Street
Anytown, CA, 123456



Dear Mr Smith,


Your new PIN is : xxxxxxxxxxxx


Please keep your PIN safe.


Regards,


AnyBank
```

The command looks like this. For visibility purposes in this example the binary data in the data block field (field 14) is presented in hexadecimal format.

```
<161#1#0##1#E08962A98076BF5C#1p#CEA90AE716BB3929D5EA8BD331846
B35#567890123456#xxxxxxxxxxxx#A#B#162#162#4D72204A6F686E20536
D6974680D0A31323334204D61696E205374726565740D0A416E79746F776E
2C2043412C203132333435360D0A0D0A0D0A44656172204D7220536D69746
82C0D0A0D0A596F7572206E65772050494E206973203A2078787878787878
7878787878200D0A0D0A506C65617365206B65657020796F75722050494E2
0736166652E0D0A0D0A52656761726732C0D0A0D0A416E7942616E6B0D0A
#>
```

The Network Security Processor returns the following response:

```
 <261##08D7##>
```

PIN Issuance: IBM 3624 Method (Command 162)

This command can generate or calculate a PIN and IBM 3624 offset. Three modes of operations are supported:

- Calculate an offset from an encrypted PIN block
- Generate a random PIN and calculate the offset
- Calculate the PIN from an offset

The response to the command will contain an encrypted PIN block encrypted under a PIN printing key. Use command 161 to print the encrypted PIN.

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

<162#Algorithm#Mode#PIN Block Format#[E_{KPE}(PIN)]#
[$E_{MFK.1}$(KPE)]#$E_{MFK.VP}$(KPP)#$E_{MFK.4}$(KPV)#[Offset]#
[PIN Length]#Conversion Table#Validation Data#Pad#
PIN Block Data#>

Response

<262#E_{KPP}(PIN Block)/Sanity Error#[Offset]#>[CRLF]

Calling Parameters

`162`

Field 0, the command identifier.

`Algorithm`

Field 1, the PIN algorithm. This field must contain the number 2.

`Mode`

Field 2, the mode of operation. This field must contain one of these values:

Mode	Description
1	Calculate an offset from an encrypted PIN block.
2	Generate a random PIN and calculate the offset.
3	Calculate the PIN from an offset.

`PIN Block Format`

Field 3, the PIN block format. This field must contain one of these values:

Value	Description
1	ANSI PIN Block (ISO-0)
8	ISO-3 PIN Block

$[E_{KPE}(PIN\ Block)]$

Field 4, the ANSI or ISO-3 PIN block encrypted under the PIN Encryption Key (KPE). This field must contain a 16 hexadecimal digit value when the mode (field 2) is 1, in all other cases this field must be empty.

$[E_{MFK.1}(KPE)]$

Field 5, the PIN Encryption Key (KPE) used to encrypt the PIN supplied in field 4. The KPE must be encrypted under variant 1 of the MFK. This field must contain either a 16 or 32 hexadecimal character value when the mode (field 2) is 1, in all other cases this field must be empty. When option 6C is enabled, this field can contain a 1key-3DES (single-length) key; otherwise it must contain a 2key-3DES key. When the mode (field 2) is 2, this field must be empty.

$E_{MFK.VP}(KPP)$

Field 6, the PIN Printing Key that will be used to encrypt the PIN returned in the response. The PIN Printing Key must be encrypted under the PIN printing variant of the MFK.This field must contain a 32 hexadecimal character value.

$E_{MFK.4}(KPV),MAC$

Field 7, the PIN Verification Key (KPV) encrypted under variant 4 of the MFK. This field must contain either a 16 or 32 hexadecimal character value.

`[Offset]`

Field 8, the PIN offset. This field must be empty when the mode (field 2) is 1 or 2. When the mode is 3, this field can contain a 4 through 12 digit numeric value or if empty the Ax160 NSP will generate an offset of all zeros equal to the PIN length.

`[PIN Length]`

Field 9, the PIN length. This field must be empty when the mode (field 2) is 1. When the mode is 2 or 3, this field contains the length of the PIN or offset to be calculated. When present, this field must contain a numerical value in the range 4 through 12.

`Conversion Table`

Field 10, a table that maps hexadecimal digits (0 through 9, A through F) to decimal digits (0 through 9). This field contains the 16 decimal digit value of the clear-text conversion table. When option 48 is enabled, this field contains the conversion table in AKB format (the header must be 1nCNE000). When option 4E is enabled, the conversion table must adhere to these rules:

- The conversion table must have at least eight unique digits.
- No single digit can occur more than four times.

`Validation Data`

Field 11, validation data. This value is unique for each card holder and is typically the account number. This field contains a 4 to 16 byte hexadecimal value. When option 4C is enabled, the value supplied in this field must be 12 digits in length and match the PIN Block Data value supplied in field 13.

`Pad`

Field 12, the pad character which right-pads the validation data. This field contains a one byte hexadecimal value. The pad character is only applied when the validation data is less than 16 characters in length.

`PIN Block Data`

Field 13, the account number digits used to format the ANSI or ISO-3 PIN block. This field contains 12 numeric digits.

Table 10-7. Command 162: PIN Issuance: IBM 3624 Method (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	162
1	Algorithm	1	2
2	Mode	1	1-3
3	PIN Block Format	1	1,8
4	[E_{KPE}(PIN)]	0,16	0-9, A-F
5	[$E_{MFK.1}$(KPE)]	16, 32	0-9, A-F
6	$E_{MFK.VP}$(KPP)	32	0-9, A-F
7	$E_{MFK.V}$4(KPV)	16, 32	0-9, A-F
8	[Offset]	0, 4-12	0-9

Table 10-7. Command 162: PIN Issuance: IBM 3624 Method (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
9	[PIN Length]	0-2	4-12
10	Conversion Table	16	0-9, A-F
11	Validation Data	4-16	0-9, A-F
12	Pad	1	0-9, A-F
13	PIN Block Data	12	0-9

Responding Parameters

`262`

Field 0, the response identifier.

E_{KPP}`(PIN Block)/Sanity Error`

If the PIN block (command-field 4) passes the PIN sanity test this field will contain the ANSI or ISO-3 PIN block encrypted under the PIN Printing Key. If the PIN block fails the sanity test this field will contain a sanity error. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. Sanity errors are:

- S – PIN failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range

`[Offset]`

Field 2, the offset. This field will contain the offset when the mode (command-field 2) is 1 or 2 and field 1 of the response does not contain a sanity error. When the mode is 3, this field will be empty.

Table 10-8. Response 262: PIN Issuance: IBM 3624 Method

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	262
1	E_{KPP}(PIN Block) / Sanity Error	16, 1	0-9, A-F, S, L
2	[Offset]	0, 4-12	0-9

Usage Notes

- Use Generate PIN Printing Key (Command 160) to generate the PIN Printing Key.

Examples

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Mode 1 - Calculate an offset from an encrypted ANSI PIN block

- PIN block type: ANSI (1)
- PAN: 5555557890123456
- ANSI PIN block: 041261A876FEDCBA, clear PIN = 1234
  The encrypted ANSI PIN block: F81BD4D6E8AC404E
- PIN Encryption Key (KPE): 0123456789ABCDEF FEDCBA9876543210
  The PIN Encryption Key encrypted under variant 1 of the MFK:
  AE86D417E64E07E0BC62A2AD72516EA1
- PIN Printing Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits = 08D7
  The KPP encrypted under the special variant of the MFK:
  CEA90AE716BB3929D5EA8BD331846B35
- PIN Verification Key (KPV): 1234123412341234 5678567856785678
  The PIN Verification Key encrypted under variant 4 of the MFK:
  2979F6551D0084AC4B2EF58A726348FE
- Conversion table: 0123456789012345
- Validation data: 7890123456
- Pad character: F
- PIN block data: 555789012345

The command looks like this:

```
<162#2#1#1#F81BD4D6E8AC404E#AE86D417E64E07E0BC62A2AD72516EA1#
CEA90AE716BB3929D5EA8BD331846B35#2979F6551D0084AC4B2EF58A7263
48FE###0123456789012345#7890123456#F#555789012345#>
```

The Network Security Processor returns the following response:

```
<262#F81BD4D6E8AC404E#3953#>
```

Mode 2 - Generate a 4 digit PIN and calculate the offset

- PIN block type: ANSI (1)
- PAN: 5555557890123456
- PIN Printing Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits = 08D7
  The KPP encrypted under the special variant of the MFK:
  CEA90AE716BB3929D5EA8BD331846B35

- PIN Verification Key (KPV): 1234123412341234 5678567856785678
  The PIN Verification Key encrypted under variant 4 of the MFK:
  2979F6551D0084AC4B2EF58A726348FE
- PIN Length: 4
- Conversion table: 0123456789012345
- Validation data: 7890123456
- Pad character: F
- PIN block data: 555789012345

The command looks like this:

```
<162#2#2#1###CEA90AE716BB3929D5EA8BD331846B35#2979F6551D0084A
C4B2EF58A726348FE##4#0123456789012345#7890123456#F#5557890123
45#>
```

The Network Security Processor generates a random PIN the response will be similar to this:

```
<262#9762AD4C109FFDD0#7630#>
```

Mode 3 - Calculate the PIN from an offset

- PIN block type: ANSI (1)
- PAN: 5555557890123456
- PIN Printing Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits = 08D7
  The KPP encrypted under the special variant of the MFK:
  CEA90AE716BB3929D5EA8BD331846B35
- PIN Verification Key (KPV): 1234123412341234 5678567856785678
  The PIN Verification Key encrypted under variant 4 of the MFK:
  2979F6551D0084AC4B2EF58A726348FE
- PIN Length: 4
- Conversion table: 0123456789012345
- Validation data: 7890123456
- Pad character: F
- PIN block data: 555789012345

The command looks like this:

```
<162#2#3#1###CEA90AE716BB3929D5EA8BD331846B35#2979F6551D0084A
C4B2EF58A726348FE#3953#4#0123456789012345#7890123456#F#555789
012345#>
```

The Network Security Processor returns the following response:

```
<262#F81BD4D6E8AC404E##>
```

PIN Issuance: Visa Method (Command 163)

This command can generate or calculate a PIN and Visa PIN Verification Value. Two modes of operations are supported:

- Calculate the PIN Verification Value (PVV) from an encrypted PIN block.
- Generate a random PIN and calculate the PIN Verification Value (PVV).

The response to the command will contain an encrypted PIN block encrypted under a PIN Printing Key. Use command 161 to print the encrypted PIN.

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

<163#Algorithm#Mode#PIN Block Format#[E_{KPE}(PIN)]#
[$E_{MFK.1}$(KPE)]#$E_{MFK.VP}$(KPP)#$E_{MFK.4}$(KPV)#[PIN Length]#PVKI#
Validation Data#PIN Block Data#>

Response

<263#E_{KPP}(PIN Block)/Sanity Error#[PVV]#>[CRLF]

Calling Parameters

163

Field 0, the command identifier.

Algorithm

Field 1, the PIN algorithm. This field must contain the number 3.

Mode

Field 2, the mode of operation. This field must contain one of these values:

Mode	Description
1	Calculate the PIN Verification Value (PVV) from an encrypted PIN block.
2	Generate a random PIN and calculate the PIN Verification Value (PVV).

`PIN Block Format`

Field 3, the PIN block format. This field must contain one of these values:

Value	Description
1	ANSI PIN Block (ISO-0)
8	ISO-3 PIN Block

`[E`$_{KPE}$`(PIN Block)]`

Field 4, the ANSI or ISO-3 PIN block encrypted under the PIN Encryption Key (KPE). This field must contain a 16 hexadecimal digit value when the mode (field 2) is 1. If the decrypted PIN contains a PIN that is more than 4 digits in length the Ax160 NSP will use only the leftmost 4 digits to calculate the PVV. When the mode (field 2) is 2, this field must be empty.

`[E`$_{MFK.1}$`(KPE)]`

Field 5, the PIN Encryption Key (KPE) used to encrypt the PIN supplied in field 4. The KPE must be encrypted under variant 1 of the MFK. This field must contain either a 16 or 32 hexadecimal character value when the mode (field 2) is 1. When the mode (field 2) is 2, this field must be empty. When option 6C is enabled, this field can contain a 1key-3DES (single-length) key; otherwise it must contain a 2key-3DES key.

`E`$_{MFK.VP}$`(KPP)`

Field 6, the PIN Printing Key that will be used to encrypt the PIN returned in the response. The PIN Printing Key must be encrypted under the printing variant of the MFK. This field must contain a 32 hexadecimal character value.

`Header,E`$_{MFK.4}$`(KPV),MAC`

Field 7, the PIN Verification Key (KPV) encrypted under variant 4 of the MFK. This field must contain a 32 hexadecimal character value (KeyLeft||KeyRight). When option 6A is enabled, KeyLeft can be the same value as KeyRight.

`[PIN Length]`

Field 8, the PIN length. This field must be empty when the mode (field 2) contains the number 1.This field must contain the number 4 when the mode (field 2) contains the number 2.

`PVKI`

Field 9, the PIN Verification Key Indicator (PVKI) used in the algorithm to calculate the PVV. This field contains a 1 byte decimal value in the range of 0 to 9.

`Validation Data`

Field 10, validation data. This value is unique for each card holder and is typically a portion of the account number. This field contains an 11 digit numeric value. When option 4C is enabled, these 11 digits must be present in the PIN block data value supplied in field 11.

`PIN Block Data`

Field 11, the account number digits used to format the ANSI or ISO-3 PIN block. This field contains 12 numeric digits.

Table 10-9. Command 163: PIN Issuance: Visa Method

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	163
1	Algorithm	1	3
2	Mode	1	1,2
3	PIN Block Format	1	1,8
4	[E_{KPE}(PIN)]	0,16	0-9, A-F
5	[$E_{MFK.1}$(KPE)]	0, 16, 32	0-9, A-F
6	$E_{MFK.VP}$(KPP)	32	0-9, A-F
7	$E_{MFK.4}$(KPV)	32	0-9, A-F
8	[PIN Length]	0,1	4
9	PVKI	1	0-9
10	Validation Data	11	0-9
11	PIN Block Data	12	0-9

Responding Parameters

`263`

Field 0, the response identifier.

``$E_{KPP}$(PIN Block)/Sanity Error``

Field 1, if the PIN block (command-field 4) passes the PIN sanity test this field will contain the ANSI or ISO-3 PIN block encrypted under the PIN Printing Key. If the PIN block fails the sanity test this field will contain a sanity error. Option 4B specifies the type of PIN sanity test to be performed on the incoming PIN block. Sanity errors are:

- S – PIN failed the sanity test. Or the length of the PIN is out of range and PIN-length error reporting has not been enabled. See PIN Sanity Error and option A1.
- L – the length of the PIN is out of range.

[PVV]

Field 2, the 4 numeric digit PIN Verification Value. This field will be empty if a sanity error is present in field 1 of the response.

Table 10-10. Response 263: PIN Issuance: Visa Method

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	263
1	E_{KPP}(PIN Block) / Sanity Error	16, 1	0- 9, A-F, S, L
2	[PVV]	0,4	0- 9

Usage Notes

- Generate the PIN Printing Key.

Examples

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Mode 1 - Calculate the PVV from an encrypted ANSI PIN block

- PIN block type: ANSI (1)
- PAN: 5555557890123456
- ANSI PIN block: 041261A876FEDCBA, clear PIN = 1234
  The encrypted ANSI PIN block: F81BD4D6E8AC404E
- PIN Encryption Key (KPE): 0123456789ABCDEF FEDCBA9876543210
  The PIN Encryption Key encrypted under variant 1 of the MFK:
  AE86D417E64E07E0BC62A2AD72516EA1
- PIN Printing Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits = 08D7
  The KPP encrypted under the special variant of the MFK:
  CEA90AE716BB3929D5EA8BD331846B35
- PIN Verification Key (KPV): 1234123412341234 5678567856785678
  The PIN Verification Key encrypted under variant 4 of the MFK:
  2979F6551D0084AC4B2EF58A726348FE
- PIN Length: empty
- PVKI: 1
- Validation data: 55789012345
- PIN block data: 555789012345

The command looks like this:

```
<163#3#1#1#F81BD4D6E8AC404E#AE86D417E64E07E0BC62A2AD72516EA1#
CEA90AE716BB3929D5EA8BD331846B35#2979F6551D0084AC4B2EF58A7263
48FE##1#55789012345#555789012345#>
```

The Network Security Processor returns the following response:

```
<263#F81BD4D6E8AC404E#0177#>
```

Mode 2 - Generate a 4 digit PIN and calculate the PVV

- PIN block type: ANSI (1)
- PAN: 5555557890123456
- PIN Printing Key: 0123456789ABCDEF FEDCBA9876543210,
  check digits = 08D7
  The KPP encrypted under the special variant of the MFK:
  CEA90AE716BB3929D5EA8BD331846B35
- PIN Verification Key (KPV): 1234123412341234 5678567856785678
  The PIN Verification Key encrypted under variant 4 of the MFK:
  2979F6551D0084AC4B2EF58A726348FE
- PIN Length: 4
- PVKI: 1
- Validation data: 55789012345
- PIN block data: 555789012345

The command looks like this:

```
<163#3#2#1###CEA90AE716BB3929D5EA8BD331846B35#2979F6551D0084A
C4B2EF58A726348FE#4#1#55789012345#555789012345#>
```

The Network Security Processor generates a random PIN the response will be similar to this:

```
<263#CF2A153CA3533E7F#6632#>
```

Divide a Key into Components (Command 16E)

This command divides a 3DES key into multiple random key components. The key components are returned encrypted under a special variant of the MFK. The minimum number of key components is 2 and the maximum number of key components is 4.

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

<16E#Variant#$E_{MFK.V}$(Key)#Number of Components#Reserved#>

Response

<26E#Key Check Digits#$E_{MFK.VC}$(Comp-1)#Comp-1 Check Digits#
$E_{MFK.VC}$(Comp-2)#Comp-2 Check Digits#[$E_{MFK.VC}$(Comp-3)]#
[Comp-3 Check Digits]#[$E_{MFK.VC}$(Comp-4)]#
[Comp-4 Check Digits]#>[CRLF]

Calling Parameters

16E

Field 0, the command identifier.

Variant

Field 1, the variant applied to the MFK that was used to encrypt the key in field 2. This field must contain a numeric value in the range of 0-31.

$E_{MFK.V}$(Key),MAC

Field 2, the 3DES key to be divided into components. This field must contain either a 16 or 32 hexadecimal character value.

Number of Components

Field 3, the number of key components to divide the key into. The minimum value is 2 and the maximum value is 4.

Reserved

Field 4, this field must be empty.

Table 10-11. Command 16E: Divide a Key into Components

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	16E
1	Variant	1-2	0-31
2	$E_{MFK.V}$(Key)	16, 32	0-9, A-F
3	Number of components	1	2-4
4	Reserved	0	empty

Responding Parameters

`26E`

Field 0, the response identifier.

`Key Check Digits`

Field 1, the check digits of the key. The check digits are the first four digits that result from encrypting zeros using the key. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the key.

$E_{MFK.VC}$`(Comp-1)`

Field 2, the first key component encrypted under the component variant of the MFK. This field will contain either a 16 or 32 hexadecimal character value.

`Comp-1 Check Digits`

Field 3, the check digits of the first key component. The check digits are the first four digits that result from encrypting zeros using the key component. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the key component.

$E_{MFK.VC}$`(Comp-2)`

Field 4, the second key component encrypted under the component variant of the MFK. This field will contain either a 16 or 32 hexadecimal character value.

`Comp-2 Check Digits`

Field 5, the check digits of the second key component. The check digits are the first four digits that result from encrypting zeros using the key component. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the key component.

`[`$E_{MFK.VC}$`(Comp-3)]`

Field 6, the third key component encrypted under the component variant of the MFK. This field will contain either a 16 or 32 hexadecimal character value.

`[Comp-3 Check Digits]`

Field 7, the check digits of the third key component. The check digits are the first four digits that result from encrypting zeros using the key component. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the key component.

`[`$E_{MFK.VC}$`(Comp-4)]`

Field 8, the fourth key component encrypted under the component variant of the MFK. This field will contain either a 16 or 32 hexadecimal character value.

`[Comp-4 Check Digits]`

Field 9, the check digits of the fourth key component. The check digits are the first four digits that result from encrypting zeros using the key component. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the key component.

Table 10-12. Response 26E: Divide a Key into Components

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	26E
1	Key Check Digits	4, 6	0 - 9, A - F
2	$E_{MFK.VC}$(Comp-1)	16, 32	0 - 9, A - F
3	Comp-1 Check Digits	4, 6	0 - 9, A - F
4	$E_{MFK.VC}$(Comp-2)	16, 32	0 - 9, A - F
5	Comp-2 Check Digits	4, 6	0 - 9, A - F
6	[$E_{MFK.VC}$(Comp-3)]	16, 32	0 - 9, A - F
7	[Comp-3 Check Digits]	4, 6	0 - 9, A - F
8	[$E_{MFK.VC}$(Comp-4])	16, 32	0 - 9, A - F
9	[Comp-4 Check Digits]	4, 6	0 - 9, A - F

Usage Notes

- Randomly generated key components are not adjusted to odd parity.

Example

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Divide a 2key-3DES PIN Verification key into 2 components

PIN Verification Key (KPV): 1234123412341234 5678567856785678
The PIN Verification Key encrypted under variant 4 of the MFK:
2979F6551D0084AC4B2EF58A726348FE

The command looks like this:

```
<16E#4#2979F6551D0084AC4B2EF58A726348FE#2##>
```

The Network Security Processor generates random component values, the response will be similar to this:

```
<26E#DB82#3CEDD01E5BAF971E2A3EA4B8C35FB14A#F066#8259E4446A0A4
0C45410018A00B2E9C5#6BD7#####>
```

Print Component Letter (Command 16F)

Command 16F is used to print a component letter for an existing component or a randomly generated component. This command supports 3DES key components.

▲ **WARNING.** The print job will contain the cleartext component. Appropriate security measures are required to ensure that only authorized personnel have access to the printer, and that communications between the Ax160 NSP and the printer are not monitored.

The host application creates a component letter template (as a standard ASCII text file or Microsoft Word document). The component letter template must contain both a component marker string and a check digits marker string. These marker strings indicate where the cleartext component and check digits will be inserted into the template. An optional reference marker string is supported.

The host application uses this command to send the component letter template to the Ax160 NSP along with the encrypted component, or it can instruct the Ax160 NSP to generate a random component. The Ax160 NSP decrypts or generates the component, searches the component letter template for the component and check digit marker strings and then replaces them with the cleartext component and check digit values, and optionally the reference value. The Ax160 NSP then sends the component letter print job to the printer.

To reduce the component letter template size, company logos and other graphics should be preprinted on the paper that is loaded into the printer.

This command is not enabled in the Ax160 NSP's default security policy. It is only allowed on the NIC1 print command port. **It is highly recommended that this command be enabled for a specific number of executions.** For information on how to configure the Ax160 NSP to limit how many times this command can be executed refer to the Command Count feature which is documented in section 4 of the

.

Command

```
<16F#Letter Type#Continuation Flag#[Continuation Index]#
[Variant]#[E_MFK.VC(Component)]#[Component Length]#
[Component Marker String]#[Check Digit Marker String]#
[Reference Marker String]#Data Encoding#Data Type#
Letter Template Size#Data Block Length#Data Block#>
```

Response

<26F#[Continuation Index]#[$E_{MFK.VC}$(Component)]#
Component Check Digits#>[CRLF]

Calling Parameters

16F

Field 0, the command identifier.

Letter Type

Field 1, this field is specifies the type of letter to be printed.

Specify a letter type of 0 (zero) to print a test page. The following restrictions apply to printing a test page: field 2 must be contain a 0, fields 3 through 9 must be empty, and field 10 must be contain the letter A.

To print a component letter specify a letter type value of 1.

Continuation Flag

Field 2, the continuation flag. The table below defines the allowed values.

Value	Description
0	Entire component letter template is included in this command.
1	The command contains the first block of a multi-block component letter template.
2	The command contains an intermediate block of a multi-block component letter template.
3	The command contains the final block of a multi-block component letter template.
4	Cancel current print job; removes a partial component letter template from Ax160 NSP's memory.

[Continuation Index]

Field 3, this index specifies which of the four internal memory storage locations the Ax160 NSP used to store the first component letter template data block.This field must be empty when the continuation flag (field 2) is set to a value of 0 or 1. This field must be empty if the command is used to send the first data block of the component letter template. For subsequent commands used to send intermediate and final data blocks the value of this field must match the value returned in field 1 of the response to the command that was used to send the first data block of the component letter template. When the continuation flag (field 2) is set to a value of 2, 3, or 4, this field can contain the values 0, 1, 2, or 3.

[Variant]

Field 4, the variant applied to MFK when it encrypts the generated component. If present, this field must contain a value in the range of 0C - 31C. The letter C indicates that this is a component and not a key. For example, to generate a random component for a CVV key which is encrypted under variant 3 of the MFK, this field would contain the value 3C.This field is ignored when the continuation flag (field 2) contains a 1, 2, or 4.

[$E_{MFK.VC}$(Component)]

Field 5, the key component encrypted under the component variant of the MFK. If present, this field can contain 16 or 32 hexadecimal characters. The Ax160 NSP will generate a random component when this field is empty and the continuation flag (field 2) contains a value of 0 or 3. This field is ignored when the continuation flag (field 2) contains a 1, 2, or 4.

[Component Length]

Field 6, the length of the component to be generated by the Ax160 NSP. The random component value will be adjusted odd parity. This field can contain one of these values:

Value	Description
S	1key-3DES key (single-length)
D	2key-3DES key (double-length)

This field must be empty when field 5 contains a component. This field is ignored when the continuation flag (field 2) contains a 1, 2, or 4.

[Component Marker String]

Field 7, the component marker string in the letter template file that identifies the location where the cleartext component will be printed. The component marker string is 19 characters it represents 16 characters of the component with spaces between each set of 4 characters. This field can contain upper and lower case letters (A-Z, a-z) and numeric digits (0-9). This field will be ignored if the continuation flag (field 2) is 1, 2 or 4.

[Check Digit Marker String]

Field 8, the marker character string in the letter template file that identifies the location where the check digits will be printed. This field can contain upper and lower case letters (A-Z, a-z) and numeric digits (0-9). The contents of the check digit marker string are arbitrary, but the length must be the same as the length of the check digits. For a 3DES key component, the length of the check digits will be 4 if option 88 is not enabled, 6 if option 88 is enabled. This field will be ignored if the continuation flag (field 2) is 1, 2 or 4.

`[Reference Marker String]`

Field 9, the marker character string in the letter template file that identifies the location where the reference value will be printed. The contents of the reference marker string are arbitrary. If present, this field must be 19 characters and can contain upper and lower case letters (A-Z, a-z) and numeric digits (0-9). The reference value that will be printed in the component letter is the leftmost 16 characters of the component cryptogram. The reference value can be included to help organizations match the component letter with the encrypted component on the host application's database. Note that printing the reference value is optional, if the reference marker string is not included in the letter template file, the reference value will not be printed. This field will be ignored if the continuation flag (field 2) is 1, 2 or 4.

`Data Encoding`

Field 10, the encoding used for the component, check digits and reference marker strings in the letter template file. This field can contain one of these values:

Value	Description
A	ASCII encoding, where 1234 = 0x31323334.
W	Windows encoding (16-char, little endian), where 1234 = 0x3100320033003400.

`Data Type`

Field 11, only binary is supported. This field must contain the letter B.

`Letter Template Size`

Field 12, the size of the complete component letter template file. The maximum size of the component letter template file is 1,048,576 bytes (1 megabyte).

`Block Data Length`

Field 13, the length of the data sent in this data block. The maximum value is 30000.

`Data Block`

Field 14, the binary data of the component letter template file. The maximum amount of binary data is 30000 bytes.

Table 10-13. Command 16F: Print Component Letter (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	16F
1	Letter Type	1	0,1
2	Continuation Flag	1	0-4
3	[Continuation Index]	0,1	0-3

Table 10-13. Command 16F: Print Component Letter (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
4	[Variant]	0,2-3	printable ASCII
5	$E_{MFK.VC}$(Component)]	0, 16, 32	printable ASCII
6	[Component Length]	0,1	S,D
7	[Component Marker String]	0, 19	0-9, A-Z, a-z,
8	[Check Digits Marker String]	0, 4, 6	0-9, A-Z, a-z
9	[Reference Marker String]	0, 19	0-9, A-Z, a-z
10	Data Encoding	1	A, W
11	Data Type	1	B
12	Letter Template Size	1-7	0-1048576
13	Block Data Length	1-5	0-30000
14	Data Block	1-30000	binary

Responding Parameters

`26F`

Field 0, the response identifier.

`[Continuation Index]`

Field 1, this field will match field 3 of the command if the continuation flag (command-field 2) is 2, 3, or 4. It will be empty if the continuation flag is 0 or 1.

`[`$E_{MFK.VC}$`(Component)]`

Field 2, the component generated by the Ax160 NSP encrypted under the component variant (specified in field 4 of the command) of the MFK. This field will contain a 16 or 32 hexadecimal character value. This field will be empty when a component is provided in field 5 of the command.

`Component Check Digits`

Field 3, the check digits of the key component.The check digits are the first four digits that result from encrypting zeros using the key component. If option 88 is enabled this field will contain the first six digits of the result from encrypting zeros using the key component.

Table 10-14. Response 26F: Print Component Letter

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	26F
1	[Continuation Index]	0,1	0-3
2	[$E_{MFK.VC}$(Component)]	0, 16, 32	0 - 9, A - F
3	Component Check Digits	4, 6	0 - 9, A - F

Usage Notes

When a letter template file requires multiple commands to process the entire letter template file, the Ax160 NSP will clear the entire letter template file from its memory on any of the error conditions listed below. In this case correct the error and send all of the commands required to process the entire letter template file again.

- Invalid letter template length - the total number of bytes received is not equal to the total number of bytes specified in the field 12 of the command.
- Invalid component specified in field 5.The Ax160 NSP will return an error code 07.
- TCP/IP connection or send/receive error is detected. The Ax160 NSP will return an error code 11.
- Ax160 NSP execution error. The Ax160 NSP will return an error code 08.
- Cannot find the marker string in the document. The Ax160 NSP will return an error code 12.

Command syntax errors, such as invalid number of fields in a command or invalid character in a command, will not cause the Ax160 NSP to erase the letter template file from its internal memory slot. In this case correct the syntax error and send the command again to print the component letter.

Example

The 2key-3DES Master File Key is:
2ABC 3DEF 4567 0189 9810 7645 FED3 CBA2, check digits = 057A. See 2key-3DES Key (Double-Length) on page A-5 for component values.

Generate a random Card Verification Value Key 2key-3DES component

- Variant: 3C
- Component Marker String: 1234567890123456789
- Check Digit Marker String: zzzz
- Reference Value Marker String: xxxxxxxxxxxxxxxxxx

ASCII Component Letter Template Text

```
Cleartext 2Key-3DES Key Component


 Block 1: 1234567890123456789
 Block 2: 1234567890123456789


Check Digits: zzzz


Reference Number: xxxxxxxxxxxxxxxxxx
```

The command looks like this. For visibility purposes in this example the binary data in the data block field (field 14) is presented in hexadecimal format.

```
<16F#1#0##3C##D#123456789012345678 9#zzzz#xxxxxxxxxxxxxxxxxx#
A#B#162#162#436C65617274657874203 24B65792D33444553204B6579204
36F6D706F6E656E740D0A0D0A20426C6F 636B20313A203132333435363738
393031323334353637383 90D0A20426C6F636B20323A20313233343536373
839303132333435363738390D0A0D0A 436865636B204469676974733A207A
7A7A7A0D0A0D0A5265666572656E6365 204E756D6265723A2078787878787
878787878787878787878787878780D 0A#>
```

The Network Security Processor generates a random component value, the response will be similar to this:

```
<26F##8C000382F8593B90EAFB7D1D2AEE6025#09D9#>
```

The cleartext generated key component is:

```
13B5 5EEA 2083 B658 E34F 61BC ABF1 19C2, check digits = 09D9
```

11 Utility Commands

This section describes the commands to, test the communications link between the host and the Network Security Processor, configure the Network Security Processor, and obtain a variety of operating information about the Network Security Processor. The term “Security Processor” refers to the Network Security Processor.

All commands in this section are enabled in the Network Security Processor’s default factory security policy.

Quick Reference

Table 11-1 identifies the utility commands.

Table 11-1. Utility Commands (page 1 of 3)

Command #	Name	Purpose
00	Echo	Tests the communications link between the host and the security processor.
9A#CLEAR_LOG	Clear Log	Clears the system log
9A#CONFIG-Request	Security Processor Configuration Status	Returns which commands are enabled and disabled.
9A#COUNT	Security Processor Count Status	Returns the commands and counter value for the commands in the security processor that are being counted.
9A#DIAGTEST	Security Processor Crypto Test	Returns the result of the cryptographic test.
9A#KEY	Security Processor Status Key	Returns the security processor's current key information.
9A#ID	Security Processor Status ID	Returns the commands and options that are enabled in the security processor.
101	Configure Security Processor Options	Enables or disables specific operating parameters.
102	Command Monitoring	Counts the number of PIN, sanity, CVV/CVC/CSC, and MAC verification failures that have been processed. It can also count the number of times an enabled command has been processed.

Table 11-1. Utility Commands (page 2 of 3)

Command #	Name	Purpose
105	Configure Premium Value Commands and Options	This command will be supplied by Atalla when a premium value command or option has been purchased.
106	Define Temporary Serial Number	Allows you to define a temporary serial number.
107	Confirm Temporary Serial Number	Activates the temporary serial number.
108	Define Security Policy	Allows you to define which commands and options will be enabled or disabled.
109	Confirm Security Policy	Activates the defined security policy.
1101	Get Image ID	Returns the image version information of the cryptographic command processor.
1102	Get Virtual NSP Information	Returns the number of the virtual NSP that the host application is connected to, the name of the virtual NSP, and number of virtual NSPs defined.
1104	Get Temporary Serial Number Information	Returns the temporary serial number and the number of hours remaining before it expires.
1105	Configure Premium Value Commands and Options in all Virtual NSPs	Enables/disables premium value commands and options in all virtual NSPs.
1110	Get System Configuration Information	Returns the version information of all components in the Network Security Processor.
1111	Get Date and Time	Returns the Network Security Processor system date and time in Universal Coordinated Time
1113	Get Average CPU Utilization	Returns a percentage value which is the average CPU utilization for the Network Security Processor.
1120	Get System Information	Returns the NSP serial number, product ID, system software information, and a personality version field.
1204	Get Log Signing Key Certificate	Returns the certificate of the key used to sign the system and virtual NSP logs.

Table 11-1. Utility Commands (page 3 of 3)

Command #	Name	Purpose
1216	Get Battery Life Remaining	Returns the number of days remaining before the battery expiration messages start appearing in the log.
1221	Return IP Address of the Network Security Processor.	Returns the IP Address of the Network Security Processor.
1223	TCP/IP Socket Information	Returns information on the number of TCP/IP sockets available on the Network Security Processor.
1226	Get Check Digits	Returns check digits of keys in the non-volatile key table.
1227	Reset to Factory State Part 1	Used to reset the Network Security Processor to factory state. This command must be sent to the SCA port.
1228	Reset to Factory State Part 2	Used to reset the Network Security Processor to factory state. This command must be sent to the SCA port.
1350	Select Virtual NSP	Use this command to select which virtual NSP will process the commands sent from the USB or serial port.
1351	Virtual NSP System Information	Use this command to return the name if defined, and Master File Key check digits (MFKCD) for each defined virtual NSP.

Echo Test Message (Command 00)

Command 00 can be used to test the communications link between the host and the Network Security Processor. A value of 00 is returned in response to the command, or in response to an error condition in another command. See Error responses on page 1-3 for information on error codes.

Command

```
<00#Message#>
```

Response

```
<00#0000Revision Level#Message#>[CRLF]
```

Calling Parameters

`00`

Field 0, the command identifier.

`Message`

Field 1, the test message to be echoed in the response. This message can be from one to 1999 bytes long and can contain any character or number except “#”, “>”, and “<”.

Table 11-2. Command 00: Echo Test Message

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	00
1	Message	1 - 2000	Any except #, <, >

Responding Parameters

`00`

Field 0, the response identifier.

`0000Revision Level`

Field 1, the software revision level.

`Message`

Field 2, the message sent in the command is returned. This field is from one to 2000 bytes long and can contain any character or number, except “#”, “>”, and “<”.

Table 11-3. Response 00: Echo Test Message

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	00
1	0000Revision level	6	0 - 9
2	Message	1 - 2000	Any except #, <, >

Example

The following examples illustrate Command 00 used to echo a message. The security processor returns both the message and the software's revision number (in this case, the revision number is 2.8).

The command looks like this:

```
<00#This is a test.#>
```

The Network Security Processor returns the following response:

```
<00#000028#This is a test.#>
```

Security Processor Clear Log (Command 9A)

Command 9A – This command closes the current system log on the USB flash memory device, clears the system log that is stored in memory, and then uses the current data and time to create a new system log on the USB flash memory device.

Command

```
<9A#CLEAR_LOG#>
```

Response

```
<AA#Status#>[CRLF]
```

Calling Parameters

`9A`

Field 0, the command identifier.

`CLEAR_LOG`

Field 1, the request to the security processor to clear the system log.

Table 11-4. Command 9A: Security Processor CLEAR_LOG

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	2	9A
1	CLEAR_LOG	9	CLEAR_LOG

Responding Parameters

`AA`

Field 0, the response identifier.

`Status`

Field 1, there are two possible status values:

DONE - confirmation that the system log has been cleared.

LOG DOES NOT EXIST - indicates an error clearing the system log.

Table 11-5. Response AA: Security Processor CLEAR LOG

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AA
1	Status	4 or 18	DONE, LOG DOES NOT EXIST

Usage Notes

In certain situations, such as when option 44 is enabled, the amount of command/response data can exceed the capacity of the system log. When this situation occurs, no new system log information can be recorded. Use the <9A#CLEAR_ LOG#> command to the clear the system log and create a new system log.

Example

This example illustrates sending Command 9A to clear the Network Security Processor's system log.

The command looks like this:

```
<9A#CLEAR_LOG#>
```

The response looks similar to this.

```
<AA#DONE#>
```

Security Processor Configuration Status (Command 9A)

Command 9A – Security Processor Configuration Status returns a list of enabled or disabled commands and options with a high security exposure, followed by a list of enabled or disabled commands with a low security exposure. It also returns the sequence number and serial number of the Network Security Processor. Use this commands to confirm that the Network Security Processor's security policy has been implemented correctly.

Some premium value commands and options listed in the response were developed for specific customers. For privacy and security reasons, they are not documented in this manual.

Command

```
<9A#CONFIG-Request#>
```

Response

```
<AA#Serial Number#Commands/Options with High Security
Exposure#Commands/Options with Low Security Exposure#
Sequence Number#>[CRLF]
```

Calling Parameters

`9A`

Field 0, the command identifier.

`CONFIG-Request`

Field 1, the request to the security processor for a list of commands and options. There are two possible values:

CONFIG-ON instructs the security processor to return in field 3, a list of commands and options that have a high security exposure. Field 4 will contain a list of commands and options that have a low security exposure.

CONFIG-OFF instructs the security processor to return in field 3, a list of commands and options that have a high security exposure. Field 4 will contain a list of commands and options that have a low security exposure. Note: there are some undocumented Atalla custom commands that may appear in the list.

CONFIG-ALL instructs the security processor to return a list of commands and options included in the Network Security Processor regardless of their on/off status. Commands and options that have a high security exposure are listed in

field 3. Commands and options that have a low security exposure are listed in field 4.

Table 11-6. Command 9A: Security Processor Configuration Status

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	2	9A
1	Enabled Commands request identifier.	9 or 10	CONFIG-ON, CONFIG-OFF, CONFIG-ALL

Responding Parameters

`AA`

Field 0, the response identifier.

`Serial Number`

Field 1, the serial number of the Network Security Processor.

`CONFIG Request`

Field 2, the Configuration request.

If the value is CONFIG-ON, fields three and four of the response will contain the list of enabled commands and options.

If the value is CONFIG-OFF, fields three and four of the response will contain the list of disabled commands and options.

If the value is CONFIG-ALL, fields three and four of the response will contain the list of all commands and options contained in the Network Security Processor.

`Commands and Options with High Security Exposure`

Field 3, the list of commands and options that have a high security exposure.

`Commands and Options with Low Security Exposure`

Field 4, the list of commands and options that have a low security exposure.

`Sequence Number`

Field 4, the number of times the security policy has been updated.

Table 11-7. Response AA: Security Processor Configuration Status

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AA
1	Serial Number	7	ASCII
2	CONFIG Request	9, or 10	CONFIG-ON, CONFIG-OFF, CONFIG-ALL
3	Commands and Options with a High Security Exposure	variable	ASCII
4	Commands and Options with a Low Security Exposure	variable	ASCII
5	Sequence Number	16	0-9

Usage Notes

You can send Command 9A to the security processor after the security policy has been implemented to confirm that correct commands and options are enabled or disabled.

When multiple virtual NSP support is enabled in the Network Security Processor, option 87 will only be included in the response when command <9A#CONFIG-ON#> is sent to VNSP0.

Examples

This example illustrates sending Command 9A and receiving list of **enabled** commands and options.

The command looks like this:

```
<9A#CONFIG-ON#>
```

The response looks similar to this.

```
<AA#JL0205#CONFIG-ON#(62),(63),(A0)="4",(A1)="S",(A2)="S"#
00,10,11,12,13,17,31,32,5C,5E,70,71,72,73,74,7E,7F,93,99,9A,
9B,9C,9E,9F,101,105,106,107,108,109,113,335,348,350,352,354,
356,357,359,35A,35F,36A#0000000000000001#>
```

This example illustrates sending Command 9A and receiving list of **disabled** commands and options.

The command looks like this:

```
<9A#CONFIG-OFF#>
```

The response looks similar to this.

```
<AA#JL0205#CONFIG-OFF#14,15,16,18,19,1A,1C,1D,1E,1F,30,33,34,
35,36,37,38,39,3A,3D,3F,55,58,59,5D,5F,75,76,77,78,79,7A,7B,
90,94,95,96,97,98,B1,B2,B3,B4,B5,B6,B7,BA,BB,BC,BD,BE,BF,D0,
D1,D2,D3,D4,D5,D6,D7,D8,D9,DA,102,110,111,112,114,115,11D,
11E,15E,160,161,162,163,16E,16F,301,302,306,307,308,309,30A,
30B,30C,30D,30E,30F,319,31A,31B,31C,31D,31E,31F,321,32A,32B,
32C,332,333,334,336,337,338,339,33A,33B,33C,33D,33E,33F,349,
34A,34B,34C,34D,34E,34F,351,35B,35C,35E,360,361,362,363,364,
370,371,372,37A,37B,381,382,386,388,3A1,3A2,3A3,3A4,3B2,3B3,
3B4,3B5,3EA,3FA,(46),(47),(48),(49),(4B),(4C),(4D),(4E),(4F),
(60),(61),(64),(65),(66),(68),(69),(6A),(6B),(6C),(6E),(6F),
(80),(81),(82),(83),(84),(87),(88),(89),(8A),(8B),(8D)#(20),
(21),(23),(27),(44)#0000000000000001#>
```

This example illustrates sending Command 9A and receiving list of all commands and options in the Ax160-NSP.

```
<9A#CONFIG-ALL#>
```

The response looks similar to this.

```
<AA#JL0205#CONFIG-ALL#14,15,16,18,19,1A,1C,1D,1E,1F,30,33,34,
35,36,37,38,39,3A,3D,3F,55,58,59,5D,5F,75,76,77,78,79,7A,7B,
90,94,95,96,97,98,B1,B2,B3,B4,B5,B6,B7,BA,BB,BC,BD,BE,BF,D0,
D1,D2,D3,D4,D5,D6,D7,D8,D9,DA,102,110,111,112,114,115,11D,
11E,15E,160,161,162,163,16E,16F,301,302,306,307,308,309,30A,
30B,30C,30D,30E,30F,319,31A,31B,31C,31D,31E,31F,321,32A,32B,
32C,332,333,334,336,337,338,339,33A,33B,33C,33D,33E,33F,349,
34A,34B,34C,34D,34E,34F,351,35B,35C,35E,360,361,362,363,364,
370,371,372,37A,37B,381,382,386,388,3A1,3A2,3A3,3A4,3B2,3B3,
3B4,3B5,3EA,3FA,(46),(47),(48),(49),(4B),(4C),(4D),(4E),(4F),
(60),(61),(62),(63),(64),(65),(66),(68),(69),(6A),(6B),(6C),
(6E),(6F),(80),(81),(82),(83),(84),(87),(88),(89),(8A),(8B),
(8D),(A0)="4",(A1)="S",(A2)="S"#00,10,11,12,13,17,31,32,5C,
5E,70,71,72,73,74,7E,7F,93,99,9A,9B,9C,9E,9F,101,105,106,107,
108,109,113,335,348,350,352,354,356,357,359,35A,35F,36A,(20),
(21),(23),(27),(44)#0000000000000001#>
```

Security Processor Count Status (Command 9A)

Command 9A – Security Processor Count Status returns a list of commands that are being counted along with the current count value (in decimal). Each time the Network Security Processor successfully processes a command that is being counted the counter value is decremented by 1. Commands that are not successfully processed by the Network Security Processor, such as commands that contain syntax error(s) that result in an error response are not counted.

△ **Caution.** Once the counter value reaches zero, the Network Security Processor will return an error <00#0300xx#> instead of processing the command.

A maximum of nine cryptographic commands can be counted. Utility commands and options cannot be counted. See command 108 for instructions on setting a count value for a command.

This command can be used to determine the number of commands processed by the Network Security Processor.

Command

```
<9A#COUNT#>
```

Response

```
<AA#Serial Number#Reserved#[Command-Counter#]
[Command-Counter#][Command-Counter#][Command-Counter#]
[Command-Counter#][Command-Counter#][Command-Counter#]
[Command-Counter#][Command-Counter#]>[CRLF]
```

Calling Parameters

`9A`

Field 0, the command identifier.

`COUNT`

Field 1, the request to the security processor for a list of commands that have been enabled for counting.

Table 11-8. Command 9A: Security Processor Count Status

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier.	2	9A
1	Function	5	COUNT

Responding Parameters

`AA`

Field 0, the response identifier.

`Serial Number`

Field 1, the serial number of the Network Security Processor.

`Reserved`

Field 2, this field is reserved for future use.

`[Command-Count#]`

Field 3, the command being counted followed by the current counter value. This field is present only if there is at least one command being counted.

`[Command-Count#]`

Field 4, the command being counted followed by the current counter value. This field is present only if there is at least two command being counted.

`[Command-Count#]`

Field 5, the command being counted followed by the current counter value. This field is present only if there is at least three command being counted.

`[Command-Count#]`

Field 6, the command being counted followed by the current counter value. This field is present only if there is at least four command being counted.

`[Command-Count#]`

Field 7, the command being counted followed by the current counter value. This field is present only if there is at least five command being counted.

`[Command-Count#]`

Field 8, the command being counted followed by the current counter value. This field is present only if there is at least six command being counted.

`[Command-Count#]`

Field 9, the command being counted followed by the current counter value. This field is present only if there is at least seven command being counted.

`[Command-Count#]`

Field 10, the command being counted followed by the current counter value. This field is present only if there is at least eight command being counted.

`[Command-Count#]`

Field 11, the command being counted followed by the current counter value. This field is present only if there are nine command being counted.

Table 11-9. Response AA: Security Processor Count Status

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AA
1	Serial Number	7	ASCII
2	Reserved	1	0-9
3	[Command-Count#]	15	0-9, A-F
4	[Command-Count#]	15	0-9, A-F
5	[Command-Count#]	15	0-9, A-F
6	[Command-Count#]	15	0-9, A-F
7	[Command-Count#]	15	0-9, A-F
8	[Command-Count#]	15	0-9, A-F
9	[Command-Count#]	15	0-9, A-F
10	[Command-Count#]	15	0-9, A-F
11	[Command-Count#]	15	0-9, A-F

Usage Notes

You can send Command 9A to the security processor to determine the number of commands processed.

Example

The command looks like this:

```
<9A#COUNT#>
```

The response looks similar to this. Field 3 of the response shows that command 10 is counted and the current counter value is 50. FIeld 4 of the response shows that command 11 is being counted and the current counter value is 40.

```
<AA#D126XL#1#0010-0000000050#0011-0000000040#>
```

Security Processor Crypto Test (Command 9A)

Command 9A – This command performs a cryptographic test.

Command

```
<9A#DIAGTEST#Algorith#RSA Option#>
```

Response

```
<AA#Result#>[CRLF]
```

Calling Parameters

`9A`

Field 0, the command identifier.

`DIAGTEST`

Field 1, the request to the security processor to perform the cryptographic test.

`Algorithm`

Field 2, the algorithm test to be performed.

Algorithm	Test Description
0	Perform all tests
1*	3DES
2*	Deterministic Random Bit Generator
3*	RSA encryption/decryption, signature generation/verification
4*	MD5
5*	SHA-1
6*	SHA-256
7	Personality and Kernel Integrity
8*	AES 128-, 192-, and 256-bit CMAC AES-256 CBC mode encryption/decryption Known Answer Test for AES 128, 192, and 256 CBC mode
9*	HMAC_SHA256

* Known Answer Test (KAT) is performed for this algorithm.

`RSA Option`

Field 3, determines if the RSA test will be performed when field 2 contains the number 0 (zero). The RSA test will be performed when field 2 of the command

contains the number 0 (zero) and this field contains a value in the range of 1 through 9. This field is ignored when the value of field 2 is not equal to 0 (zero).

Table 11-10. Command 9A: Security Processor Crypto Test

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	9A
1	DIAGTEST	8	DIAGTEST
2	Algorithm	1	0 - 8
3	RSA Option	0,1	0-9

Responding Parameters

`AA`

Field 0, the response identifier.

`Results`

Field 1, the result of the test. A response of "OK" means the test completed successfully. Any other response indicates a failure.

Table 11-11. Response AA: Security Processor Crypto Test

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AA
1	Result	2, varies	OK, A-Z

Usage Notes

- The RSA algorithm test can take up to 10 seconds to return a response. All other tests complete in less than 1 second.
- Test results are recorded in the system log or virtual NSP log. If a test fails the NSP must be power cycled to clear the condition. If it fails again the Network Security Processor must be replaced.

Example

Test all algorithms

The command looks like this:

```
<9A#DIAGTEST#0#1#>
```

When the test completes successfully, the Network Security Processor returns this response.

```
<AA#OK#>
```

Security Processor Status ID (Command 9A)

Command 9A – Security Processor Status ID returns the security processor's current configuration and serial number. Use this command to monitor the configuration of the Network Security Processor, to ensure that only authorized commands and options are enabled.

This command does not return status of four digit utility commands, nor does it highlight high security exposure commands or options, see Security Processor Configuration Status (Command 9A) on page 11-8, for more information on obtaining commands listed by security exposure.

Command

```
<9A#ID#>
```

Response

```
<AA#Serial No.#Type[,Currently Enabled Options]#
Minimum PIN Length,PIN Length Character,
DUKPT Session Key Length#[Enabled 0X Commands]#
[Enabled 1X Commands]#[Enabled 3X Commands]#
[Enabled 5X Commands]#[Enabled 7X Commands]#
[Enabled 9X Commands]#[Enabled BX Commands]#
[Enabled DX Commands]#[Enabled 10X Commands]#
[Enabled 11X Commands]#[Enabled 15X Commands]#
[Enabled 16X Commands]#[Enabled 30X Commands]#
[Enabled 31X Commands]#[Enabled 32X Commands]#
[Enabled 33X Commands]#[Enabled 34X Commands]#
[Enabled 35X Commands]#[Enabled 36X Commands]#
[Enabled 37X Commands]#[Enabled 38X Commands]#
[Enabled 3AX Commands]#[Enabled 3BX Commands]#
[Enabled 3EX Commands]#[Enabled 3FX Commands]#>[CRLF]
```

Calling Parameters

`9A`

Field 0, the command identifier.

`ID`

Field 1, the request to the security processor for current configuration information.

Responding Parameters

`AA`

Field 0, the response identifier.

`Serial No.`

Field 1, the factory-assigned serial number. This field is six bytes long and contains ASCII characters.

`Type[,Currently Enabled Options]`

Field 2, Product identification, followed by all currently enabled options. This field's length depends on the security processor’s configuration. This field can contain the numbers 0 through 9,

`[Enabled 3X Commands]`

Field 6, a listing of the 3X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 3X commands have been configured for use, then this field is empty.

`[Enabled 5X Commands]`

Field 7, a listing of the 5X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 5X commands have been configured for use, then this field is empty.

`[Enabled 7X Commands]`

Field 8, a listing of the 7X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 7X commands have been configured for use, then this field is empty.

`[Enabled 9X Commands]`

Field 9, a listing of the 9X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 9X commands have been configured for use, then this field is empty.

`[Enabled BX Commands]`

Field 10, a listing of the BX commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the BX commands have been configured for use, then this field is empty.

`[Enabled DX Commands]`

Field 11, a listing of the DX commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the DX commands have been configured for use, then this field is empty.

`[Enabled 10X Commands]`

Field 12, a listing of the 10X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 10X commands have been configured for use, then this field is empty.

`[Enabled 11X Commands]`

Field 13, a listing of the 11X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 11X commands have been configured for use, then this field is empty.

`[Enabled 15X Commands]`

Field 14, a listing of the 15X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 15X commands have been configured for use, then this field is empty.

`[Enabled 16X Commands]`

Field 15, a listing of the 16X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 16X commands have been configured for use, then this field is empty.

`[Enabled 30X Commands]`

Field 16, a listing of the 30X commands that have been configured for use. This field's length depends on the Network Security Processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 30X commands have been configured for use, then this field is empty.

`[Enabled 31X Commands]`

Field 17, a listing of the 31X commands that have been configured for use. This field's length depends on the Network Security Processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 31X commands have been configured for use, then this field is empty.

`[Enabled 32X Commands]`

Field 18, a listing of the 32X commands that have been configured for use. This field's length depends on the Network Security Processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 32X commands have been configured for use, then this field is empty.

`[Enabled 33X Commands]`

Field 19, a listing of the 33X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 33X commands have been configured for use, then this field is empty.

`[Enabled 34X Commands]`

Field 20, a listing of the 34X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 34X commands have been configured for use, then this field is empty.

`[Enabled 35X Commands]`

Field 21, a listing of the 35X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 35X commands have been configured for use, then this field is empty.

`[Enabled 36X Commands]`

Field 22, a listing of the 36X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 36X commands have been configured for use, then this field is empty.

`[Enabled 37X Commands]`

Field 23, a listing of the 37X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 37X commands have been configured for use, then this field is empty.

`[Enabled 38X Commands]`

Field 24 a listing of the 38X commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 38X commands have been configured for use, then this field is empty.

`[Enabled 3AX Commands]`

Field 25, a listing of the 3AX commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 3AX commands have been configured for use, then this field is empty.

`[Enabled 3BX Commands]`

Field 26, a listing of the 3BX commands that have been configured for use. This field's length depends on the security processor's configuration. It can contain the characters 0 through 9, A to Z, and ",". If none of the 3BX commands have been configured for use, then this field is empty.

`[Enabled 3EX Commands]`

Field 27, a listing of the 3EX commands that have been configured for use. This field's length depends on the security processor’s configuration. It can contain the characters 0 through 9, A to Z, and “,”. If none of the 3EX comm

Table 11-13. Response AA: Security Processor Status ID (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
23	[Enabled 37X commands]	0 - *	0 - 9, A - F, “,”
24	[Enabled 38X commands]	0 - *	0 - 9, A - F, “,”
25	[Enabled 3AX commands]	0 - *	0 - 9, A - F, “,”
26	[Enabled 3BX commands]	0 - *	0 - 9, A - F, “,”
27	[Enabled 3EX commands]	0 - *	0 - 9, A - F, “,”
28	[Enabled 3FX commands]	0 - *	0 - 9, A - F, “,”

*Length varies.

Usage Notes

- There are some undocumented commands that may appear in the list. They are customer specific commands and are not generally available.
- When multiple virtual NSP support is enabled in the Network Security Processor, option 87 will only be included in the response when this command is sent to VNSP0.

Example

Obtain a list of enabled commands and options.

The command looks like this:

```
<9A#ID#>
```

The security processor issues a response that contains the following information:

- Serial number: JL012S
- Device Type: A10160V
- Options enabled: 62, 63
- Minimum PIN length: 4
- PIN Sanity error: S
- DUKPT session key length is Single: S
- Enabled 0X commands: 00
- Enabled 1X commands: 10,11,12,13,17
- Enabled 3X commands: 31, 32
- Enabled 5X commands: 5C, 5E
- Enabled 7X commands: 70, 71, 72, 73, 74, 7E, 7F
- Enabled 9X commands: 93, 99, 9A, 9B, 9C, 9E, 9F

- Enabled 10X commands: 101, 105, 106, 107, 108, 109
- Enabled 11X commands: 113
- Enabled 33X commands: 335
- Enabled 34x commands: 348
- Enabled 35X commands: 350, 352, 354, 356, 357, 359, 35A, 35F
- Enabled 36A commands: 36A

The Network Security Processor returns a response similar to this.

```
<AA#JL012S#A10160V,62,63#4,S,S#00#10,11,12,13,17#31,32#5C,5E#
70,71,72,73,74,7E,7F#93,99,9A,9B,9C,9E,9F###101,105,106,107,
108,109#113########335#348#350,352,354,356,357,359,35A,35F#
36A########>
```

Security Processor Status Key (Command 9A)

Command 9A – Security Processor Status Key returns the number of available key slots in the volatile table, as well as the check digits of keys stored in the security processor's non-volatile key table.

Command

```
<9A#KEY#>
```

Response

```
<AA#Remaining Slots#[MFK Name]#[MFK Check Digits]#
[MFK Length]#[Pending MFK Name]#[Pending MFK Check Digits]#
[Pending MFK Length]#[Retired MFK Name]#
[Retired MFK Check Digits]#[Retired MFK Length]#
[KEK Check Digits]#[KEK Length]#Reserved#>[CRLF]
```

Calling Parameters

`9A`

Field 0, the command identifier.

`KEY`

Field 1, the request to the security processor for current key information.

Table 11-14. Command 9A: Security Processor Status Key

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	2	9A
1	Key-information request identifier	3	KEY

Responding Parameters

`AA`

Field 0, the response identifier.

`Remaining Slots`

Field 1, the number of available slots in the key table. This field contains a 4 byte decimal value.

`[MFK Name]`

Field 2, the Master File Key's name, MFK1. This field is empty if a Master File Key does not exist or if it does not have a name.

`[MFK Check Digits]`

Field 3, the Master File Key's check digits. This field contains a 4 byte hexadecimal value. This field is empty if a Master File Key does not exist.

`[MFK Length]`

Field 4, the Master File Key's length. This field returns a D to indicate that the Master File Key is 2key-3DES (double-length). This field is empty if a master file key does not exist.

`[Pending MFK Name]`

Field 5, the pending Master File Key's name, PMFK1. This field is empty if a pending Master File Key does not exist or if it does not have a name.

`[Pending MFK Check Digits]`

Field 6, the pending Master File Key's check digits. This field is a 4 byte hexadecimal value. This field is empty if a pending Master File Key does not exist.

`[Pending MFK Length]`

Field 7, the pending Master File Key’s length. This field returns a D to indicate that the Master File Key is 2key-3DES (double-length). This field is empty if a pending Master File Key does not exist.

`[Retired MFK Name]`

Field 8, the retired Master File Key's name. This field will contain the name of the retired Master File Key. This field is empty if a retired Master File Key does not exist.

`[Retired MFK Check Digits]`

Field 9, the retired Master File Key's check digits. This field will contain the check digits of the retired Master File Key. This field is empty if a retired Master File Key does not exist.

`[Retired MFK Length]`

Field 10, the retired Master File Key's length. This field will contain the length of the retired Master File Key. This field is empty if a retired Master File Key does not exist.

`[KEK Check Digits]`

Field 11, the Key Exchange Key's check digits. This field is a 4 byte hexadecimal value. This field is empty if a Key Exchange Key does not exist.

`[KEK Length]`

Field 12, the Key Exchange Key's length, single or double. This field returns an S if the Key Exchange Key is 1key-3DES (single-length); it returns a D if the Key Exchange Key is 2key-3DES (double-length). This field is empty if a Key Exchange Key does not exist.

`Reserved`

Field 13, this field is reserved and must be empty.

Table 11-15. Response AA: Security Processor Status Key

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	2	AA
1	Remaining slots	4	0 - 9
2	[MFK name]	0, 4	0 - 9, A - Z
3	[MFK Check Digits]	0, 4	0 - 9, A - F
4	[MFK length]	0, 1	D
5	[Pending MFK name]	0, 5	PMFK1
6	[Pending MFK Check Digits]	0, 4	0 - 9, A - F
7	[Pending MFK l5.1(l .4(A d773.6((, 1)-884737(D)]TJT*0003 Tc0006 Tw[82)-5454.5([)-.6Retir		
8	[KEK Check Digits]	0, 4	0 - ,(A - F)]TJT*0005 Tc0004 Tw[120
9	[KEK Length]	0, 1	S, D
10	Reserved	0	

Usage Notes

You can send Command 9A to the Network Security Processor after it has become a member of a security association.

Example

Master File Key = 2ABC3DEF45670189 98107645FEDCBA2, check digits = 057A.
See 2key-3DES Key (Double-Length) on page A-6 for component values.

Obtaining Key status.

The command looks like this:

```
<9A#KEY#>
```

The Network Security Processor returns a response that contains the following information:

- Remaining slots in key table: 4000.
- Master file key's name: MFK1.
- Master file key's check digits: 057A.
- Master file key's length: double-length (D).
- Pending master file key's name: PMFK1.
- Pending master file key's check digits: 2590.
- Pending master file key's length: double-length (D).
- Key exchange key's check digits: 50B0.
- Key exchange key's length: double-length.

The response looks like this.

```
<AA#9999#MFK1#057A#D#PMFK1#2590#D####50B0#D##>
```

Configure Security Processor Option (Command 101)

Command 101 enables and disables various operating parameters. The values defined for these options are stored in non-volatile memory. Power cycling the Network Security Processor does not change the value of an option.

Command

```
<101#[Option Text]#>
```

Response

```
<201#Y#>[CRLF]
```

Calling Parameters

`101`

Field 0, the command identifier.

`Option Text`

Field 1, the option text. Option text is made up of option words. Each option word consists of a three-digit option ID and a one-digit action flag. The length of this field must be zero or a multiple of four. When this field is empty, all options will be set to their default values.

Option ID #	Description
020	Append the Master File Key name to all responses except the response of the status command, 9A; default – do not append name.
021	Append the detailed error information to the error response, 00; default – do not append detailed error.
023	Remove the carriage return and line feed from all responses; default – CR/LF appended to all responses.
027	Use the rightmost 4 PIN digits for Diebold PIN verification; default is to use the leftmost 4 PIN digits.

Option ID #	Description
044	Logs command in error and response; default - do not log error. The command in error and the NSP response are logged to the system log. **When enabled, this option can have a significant negative impact on the performance of the NSP. This option should only be enabled to capture an invalid command that generates an NSP error response. Once the invalid command has been captured it is highly recommended that this option be disabled.** When this option is enabled, warning messages may appear in the system log for commands sent by the SCA-3 to the NSP, such as <Remote#Info#> and <9A#HEADERS#>. This is not an error condition it is normal behavior when this option is enabled.

Action Flag	Meaning
D	Disable option.
E	Enable option.

Note. Options 24 and 40 have been replaced by options A1 and A0, respectively.

Table 11-16. Command 101: Configure Security Processor Option

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	101
1	Option text	multiple of 4	0 - 9, E, D

Responding Parameters

`201`

Field 0, the command identifier.

`Y`

Field 1, an indicator that the table has been configured with the options specified in the command.

Table 11-17. Response 201: Configure Security Processor Option

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	201
1	Configuration confirmation	1	Y

Examples

Using Command 101 to set various configuration options.

- Append the Master File Key name to all responses except 9A; indicated by the option text 020E.
- Append detailed error information to response 00; indicated by the option text 021E.
- Use the rightmost 4 PIN digits for Diebold PIN verification; indicated by option text 027E

The command looks like this:

```
<101#020E021E027E#>
```

The Network Security Processor returns the following response:

```
<201#Y#>
```

Using Command 101 to disable the carriage return and line feed (CRLF).

The command looks like this:

```
<101#023E#>
```

The Network Security Processor returns the following response:

```
<201#Y#>
```

Using Command 101 to enable error logging.

The command looks like this:

```
<101#044E#>
```

The Network Security Processor returns the following response:

```
<201#Y#>
```

Using Command 101 to disable error logging.

The command looks like this:

```
<101#044D#>
```

The Network Security Processor returns the following response:

```
<201#Y#>
```

Using Command 101 to reset options to their default values.

The command looks like this:

```
<101##>
```

The Network Security Processor returns the following response:

```
<201#Y#>
```

Command Monitoring (Command 102)

Command 102 allows you to obtain the number of PIN, sanity, CVV/CVC/CSC, and MAC verification failures that have been processed by the Network Security Processor. It can also be used to count the number of times an enabled command has been processed by the Network Security Processor.

Note. This command is only allowed on the Management Port and is not enabled in the Network Security Processor's default security policy. To use this command enable it in the Network Security Processor's security policy.

Command

```
<102#Action#Mode#[Command]#>
```

Response

```
<202#Action#Mode#Start Time#End Time#[Count]#>
```

Calling Parameters

`102`

Field 0, the command identifier.

`Action`

Field 1, the action to be performed. The allowed values are:

Value	Description
START	Start monitoring.
RETRIEVE	Return the count value, reset the count value to zero and continue monitoring.
STOP	Stop monitoring.

`Mode`

Field 2, the mode of operation. The allowed values are:

Value	Description
0	Count the number of PIN verification failures when the Network Security Processor processes any of these commands: D0, 32, 36, 37, 38, 3A, 3F, 322, 323, 328, 329, 32A and 387.
1	Count the number of PIN sanity failures when the Network Security Processor processes any of these commands: 31, 32, 33, 35, 36, 37, 38, 39, 3A, 3D, 3F, 90, BA, BB, BD, 161, 163, 163, 322, 323, 328, 329, 32A, 331, 335, 346, 347, 362, 363, 364, 370, 371, 372, 387, 3A2 and 3A3.
2	Count the number of CVV/CVC/CSC verification failures when the Network Security Processor processes any of these commands: 3A, 5E, 357, 359, 35A, 35F and 36A.
3	Count the number of MAC verification failures when the Network Security Processor processes any of these commands:58, 5C, 5F, 99, 9C, BA, BB, DA, 301, 30B, 30D, 30E, 346, 348, 355 and 381.
4	Count the number of times the commands, specified in field 3, have been successfully processed.

`[Command]`

Field 3, the list of enabled commands to be counted. This field must contain a command ID, or a comma separated list of command IDs when field 1 (Action) contains the word START and field 2 (Mode) contains the value 4; otherwise it must be empty. A maximum of 16 commands can be counted.

Table 11-18. Command 102: Command Monitoring

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	102
1	Action	4, 5, 8	START, RETRIEVE, STOP
2	Mode	1	0-4
3	[Command]	0-80	0 -9, A-F, “,”

Responding Parameters

`202`

Field 0, the response identifier.

`Action`

Field 1, the action value supplied in field 1 of the command.

`Mode`

Field 2, the mode value supplied in field 2 of the command.

`Start Time`

Field 3, the date/time when the monitoring task was started.

The format is: `YYYYMMDD HH:MM:SS.`
A start time value of 20121221 19:03:12, is December 21, 2012 7:03:12 PM.

`End Time`

Field 4, the date/time when the monitoring task was stopped or data was retrieved. This field will be empty when the action in field 1 contains the value START.

The format is: `YYYYMMDD HH:MM:SS.`
An end time value of 20121221 20:03:12, is December 21, 2012 8:03:12 PM.

`[Count]`

Field 5, the count value. This field will be empty when the action specified in field 1 contains the value START.

The count value and format of this field depends on the mode specified in field 2. When field 2 contains a mode value in the range of 0-3, this field will contain a count value indicating the number of times the mode being counted has occurred. When field 2 contains a mode value of 4, this field will contain a count value of the command being counted, the format is `CMDID=COUNT`. Multiple commands are separated by a comma.

Table 11-19. Response 202: Command Monitoring

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	202
1	Action	4, 5, 8	START, RETRIEVE, STOP
2	Mode	1	0 - 4
3	Start Time	17	0 -9, “:”
4	End Time	17	0 -9, “:”
5	[Count]	0, varies	0 - 9, A - F, “=”, “,”

Usage Notes

- Multiple instances of this command can run concurrently however only one instance of each mode is allowed.
- The RETRIEVE action resets the count value to zero.
- The maximum count value is 4,294,967,295. The count value will be reset to zero if the maximum count value is exceeded.

Examples

Start counting the number of PIN Verification failures

The command looks like this:

```
<102#START#0##>
```

The Network Security Processor issues a response similar to this:

```
<202#START#0#20121221 19:57:58###>
```

Get the number of times a PIN has failed to verify

```
<102#RETRIEVE#0##>
```

The Network Security Processor issues a response similar to this:

```
<202#RETRIEVE#0#20121221 19:57:58#20121221 20:20:41#3#>
```

Stop counting the number of PIN verification failures

The command looks like this:

```
<102#STOP#0##>
```

The Network Security Processor issues a response similar to this:

```
<202#STOP#0#20121221 20:20:41#20121221 20:20:51#0#>
```

Start a command count for commands 31 and 335

The command looks like this:

```
<102#START#4#31,335#>
```

The Network Security Processor issues a response similar to this:

```
<202#START#4#20121221 19:03:12###>
```

Get the number of times commands 31 and 335 have been successfully processed

The command looks like this:

```
<102#RETRIEVE#4##>
```

The Network Security Processor issues a response similar to this:

```
<202#RETRIEVE#4#20121221 20:07:25#20121221 20:09:12#
31=971,335=244#>
```

Stop the command counting test

The command looks like this:

```
<102#STOP#4##>
```

The Network Security Processor issues a response similar to this:

```
<202#STOP#4#20121221 20:09:12#20121221 20:09:24#31=88,335=6#>
```

Enable Premium Value Commands and Options (Command 105)

The Network Security Processor serial number is required when placing an order for a premium value commands and options. When the order is processed, Atalla Technical Support will provide a Command 105 for that specific serial number. Be sure you send the Command 105 to the correct Network Security Processor. You can use command 9A to obtain the Network Security Processor serial number. The Network Security Processor serial number and MAC are validated before the configuration text is processed.

This command updates nonvolatile memory with the configuration text. Premium value commands and options are not lost if the Network Security Processor is powered off. It is not necessary to send this command each time the is powered on.

Note. After sending the command 105 to the Network Security Processor, you must also add the premium value command(s) or option(s) to the Network Security Processor's security policy using either the SCA-3 or commands108 and 109. The Network Security Processor requires that the command 105 be executed before the premium value command(s) or option(s) can be added to the Network Security Processor's security policy.

Command

```
<105#Serial Number#Encrypted Configuration Text#MAC#>
```

Response

```
<205#Status#Version#>[CRLF]
```

Calling Parameters

`105`

Field 0, the command identifier.

`Serial Number`

Field 1, the serial number of the Network Security Processor. In versions 1.13 and above, lowercase characters are allowed.

`Encrypted Configuration Text`

Field 2, the encrypted Configuration Text. When decrypted by the Network Security Processor, this field defines the premium value commands and options to be enabled.

`MAC`

Field 3, the Message Authentication Code. The Network Security Processor validates the MAC before it processes the configuration text.

Table 11-20. Command 105:Enable Premium Value Commands and Options

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	105
1	Serial Number	6	0 - 9, A - Z
2	Encrypted Configuration Text	various	0 - 9, A - F
3	MAC	9	0 - 9, A - F, space

Responding Parameters

`205`

Field 0, the command identifier.

`Status`

Field 1, the status of processing the command.

- COMPLETED indicates the command was successfully processed.
- MAC MISMATCH indicates that the MAC did not validate.
- CONF INVALID indicates that the decrypted configuration text contained an error.
- SN MISMATCH indicates that the serial number of the Network Security Processor does not match the serial number in the command 105.

`Version`

Field 2, the version of the command.

Table 11-21. Response 205: Enable Premium Value Commands and Options

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	205
1	Status	12	COMPLETED, MAC MISMATCH CONFIG INVALID SN MISMATCH
2	Version	4	0 - 9, A - Z

Example

Using Command 105 to enable a premium value command or option. The serial number is 123456. This is not a working example.

```
<105#123456#E7F35DA354A09F32#B65F 3CA0#>
```

The Network Security Processor returns a response similar to this:

```
<205#COMPLETED#VER1#>
```

Define Temporary Serial Number (Command 106)

Each Network Security Processor has a unique permanent serial number. This serial number is used to create a unique command 105, that when sent to the Network Security Processor, licenses premium value commands or options. The licensed premium value commands or options must then be enabled in the Network Security Processor's security policy using either the SCA-3 or commands 108 and 109.

If a Network Security Processor that is configured with premium value commands or options fails, it will be replaced with a Network Security Processor that has a different permanent serial number. To quickly configure the replacement Network Security Processor with the same premium value commands or options as those licensed in the failed Network Security Processor, the replacement Network Security Processor must be loaded with a temporary serial number which is the serial number of the failed Network Security Processor. This allows the replacement Network Security Processor to accept the command 105 created for the failed Network Security Processor.

Commands 106 and 107 operate as a pair, they are used to temporarily load the serial number of another Network Security Processor into a replacement Network Security Processor.

Note. If the Secure Configuration Assistant-3 (SCA-3) is used to initialize the Network Security Processor, use the SCA-3's Set Temporary Serial Number feature instead of commands 106 and 107.

The temporary serial number is stored when the Network Security Processor successfully processes a command 107. If power is lost before the command 107 is processed, the temporary serial number is erased. If this should happen you must send the command 106 again, then the corresponding command 107.

▲ **WARNING.** This temporary serial number is valid for 120 hours (5 days) from the time that the temporary serial number was set in the Network Security Processor. For example, if the Network Security Processor receives the temporary serial number on Wednesday at 6:30 AM, the temporary serial number will expire at 6:00AM on Monday.

If the Network Security Processor does not receive a command 105 based on its permanent serial number within 120 hours **all** premium value commands and options are reset to the factory default security policy. To prevent this from happening you must perform these steps within this 120 hour time frame:

a) Contact Atalla Technical Support and provide the serial numbers of the failed and replacement Network Security Processor. Atalla Technical Support will generate a new command 105 based on the replacement Network Security Processor's serial number.

b) Send this new command 105 to the Network Security Processor.

The temporary serial number is erased when the Network Security Processor receives a command 105 based on its permanent serial number.

Command

```
<106#Permanent Serial Number#Temporary Serial Number#>
```

Response

```
<206#Status#Permanent Serial Number#
Temporary Serial Number#Challenge#Check Digits#>[CRLF]
```

Calling Parameters

`106`

Field 0, the command identifier.

`Permanent Serial Number`

Field 1, the permanent Network Security Processor serial number.

`Temporary Serial Number`

Field 2, the temporary serial number you wish to load into the replacement Network Security Processor. This should be the permanent serial number of the defective Network Security Processor. You can obtain this value from the Command 105 issued for the defective Network Security Processor. Or you can also obtain this value from the back of the defective Network Security Processor.

Table 11-22. Command 106: Define Temporary Serial Number

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	106
1	Permanent Serial Number	6	0 - 9, A - Z
2	Temporary Serial Number	6	0 - 9, A - Z

Responding Parameters

`206`

Field 0, the command identifier.

`Status`

Field 1, the status of processing the command.

- COMPLETED indicates that the command completed successfully.

- SN MISMATCH indicates that the permanent serial in the Network Security Processor does not match the permanent serial number in the command 106. Use command 9A to obtain the permanent serial number.
- TMP EXISTS indicates that the Network Security Processor already has a temporary serial number. If the wrong temporary serial number has been loaded you can power cycle the Network Security Processor to erase it.

`Permanent Serial Number`

Field 2, the Network Security Processor permanent serial number.

`Temporary Serial Number`

Field 3, the temporary serial number input as field 2 of the command. This field will be empty unless the status field in the response contains COMPLETED.

`Challenge Number`

Field 4, the challenge number. This random value must be encrypted under variant 30 of the MFK. Use the SCA-3 Calculate AKB/Cryptogram feature to perform this task, see the for the procedure.

The encrypted value is used in field 1 of command 107. This field will be empty unless the status field in the response contains COMPLETED.

`Check Digits`

Field 5, the check digits of the challenge number. Use this value to confirm that you have correctly entered the challenge into the SCA-3. This field will be empty unless the status field in the response contains COMPLETED. If option 88 is enabled, this field will contain six bytes of check digits.

Table 11-23. Response 206: Define Temporary Serial Number (page 1 of 2)

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	206
1	Status	11	COMPLETED SN MISMATCH TMP EXISTS
2	Permanent Serial Number	6	0 - 9, A - Z

Table 11-23. Response 206: Define Temporary Serial Number (page 2 of 2)

Field #	Contents	Length (bytes)	Legal Characters
3	Temporary Serial Number	0, 6	0 - 9, A - Z. This field will be empty unless field 1 indicates COMPLETED.
4	Challenge Number	0, 16	0 -9, A - F. This field will be empty unless field 1 indicates COMPLETED.
5	Check Digits	4 or 6	0 - 9, A - F.

Example

Using Command 106 to define a temporary serial number.

The command looks like this:

```
<106#123456#654321#>
```

The Network Security Processor returns a response similar to this:

```
<206#COMPLETED#123456#654321#7C54B39AAE85A011#A371#>
```

Confirm Temporary Serial Number (Command 107)

Command 107 is used to implement the temporary serial number defined using command 106.

Note. If the Secure Configuration Assistant-3 (SCA-3) is used to initialize the Network Security Processor, use the SCA-3's Set Temporary Serial Number feature instead of commands 106 and 107.

Command

```
<107#Cryptogram of the Challenge#>
```

Response

```
<207#Status#Permanent Serial Number#
Temporary Serial Number#>[CRLF]
```

Calling Parameters

`107`

Field 0, the command identifier.

`Cryptogram of the Challenge`

Field 1, the challenge, from the response to command 106, encrypted under variant 30 of the Master File Key (MFK). Use the SCA-3 Calculate AKB/Cryptogram feature to perform this task, see the for the procedure.

Table 11-24. Command 107: Implement Temporary Serial Number

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	107
1	Cryptogram of the Challenge	16	0 - 9, A - F

Responding Parameters

`207`

Field 0, the command identifier.

`Status`

Field 1, the security policy that was just implemented.

- COMPLETED indicates that the command was successfully processed.
- NO TMP SN indicates that the Network Security Processor does not have a challenge number or serial number in memory. Repeat command 106.
- BAD CHALLENGE indicates that the challenge was not correct. Be sure to enter the challenge correctly into the SCA-3.

`Permanent Serial Number`

Field 2, the permanent serial number of the Network Security Processor.

`Temporary Serial Number`

Field 3, the temporary serial number defined with command 106. This field will be empty unless the status field in the response contains COMPLETED.

Table 11-25. Response 207: Implement Temporary Serial Number

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	207
1	Status	13	COMPLETED NO TMP SN BAD CHALLENGE
2	Permanent Serial Number	6	0 - 9, A - Z
3	Temporary Serial Number	6	0 - 9, A - Z. This field will be empty unless field 1 indicates COMPLETED.

Example

Using Command 107 to implement a temporary serial number.

This is not a working example.

The command looks like this:

```
<107#B203A98A64C8F906#>
```

The Network Security Processor returns a response similar to this:

```
<207#COMPLETED#123456#654321#>
```

Define Security Policy (Command 108)

Commands 108 and 109 work as a pair to define and then implement a security policy. Use command 108 to define the security policy. The response to the 108 command is a challenge that must be encrypted under variant 30 of the Master File Key (MFK). Use the SCA-3 Calculate AKB/Cryptogram feature to perform this task, see the for the procedure. This encrypted challenge is then used as an input to command 109 to implement the security policy.

Note. If the Secure Configuration Assistant-3 (SCA-3) is used to initialize the Network Security Processor, use the SCA-3's Configuration Management feature instead of commands 108 and 109.

If power is cycled after the command 108 has been processed, but before the command 109 has been processed, the security policy defined by command 108 will not be implemented. Before a security policy can take effect, commands 108 and 109 must be successfully processed as a pair, without an intervening power cycle.

Using this command it is possible to disable the serial number validation and sequence counter checking, if both of these security parameters are disabled, a warning message will be returned in the response message. You must acknowledge this message in the subsequent command 109.

See Appendix C, Summary of Commands and Options for a complete list of commands and options that can be enabled or disabled using this command.

Premium value commands and options, enabled with command 105, must be added to the Network Security Processor's security policy with commands 108 and 109 **before** they can be used by the Network Security Processor.

Command Counting

The command count table resides in non-volatile RAM - it is maintained even if the Network Security Processor loses power. The table is constructed such that a maximum of nine cryptographic commands can be counted.

Command 108 supports the ability to specify a command count. The count value must be in the range of 1 to 4 billion (4,000,000,000). Utility commands and options cannot be counted. Premium value commands must be first enabled with a command 105 before they can be counted.

Each time the Network Security Processor successfully processes a command that is being counted the count value is decremented by 1. Commands that are not successfully processed by the Network Security Processor, such as commands that contain syntax error(s) that result in an error response, are not counted.

△ **Caution.** Once the count value reaches zero, the Network Security Processor will return an error <00#0300xx#> instead of processing the command. The command <9A#COUNT#> can be used to obtain the current count value for all commands in the command count table.

The count value is specified using the letter “N” or “n”, followed by the count value (decimal). When a command count has be specified, the command is automatically enabled in the Network Security Processor’s security policy for that number of executions, any previously defined count value is replaced by the count value currently being specified.

If a command that is currently being counted is disabled in the Network Security Processor’s security policy, the count value for that command remains in the command count table, such that, if the command is ever enabled the count value will be applied.

A command that is currently being counted can be removed from the command count table using the letter “R” or “r”. When a command is removed from the command count table, it is also disabled in the Network Security Processor’s security policy.

When a Network Security Processor is reset to factory state, or the Network Security Processor’s security policy is reset to factory state, all data stored in the command count table is erased.

See Examples on page 11-51, for some security polices that demonstrate command counting.

Command

```
<108#Security Policy#>
```

Response

```
<208#[Warning Message]#Left Challenge#
Left Challenge Check Digits#Right Challenge#
Right Challenge Check Digits#Counter#Sequence Number#
Serial Number#>[CRLF]
```

Calling Parameters

`108`

Field 0, the command identifier.

`Security Policy`

Field 1, the security policy string. The security policy is a string that defines what commands and options are enabled or disabled. The format is:

Command ID followed by an equal sign “=”, followed by a one-digit action flag “e” or “E” for enable, “d” or “D” for disable, “r” or “R” to remove a command from the counter table, and “n” or “N” followed by a count value to specify the number of successful command executions allowed. The Command ID must be upper case.

For example, to enable command 1A, the security policy string would be 1A=e, or 1A=E. The security policy strings 1a=e or 1a=E, are not correct because the command ID is not upper case.

Options are surrounded by parenthesis, they must be uppercase, for example (6E) not (6e). The option is followed by an equal sign “=”, followed by a value which is surrounded by double quotes. The option value can be either upper or lower case. For example, to enable option 6E the security policy string would be (6E)=”e”, or (6E)=”E”.

If multiple commands or options are to be enabled or disabled in the same security policy string, they must be separated by a semicolon”;”. For example, 1A=e;(6E)=”D”. See Examples on page 11-51 for some typical security policies.

A command or option can only have one value for a given security policy. For example, if a security policy string enables a command then subsequently disables it in the same string, an error 20 will be returned.

If this field contains the word “FACTORY”, **all** commands and options will be set to the Network Security Processor’s factory default security policy, therefore to enable premium value commands you must send the command 105 to the Network Security Processor, after using this value. If necessary, you can use this value to quickly undue a security policy and return the Network Security Processor to a known factory state. The word factory is not case-sensitive, “factory”, and “Factory” are also valid.

Table 11-26. Command 108: Define Security Policy

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	108
1	Security Policy	0-4000	0 - 9, A - F, FACTORY

Responding Parameters

`208`

Field 0, the command identifier.

`[Warning Message]`

Field 1, a warning message that indicates that both the serial number checking option (6E) and sequence counter checking option (6F) have been disabled, either prior to, or as a result of, this command. This warning field must be acknowledged in command 109 as part of the response to the challenge. The warning message is:

“SECURITY PRECAUTION: Are you sure?”

This message will only appear when options 6E and 6F are enabled in the security policy.

`Left Half Challenge`

Field 2, the left half of the challenge. This value must be encrypted under variant 30 of the MFK. If you use the Calculate Crypto feature in the SCA-3 to encrypt the challenge.

`Left Half Challenge Check Digits`

Field 3, the check digits for the left challenge. Use this value to confirm that you have correctly entered the left half of the challenge into the SCA-3. If option 88 is enabled, this field will contain a six-byte check digits.

`Right Half Challenge`

Field 4, the right half of the challenge. This value must be encrypted under variant 30 of the MFK. If you use the Calculate Crypto feature in the SCA-3 to encrypt the challenge.

`Right Half Challenge Check Digits`

Field 5, the right half of the challenge. Use this value to confirm that you have correctly entered the right half of the challenge into the SCA-3. If option 88 is enabled, this field will contain a six-byte check digits.

`Counter`

Field 6, the number of times an attempt has been made to update the security policy. It is displayed so you can monitor the number of times command 108 has been attempted. This value is maintained in volatile memory, therefore each time the Network Security Processor is powered on this value will be reset to zero.

`Sequence Number`

Field 7, the number of times the security policy has been successfully updated. This value is used in processing of the security policy. It is included in the response so you can keep track of the number of times the Network Security Processor security policy has been updated. This value is stored in non-volatile memory and is incremented as the result of successfully processing a command 109.

`Serial Number`

Field 8, the serial number of the unit. This value is unique to each Network Security Processor, and is used in processing of the security policy. It is included in the response so you can keep track of which unit to send the subsequent command 109. This value is stored in non-volatile memory and cannot be changed.

Table 11-27. Response 208: Define Security Policy

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	208
1	[Warning Message]	0 or 34	
2	Left Half Challenge	16	0 - 9, A - F
3	Left Half Check Digits	4 or 6	0 - 9, A - F
4	Right Half Challenge	16	0 - 9, A - F
5	Right Half Check Digits	4 or 6	0 - 9, A - F
6	Counter	1-4	0 - 9
7	Sequence Number	16	0 - 9
8	Serial Number	6-7	variable

Usage Notes

Use the SCA-3 Calculate AKB/Cryptogram feature to encrypt the challenge, see the for the procedure.

Examples

Using Command 108 to define several security policies.

If you use Hyperterminal to communicate with the Network Security Processor be advised that the Hyperterminal feature Paste to Host (Control V) will either truncate or change the value of the double quote character. Therefore you cannot copy (Control C) and paste (Control V) examples that have an option ID and value. Instead you must manually enter the command 108 into Hyperterminal. An indication you are experiencing this problem is you will get an error 23 as a response instead of the response listed below.

Enabling specific commands and options

This example illustrates using Command 108 to enable the following commands and options:

Commands: 30, and 90.
Options: 66, and setting the minimum PIN length to 6.

The command looks like this:

```
<108#30=e;90=E;(66)=”e”;(A0)=”6”#>
```

The Network Security Processor returns a response similar to this:

```
<208##23A4DF7983208992#4AF3#12C42BDAD34798FF#7BB2#1#
0000000000000001#A7PV87#>
```

Disabling specific commands and options

This example illustrates using Command 108 to disable the following commands and options:

Commands: 30, 90.
Option: 66.

The command looks like this:

```
<108#30=D;90=d;(66)=”d”#>
```

The Network Security Processor returns a response similar to this:

```
<208##23A4DF7983208992#4AF3#12C42BDAD34798FF#7BB2#1#
0000000000000001#A7PV87#>
```

Enabling and disabling commands and options with a single command

This example illustrates using Command 108 to enable the following commands and options:

Commands: 30, 90.
Option: 66.

And disable the following commands and options:

Commands: 10, 98.
Options: 60, 65.

And to set the minimum PIN length to 6, and to set the sanity indicator tool”.

The command looks like this:

```
<108#30=e;90=e;(66)=”e”;10=d;98=d;(60)=”d”;(65)=”d”;
(A0)=”6”;(A1)=”L”#>
```

The Network Security Processor returns a response similar to this:

```
<208##23A4DF7983208992#4AF3#12C42BDAD34798FF#7BB2#1#
0000000000000001#A7PV87#>
```

Enabling and disabling the same command in a single command

This example illustrates using Command 108 to enable the following commands and options.

Commands: 30, 90, 32, and 37.

Options: 66 and setting the minimum PIN length to 6.

And disables the following commands and options:

Commands: 10, 30, 98.

Option: 65.

The command looks like this:

```
<108#30=e;90=e;32=e;37=e;(66)=”e”;(A0)=”6”;10=d;30=d;
98=d;(65)=”d”#>
```

This example produces an error because command 30 is both and disabled in the same security policy string. The Network Security Processor returns a response similar to this:

```
<00#270127#030=d#>
```

Disabling the sequence number and serial number validation.

This example produces a warning message because the security policy disables both the sequence number and serial number validation, by enabling options (6E) and (6F), respectively.

The command looks like this:

```
<108#(6E)=”e”;(6F)=”e”#>
```

The Network Security Processor returns a response similar to this:

```
<208#SECURITY PRECAUTION: Are you sure?#
23A4DF7983208992#4AF3#12C42BDAD34798FF#7BB2#1#
00000000000000001#A7PV87#>
```

Enabling the Factory security policy

This example shows how to reinstate the factory security policy. If this command is processed by the Network Security Processor, the configuration information will be erased. Therefore to configure the Network Security Processor for premium value commands and options the command 105 must be sent again to the Network Security Processor.

The command looks like this:

```
<108#FACTORY#>
```

The Network Security Processor returns a response similar to this:

```
<208##23A4DF7983208992#4AF3#12C42BDAD34798FF#7BB2#1#
00000000000000001#A7PV87#>
```

Enabling command counting

This example shows how to enable commands 10 for 100 executions and command 31 for 50,000 executions.

The command looks like this:

```
<108#10=n100;31=N50000#>
```

The Network Security Processor returns a response similar to this:

```
<208##0DC2D50EF492E33D#30D6#DAF29816C8B96843#9EBC#1#
0000000000000004#A7PV87#>
```

Disabling a command and removing it from the counter table

This example shows how to remove command 10 from the command count table and disable it in the Network Security Processor's security policy.

```
<108#10=r#>
```

The Network Security Processor returns a response similar to this:

```
<208##AD4029E607385DDA#99D5#CEDF326710E08F49#0D37#1#
0000000000000005#A7PV87#>
```

Confirm Security Policy (Command 109)

Command 109 is used to implement the security policy you defined using command 108.

Note. If the Secure Configuration Assistant-3 (SCA-3) is used to initialize the Network Security Processor, use the SCA-3's NSP Configuration Management feature instead of commands 108 and 109.

Command

```
<109#[Warning Acknowledgement]#Cryptogram of the Challenge#>
```

Response

```
<209#Security Policy#Sequence Number#Serial Number#>[CRLF]
```

Calling Parameters

`109`

Field 0, the command identifier.

`[Warning Acknowledgement]`

Field 1, the warning acknowledgment. If the security policy, defined in command 108 command, disabled both the sequence number and serial number validation, options (6E) and (6F), a warning message "SECURITY PRECAUTION: Are you sure?" was included in the 208 response. You must supply the following warning acknowledgment message in this field:

```
I accept
```

before the Network Security Processor's security policy will be implemented. This field is not case sensitive. Leave this field blank if the Network Security Processor's security policy does not disable both of these options.

`Cryptogram of the Challenge`

Field 2, the challenge encrypted under variant 30 of the Master File Key (MFK). Use the SCA-3 Calculate AKB/Cryptogram feature to perform this task, see the for the procedure.

Table 11-28. Command 109: Confirm Security Policy

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	3	109
1	[Warning Acknowledgment]	0 or 8	I accept
2	Cryptogram of the Challenge	32	0 - 9, A - F

Responding Parameters

`209`

Field 0, the command identifier.

`Security Policy`

Field 1, the security policy that was just implemented. If the security policy is defined by several command 108 commands, this field will only show the security policy for the most recent 108 command. You can use the Security Processor Configuration Status (Command 9A) on page 11-8 to obtain a complete list of commands and options enabled and disabled in the Network Security Processor.

`Sequence Number`

Field 2, the number of times the security policy has been successfully updated. This value is used in processing of the security policy. It is displayed so you can keep track of the number of times each of your Network Security Processor's security policy has been updated.

`Serial Number`

Field 4, the serial number of the unit. This value is unique to each Network Security Processor, and is used in processing of the security policy. This value is displayed so you can be certain which Network Security Processor has had its security policy updated.

Table 11-29. Response 209: Confirm Security Policy

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	3	209
1	Security Policy	variable, 4000 characters maximum	variable
2	Sequence Number	16	0 - 9
3	Serial Number	8	variable, ASCII

Example

Using Command 109 to implement a security policy.

The command looks like this:

```
<109##203A98A64C8F900C62E1E8368E43A751#>
```

The Network Security Processor returns a response similar to this:

```
<209#30=e;90=e;32=e;37=e;(66)=”e”;(A0)=”6”;10=d;30=d;
98=d;(65)=”d”#0000000000000001#A7PV87#>
```

Get ID of Current Image (Command 1101)

Command 1101 allows you to obtain the image ID, CRC checksum, and the product code of the cryptographic command processor in the Network Security Processor.

Command

```
<1101#>
```

Response

```
<2101#Image ID#Image CRC Checksum#Product Code#>
```

Calling Parameters

`1101`

Field 0, the command identifier.

Table 11-30. Command 1101: Get ID of Current Image

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1101

Responding Parameters

`2101`

Field 0, the response identifier.

`Image ID`

Field 1, the Network Security Processor's image ID, which consists of the image name, version number, and creation date.

`Image CRC Checksum`

Field 2, the CRC checksum of the Network Security Processor image.

`Product Code`

Field 3, the Network Security Processor's product code. The number 2 indicates that the NSP supports the improved security features introduced in NSP version 2.0.

Table 11-31. Response 2101: Get ID of Current Image

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2101
1	Image ID	0 - n	Any except # < >
2	Image CRC Checksum	4	0 - 9, A - F
3	Product Code	1	2

Example

Using Command 1101 to obtain the version of the image in the Network Security Processor.

The command looks like this:

```
<1101#>
```

The Network Security Processor issues a response similar to this:

```
<2101#HP Atalla A10160-VAR Version: 2.00, Date: Apr  8 2013,
Time: 10:01:32#6CFF#2#>
```

Get Virtual NSP Information (Command 1102)

Command 1102 allows you to obtain the number of the virtual NSP that the host application is connected to, the name of the virtual NSP, and number of virtual NSPs defined.

Note. This command is only allowed on the Management Port.

Command

```
<1102#>
```

Response

```
<2102#VNSPx#[NAME]#VNSP Count#>
```

Calling Parameters

`1102`

Field 0, the command identifier.

Table 11-32. Command 1102: Get Virtual NSP Information

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1102

Responding Parameters

`2102`

Field 0, the response identifier.

`VNSPx`

Field 1, the virtual NSP number that the host application is connected to. Values can be in the range of VNSP0 through VNSP9.

`[NAME]`

Field 2, the name of the virtual NSP (if one has been defined in the config.prm file).

`VNSP Count`

Field 3, the number of virtual NSPs defined in the config.prm file.

Table 11-33. Response 2102: Get Virtual NSP Information

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2102
1	VNSPx	4	VNSPx, x=0-9.
2	[NAME]	0 - 64	0 - 9, A - Z, a - z, _

Usage Notes

When virtual NSP support is not enabled, the response to this command will be

```
<2102#VNSP0##1#>
```

Example

Use Command 1102 to obtain the virtual NSP information. The host application is connected to VNSP2 that has a name defined as “PRODUCTION_SOUTH”; there are 5 virtual NSP defined in the physical NSP.

The command looks like this:

```
<1102#>
```

The Network Security Processor issues a response similar to this:

```
<2102#VNSP2#PRODUCTION_SOUTH#5#>
```

Get Temporary Serial Number Information (Command 1104)

Command 1104 allows you to obtain the temporary serial number and the number of hours remaining before it expires.

Command

```
<1104#>
```

Response

```
<2104#Temporary Serial Number#Remaining Hours#>
```

Calling Parameters

`1104`

Field 0, the command identifier.

Table 11-34. Command 1104: Get Virtual NSP Information

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1104

Responding Parameters

`2104`

Field 0, the response identifier.

`Temporary Serial Number`

Field 1, the virtual NSP number that the host application is connected to. Values can be in the range of VNSP0 through VNSP9.

`Remaining Hours`

Field 2, the number of hours before the temporary serial number expires.

Table 11-35. Response 2104: Get Temporary Serial Number Information

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2104
1	Temporary Serial Number	0,6	A-Z, 0-9
2	Remaining Hours	1-3	0-120

Usage Notes

If no temporary serial number has been defined of if it has expired the response to this command will be:

```
<2104##0#>
```

Example

Use Command 1104 to obtain the temporary serial number information.

The command looks like this:

```
<1104#>
```

The Network Security Processor issues a response similar to this indicating that the temporary serial number is 123456 and it will expire in 48 hours.

```
<2104#123456#48#>
```

License Premium Value Commands/Options in all Virtual NSPs (Command 1105)

Command 1105 is very similar to command 105, however it will simultaneously license the premium value command/option configuration for all virtual NSPs that are currently configured.

Note. After sending the command 1105 to the Network Security Processor, you must also add/delete the premium value commands or options on each of the Virtual Network Security Processor's security policy using the SCA-3.

Note. When multiple virtual NSP support is enabled in the Network Security Processor, this command must be sent to VNSP0.

Command

```
<1105#Serial Number#Encrypted Configuration#MAC#>
```

Response

```
<2105#Status#Version#>
```

Calling Parameters

`1105`

Field 0, the command identifier.

`Serial Number`

Field 1, the serial number of the Network Security Processor.

`Encrypted Configuration`

Field 2, the encrypted Configuration. When decrypted by the Network Security Processor, this field defines the premium value commands and options to be licensed.

`MAC`

Field 3, the Message Authentication Code. The Network Security Processor validates the MAC before it processes the configuration.

Table 11-36. Command 1105: License Premium Value Commands/Options in all Virtual NSPs

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1105
1	Serial Number	6	0 - 9, A - Z, a -z
2	Encrypted Configuration	various	0 - 9, A - F
3	MAC	9	0 - 9, A - F, space

Responding Parameters

`2105`

Field 0, the command identifier.

`Status`

Field 1, the status of processing the command.

- COMPLETED indicates the command was successfully processed.
- MAC MISMATCH indicates that the MAC did not validate.
- CONF INVALID indicates that the decrypted configuration contained an error.
- SN MISMATCH indicates that the serial number of the Network Security Processor does not match the serial number in the command 1105.

`Version`

Field 2, the version of the command.

Table 11-37. Response 2105: License Premium Value Commands and Options in all Virtual NSPs

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2105
1	Status	12	COMPLETED, MAC MISMATCH CONFIG INVALID SN MISMATCH
2	Version	4	0 - 9, A - Z

Example

Using Command 1105 to license a premium value command or option. The serial number is 123456. This is **not** a working example.

```
<1105#123456#E7F35DA354A09F32#B65F 3CA0#>
```

The Network Security Processor returns a response similar to this:

```
<2105#COMPLETED#VER1#>
```

Get System Configuration Information (Command 1110)

Command 1110 allows you to obtain the Network Security Processor's system software and cryptographic subsystem software information.

Command

```
<1110#>
```

Response

```
<2110#System Software Version Information#
Cryptographic Subsystem Software Version Information#
CRC Checksum#Product Code#>
```

Calling Parameters

`1110`

Field 0, the command identifier.

Table 11-38. Command 1110: Get System Configuration Information

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1110

Responding Parameters

`2110`

Field 0, the response identifier.

`System Software Version Information`

Field 1, consists of the name, version number, and date and time of the operating system and transport layer.

`Cryptographic Subsystem Software Version Information`

Field 2, consists of the name, version number, and creation date and time of the Atalla Cryptographic Subsystem.

`CRC Checksum`

Field 3, the CRC checksum of the Atalla Cryptographic Subsystem.

`Product Code`

Field 4, the Atalla Cryptographic Subsystem product code. The number 2 indicates that the NSP supports the improved security features introduced in NSP version 2.0.

Table 11-39. Response 2110: Get System Configuration Information

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2110
1	System Software Version Information	variable	Any except #, <, or >
2	Cryptographic Subsystem Software Version Information	variable	Any except #, <, or >
3	CRC Checksum	4	0-9, A-F
4	Product Code	1	2

Example

Using Command 1110 to obtain the system configuration information.

The command looks like this:

```
<1110#>
```

The Network Security Processor returns a response similar to this:

```
<2110#Axx160, Version: 2.00, Date: Apr  8 2013, Time:
10:17:29#HP Atalla A10160-VAR Version: 2.00, Date: Apr  8
2013, Time: 10:01:32#6CFF#2#>
```

Get System Date and Time (Command 1111)

Command 1111 returns the Network Security Processor's system date and time in Universal Coordinated Time.

Command

```
<1111#>
```

Response

```
<2111#YYMMDDHHMMSS#>[CRLF]
```

Calling Parameters

`1111`

Field 0, the command identifier.

Table 11-40. Command 1111: Get System Date and Time

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1111

Responding Parameters

`2111`

Field 0, the response identifier.

`YYMMDDHHMMSS`

Field 1, two digit year, two digit month, two digit day, two digit hour, two digit minute, two digit second.

Table 11-41. Response 2111: Get System Date and Time

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2111
1	YYMMDDHHMMSS	12	0 - 9

Example

Using Command 1111 to obtain the system date and time.

The command looks like this:

```
<1111#>
```

The Network Security Processor returns a response similar to this:

```
<2111#060724115300#>
```
(July 24, 2006 11:53:00)

Get Average CPU Utilization (Command 1113)

Command 1113 allows you to obtain a percentage value which is the average CPU utilization for the Network Security Processor. The time period for the measurement is specified in the command. At the end of the time period the Network Security Processor returns a response which contains a percentage value indicating the average CPU utilization.

Note. This command is only allowed on the Management Port.

Command

```
<1113#Test Period#>
```

Response

```
<2113#Percent Utilized#>[CRLF]
```

Calling Parameters

`1113`

Field 0, the command identifier.

`Test Period`

Field 1, the number of seconds that the test will run. The minimum value is 1 the maximum value is 10.

Table 11-42. Command 1113: Get Average CPU Utilization

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1113
1	Test Period	1-2	1-10

Responding Parameters

`2113`

Field 0, the response identifier.

`Percent Utilized`

Field 1, the average CPU utilization during the test period.

Table 11-43. Response 2113 Get Average CPU Utilization

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2113
1	Percent Utilized	1-3	0 - 100

Usage Notes

The Network Security Processor does not return a response until the test completes.

Example

Using Command 1113 to obtain the average CPU utilization for a 10 second time period.

The command looks like this:

```
<1113#10#>
```

The Network Security Processor returns a response similar to this:

```
<2113#37#>
```

Get System Information (Command 1120)

Command 1120 allows you to obtain the NSP serial number, product ID, system software information, and a personality version.

Command

```
<1120#>
```

Response

```
<2120#SerialNumber#ProductID#LoaderVersion#
PersonalityVersion#>
```

Calling Parameters

`1120`

Field 0, the command identifier.

Table 11-44. Command 1120: Get System Information

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1120

Responding Parameters

`2120`

Field 0, the response identifier.

`SerialNumber`

Field 1, the serial number of the Atalla Cryptographic Subsystem.

`ProductID`

Field 2, the model number of the Ax160 NSP.

`LoaderVersion`

Field 3, the version number of the program used to load system images into the Atalla Cryptographic Subsystem.

`PersonalityVersion`

Field 4, the Ax150 personality major version number that was used as a base to create this version. For example a value of 3.70 in this field indicates that the Ax150 version 3.70 was used as a base to create this Ax160 version.

Additional capability has been added to the Ax160 version when the value in this field ends with the letter “X”. For example a value of 3.7x in this field indicates that features and functions added after Ax150 version 3.70 but before Ax150 version 3.80 are present in this Ax160 version.

Customers that have both Ax150 and Ax160 NSPs can use this command to verify that both NSP models are running functionally equivalent software.

Table 11-45. Response 2120: Get System Information

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2120
1	SerialNumber	variable	0 - 9, A - Z
2	ProductID	variable	0 - 9, A - Z
3	LoaderVersion	variable	0 - 9, A - Z
4	PersonalityVersion	variable	0 - 9, A - Z

Example

Use Command 1120 to obtain the system information of the Network Security Processor.

The command looks like this:

```
<1120#>
```

The Network Security Processor returns a response similar to this:

```
<2120#SerialNumber=JL014M#ProductID=A10160#LoaderVersion=0.65
10 SEP 2010 10:54:29#PersonalityVersion=VAR 3.9X#>
```

Get Log Signing Key Certificate (Command 1204)

Command 1204 – Use this command to return the certificate of the RSA signing key used, when the Network Security Processor is operating in PCI-HSM mode, to sign the system and virtual NSP log files.

Command

```
<1204#>
```

Response

```
<2204#Certificate#Digital Signature#>
```

Calling Parameters

`1204`

Field 0, the command identifier.

Table 11-46. Command 1204: Get Log Signing Key Certificate

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1204

Responding Parameters

`2204`

Field 0, the response identifier.

`Certificate`

Field 1, the certificate of the log signing key. The format is:

`Digital Signature`

Field 2, the digital signature of the certificated signed by Atalla.

Table 11-47. Response 2204: Get Log Signing Certificate

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	2204
1	Certificate	3112	0 - 9, A - F
2	Days Remaining	512	0 - 9, A - F

Example

Using Command 1204 to get certificate.

The command looks like this:

```
<1204#>
```

The NSP returns a response similar to this:


```
<2204#4A4C30313448000000030100010100EFFF6306B4D9D4374B0464867
B4E3E5A3879B0232E3F51BD543B5DC9F04A3FEE2293E587A2DBA5EE7E68A9
F765748489844340EE3E723D8F0025AAFBBAB8ECB18FB9B6C789DABAFB2C5
B15482FE7572B1076C6044AAF69DCCB779796DD8ECC751D3FE573E848EEE3
C3B00ABEF9432E73742EB5AE02F06C8C0EA98CE272F687AC181BF8AFF0AE2
4F076B36A65996E08C9D828BFF6A76F201A72EE89423AE3F529BC8987A98A
441C34971926256FFD6BB305B1BF89F81486740730293EBB533E10C333029
B32B49CD196BD19E916F5DBEE62C7B96BA150F9754BDE5F886E2CFE7AECBF
BC25565EE89F28FEB9072B6DEA21F4F6217FE9C60E685AC1F5F490D8505F0
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000001070000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000001DD02803B83C06331
1738AC6EEB5E2D6D75A9F0751454F4624EA6D9E836A5BBAE33F2AAB59D0EE
2EF50C86992BFFA9884FA3A83CE3A467303912827454BED837ED61C4E07B9
D5A07A799416C4C55822F1BE766F60D56F68B8A1FD2E9EBAD0961D159A15D
```

```
C77250C1FC6299539FAD8E1FD7DB5813A2836BDF7103B5BD53E724CCDEBDD
580C3955713E61B952AB2E82251CDA0F86ED26EBDBEAA66DAB46F8B03B824
DDEFAA8FC8E0F58D123FA6B141FDADC33EA74FCED066904491C559559DBCB
CE0BE8CD5F06B38660F731964EB5238C3B3F10314242260BF6A50FD478975
2DA1FBA56ABEDB2A93A9B4D9DA7D462476136F5AC1BF0FAB7515404835B49
9ABB517#3CAF686981727329D21993E3E81346F4C835D7461905A9D1EB1B2
F95B639D8A73A280463E4C8F3C96761E0723A76F5925471A7905EF31EDC42
CC552DDA90867B1E7D2F3F48BAB3AE3B16674EF3672468B246641A0DDAB42
7A34A4D7E7B070B5AB3AF5AF03E6574F2AA5BCF249A05D54AD3FA2840D75D
846AB1061BD6F74347AD95C619CA15EA0278D3E25759A9025F9D17F4592F8
0F4A0AE26A25E326889229977A839880FC59FFEA32C8AD7C3B1E99E940E47
55B256E3354EAC4CAB822E8DFCCACC77C644D8D6A1210802A3507BBAA3520
02753C705690A6B1BDA8830843ED7B268E930729C9E9CA62C42B5F3FB2765
2B6226209DC07F7E6A34A0696FC06E04#>
```

Get Battery Life Remaining (Command 1216)

Command 1216 – Use this command to return the expected number of days remaining for the internal batteries. Before this value reaches zero the batteries should be replaced by a HP technician.

Command

```
<1216#1#>
```

Response

```
<2216#1#Days Remaining#>
```

Calling Parameters

`1216`

Field 0, the command identifier.

`1`

Field 1, the sub-command: Get Battery Life Remaining.

Table 11-48. Command 1216: Get Battery Life Remaining

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1216
1	1	1	1

Responding Parameters

`2216`

Field 0, the response identifier.

`1`

Field 1, the sub-command specified by the command request.

`Days Remaining`

Field 2, the number of days remaining of battery life, in the range of 0 - 700. The value is the approximate number of days until the battery voltage level drops below an acceptable level.

Table 11-49. Response 2216: Get Battery Life Remaining

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	2216
1	1	1	1
2	Days Remaining	4	0-9

Example

Use Command 1216 to get battery life remaining.

The command looks like this:

```
<1216#1#>
```

The NSP returns a response similar to this (which indicates that there are 200 days of battery life remaining):

```
<2216#1#200#>
```

Return IP Address of NSP (Command 1221)

Command 1221 – Use this command to return the IP Address of the Network Security Processor.

Command

```
<1221#>
```

Response

```
<2221#NIC1 IP Address#[NIC2 IP Address#]>
```

Calling Parameters

`1221`

Field 0, the command identifier.

Table 11-50. Command 1221: Return IP Address of NSP

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1221

Responding Parameters

`2221`

Field 0, the response identifier.

`NIC1 IP Address`

Field 1, the NIC1 IP Address of Network Security Processor.

`[NIC2 IP Address#]`

Field 1, the NIC2 IP Address of Network Security Processor. This field will be present when:

- Option 87 has been enabled in the Network Security Processor's security policy.
- The Network Security Processor was powered on with a config.prm file that contained a valid IP address in the TCPIP parameter IPADDR_2.

Table 11-51. Response 2221: Return IP Address of NSP

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	2221
1	NIC1 IP Address	varies	0 - 9
2	[NIC2 IP Address]	varies	0 - 9

Examples

The command looks like this:

```
<1221#>
```

When only one NIC is configured, the Network Security Processor returns a response similar to this:

```
<2221#127.0.0.1#>
```

When both NICs are configured, the Network Security Processor returns a response similar to this:

```
<2221#127.0.0.1#127.0.1.2#>
```

TCP/IP Socket Information (Command 1223)

Command 1223 – Use this command to return the number of sockets on the Network Security Processor that are available for new connections, the total number of new sockets the Network Security Processor can open, and the number of sockets available for reconnect from the host that sent this command.

In version 1.14 and above option 023 applies to this command. All of the other options that can be defined by command 101 do not apply to this command.

Command

```
<1223#[Port#][NIC#]>
```

Response

```
<2223#Remaining Sockets#Total Sockets#Reconnect Sockets#>
```

Calling Parameters

`1223`

Field 0, the command identifier.

`[Port#]`

Field 1, this field is optional if field 2 is not included in the command. This field is required if field 2 is included in the command. If present, socket information will be returned in the response for the port number specified in this field (PORT_ASCII, PORT_STATUS or PORT_MANAGEMENT). If the command is received by the NSP on either the serial or USB port, and field 2 is not specified, socket information will be returned in the response for the NIC1 port number specified in this field.

If this field is not present, and the command was received by either NIC1 or NIC2, socket information on the port number that received the command is returned in the response. If the command is received by the NSP on either the serial or USB port, and this field is not present, socket information for the NIC1 PORT_ASCII will be returned in the response.

`[NIC#]`

Field 2, this field is optional. If present, socket information will be returned in the response for the NIC specified in this field (1 or 2).

If this field is not present, socket information for the NIC that received the command is returned in the response. If the command is received by the NSP on either the serial or USB port, socket information will be returned in the response for the NIC1 port number specified in field 1 of the command.

Table 11-52. Command 1223: TCP/IP Socket Information

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1223
1	[port#]	1-5	0-9
2	[NIC#]	1	1 or 2

Responding Parameters

`2223`

Field 0, the response identifier.

`Remaining Sockets`

Field 1, the number of sockets available for a new connection. This value is the difference between the total number of sockets minus the number of sockets opened on the Network Security Processor. For example, if the total number of sockets is 16, and the application running on host A has 6 open sockets on the Network Security Processor, and the application running on host B has 3 open sockets on the Network Security Processor, this field would contain the value 7.

`Total Sockets`

Field 2, the number of TPC/IP sockets available for use in the Network Security Processor. This value is the number of sockets specified in the MAX_CLIENTS_ASCII parameter in the CONFIG.PRM file. If this parameter is not specified in the file the maximum number of sockets is 16.

`Reconnect Sockets`

Field 3, the number of reconnect sockets is equal to the number of Network Security Processor sockets that are connected to the host that sent the <1223#> command. If reconnect sockets are not enabled, this field will contain the letter “X”.

If a specific host establishes 10 new socket connections with the Network Security Processor, then it has 10 reconnect sockets available. Reconnect sockets are used by then Network Security Processor only when all available sockets are in use.

Here is an example of when reconnect sockets are used. Assume that the Network Security Processor is configured for 16 sockets, and host A has 11 sockets open on the Network Security Processor, and host B has 3 sockets open on the Network Security Processor; 14 of the 16 possible sockets in use on the Network Security Processor.

Host B loses power which leaves 3 sockets on the Network Security Processor in a hung state. Host B is immediately restarted and attempts to reconnect to the Network Security Processor, the first 2 socket open requests from Host B will be granted as new socket connects by the Network Security Processor, whereas the

third socket open request from Host B will be granted as a reconnect socket by the Network Security Processor.

Any attempts by a host other than A or B to connect with the Network Security Processor will fail as all available sockets are in use (Host A is using 11, and Host B is using 5, three of which are hung). At this point, Host A has 11 reconnect sockets available, and Host B has 2 reconnect sockets available.

Assume that another application on Host A now tries to establish 12 socket connections to the Network Security Processor, only the first 11 will be granted as reconnect sockets, the 12th open request will fail as there are no available sockets on the Network Security Processor and Host A has used all of its reconnect sockets.

Assume another application on Host B now tries to establish 4 socket connections to the Network Security Processor, only the first 2 will be granted as reconnect sockets, the last 2 open requests will fail as there are no available sockets and Host B has used all of its reconnect sockets.

After the KEEP_ALIVE_TIME expires (default is 20 minutes) the Network Security Processor detects that the 3 sockets originally established with Host B are hung, they are deleted from the Network Security Processor, and the number of available sockets is set to 3.

Table 11-53. Response 2223: TCP/IP Socket Information

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	2223
1	Remaining Sockets	1-2	0-9
2	Total Sockets	1-2	0-9
3	Reconnect Sockets	1-2	0-9

Example

Use Command 1223 to return the number of available sockets on the Network Security Processor.

The command looks like this:

```
<1223#>
```

The Network Security Processor returns the following response:

```
<2223#10#16#6#>
```

which indicates that there are 10 sockets available (six sockets in use on the Network Security Processor) out of a total of 16, and for this host there are six sockets available for reconnect.

Get Application Key Check Digits (Command 1226)

Command 1226 – Use this command to obtain the check digits of the application keys you have loaded into the Network Security Processor non-volatile key table. The application keys include the MFK and PMFK. Check digits are returned for keys loaded into the Network Security Processor non-volatile key table.

Command

```
<1226#>
```

Response

```
<2226#[MFK1=xxxx]#[PMFK1=xxxx]#####>
```

Calling Parameters

`1226`

Field 0, the command identifier.

Table 11-54. Command 1226: Get Application Key Check Digits

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1226

Responding Parameters

`2226`

Field 0, the response identifier.

`Key Names and Check Digits`

Field 1, the names of the application keys and their check digits. If you have not loaded an application key its name and check digits will not be displayed.

Table 11-55. Response 2226: Get Application Key Check Digits

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	2226
1	Key names and check digits.	Variable	0-9, A-F, “=”, “MFK1”,“PMFK1”

Example

Using Command 1226 to obtain check digits.

The command looks like this:

```
<1226#>
```

The Network Security Processor returns a response similar to this.

```
<2226#MFK1=057A######>
```

Reset to Factory State (Command 1227)

Command 1227 – Use this command to start a process to reset the Network Security Processor to the factory state. The response to this command is a random value that must be sent in command 1228 to confirm your request to reset the Network Security Processor to the factory state.

This command pair erases the user-defined security policy, all user-defined keys, and optionally erases the security audit log. The default security policy is restored.

You should use this command only when you need to add the Network Security Processor to a security association or remove it from service.

Command

Response

Calling Parameters

`1227`

Field 0, the command identifier.

`RESET_TO_FACTORY_STATE`

Field 1, statement sent to the Network Security Processor.

`[MODE#]`

Field 2, an optional field. When this field is present and contains the word “CLEAR”, the Network Security Proce he eie

Responding Parameters

`2227`

Field 0, the response identifier.

`nnnnnn`

Field 1, representing the six random digits returned by the Network Security Processor. These digits are used in field one of command 1228 to confirm and complete the reset process. If you provide the incorrect value in the 1228 command you will need to repeat the 1227 command again to generate a new random value.

Table 11-57. Response 2227: Reset to Factory State

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	2227
1	Random digits	6	0-9, A-F

Examples

The following example illustrates using Command 1227 to reset the Network Security Processor to factory state.

The command looks like this:

```
<1227#RESET_TO_FACTORY_STATE#>
```

The Network Security Processor issues a response similar to this:

```
<2227#nnnnnn#>
```

The following example illustrates using Command 1227 to reset the Network Security Processor to factory state and clear the security audit log.

The command looks like this:

```
<1227#RESET_TO_FACTORY_STATE#CLEAR#>
```

The Network Security Processor issues a response similar to this:

```
<2227#nnnnnn#>
```

Confirm Reset to Factory State (Command 1228)

Command 1228 – Use this command to complete the reset of the Network Security Processor to factory state.This command erases the user-defined security policy, all user-defined keys, and optionally erases the security audit log. At the completion of the process the Network Security Processor's default security policy is restored.

Prior to sending this command to the Network Security Processor you must first use command 1227 to generate the input for field 1.

Note. This command is only allowed on either the USB or serial port.

Command

```
<1228#nnnnnn#>
```

Response

```
<2228#status#>
```

Calling Parameters

`1228`

Field 0, the command identifier.

`nnnnnn`

Field 1, the six random digits from field 1 of the response 2227.

Table 11-58. Command 1228: Confirm Reset to Factory State

Field #	Contents	Length (bytes)	Legal Characters
0	Command identifier	4	1228
1	nnnnnn	6	0-9, A-F

Responding Parameters

`2228`

Field 0, the response identifier.

`status`

Field 1, the result of processing the command.

ok, indicates confirmation that the Network Security Processor is in the Factory state.

Bad Confirmation, indicates that the wrong value was entered. You must repeat the command 1227 again to obtain a new random value.

Table 11-59. Response 2228: Confirm Reset to Factory State

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2228
1	ok	2	ok

Example

The following example illustrate using Command 1228 to confirm the reset to factory state.

The command looks like this:

```
<1228#nnnnnn#>
```

The Network Security Processor returns the following response:

```
<2228#ok#>
```

Select Virtual NSP (Command 1350)

Use this command to select which virtual NSP will process the commands sent from the USB or serial port.

Note. This command is only allowed on the USB or serial port. It should only be sent to an A10160 NSP that is configured for mutiple virtual NSPs.

Command

```
<1350#VNSPx#>
```

Response

```
<2350#status#>
```

Calling Parameters

`1350`

Field 0, the command identifier.

`VNSPx`

Field 1, the virtual NSP number to be selected. Values can be VNSP0, VNSP1, VNSP2, VNSP3, VNSP4, VNSP5, VNSP6, VNSP7, VNSP8, or VNSP9.

Table 11-60. Command 1350: Select Virtual NSP

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	1350
1	VNSPx	5	VNSPx, x=0-9

Responding Parameters

`2350`

Field 0, the response identifier.

`status`

Field 1, the status response can be either "ok" if successful, or "not defined" if a virtual NSP is specified in the range of 1-9 and it is not defined in the "config.prm" file. If multiple-VNSP support is not enabled, only "VNSP0" is allowed.

Table 11-61. Response 2350: Select Virtual NSP

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2350
1	status	2, 11	ok , not defined

Example

The following example illustrates selecting virtual NSP3.
The command looks like this:

```
<1350#VNSP3#>
```

The Network Security Processor returns the following response:

```
<2350#ok#>
```

If the virtual NSP has not been defined in the config.prm file the Network Security Processor returns the following response:

```
<2350#not defined#>
```

Virtual NSP System Information (Command 1351)

Use this command to return the name if defined, and Master File Key check digits (MFKCDx) for each defined virtual NSP. If the virtual NSP has not been defined, the entire field for that virtual NSP will be empty. Virtual NSP 0 will always be present. If multiple-VNSP support is not enabled, only information for VNSP0 will be returned.

Note. This command is only allowed on the USB or serial port.

Command

```
<1351#>
```

Response

```
<2351#VNSP0=NAME0(MFKCD0)#[VNSP1=NAME1(MFKCD1)]#
[VNSP2=NAME2(MFKCD2)]#[VNSP3=NAME3(MFKCD3)]#
[VNSP4=NAME4(MFKCD4)]#[VNSP5=NAME5(MFKCD5)]#
[VNSP6=NAME6(MFKCD6)]#[VNSP7=NAME7(MFKCD7)]#
[VNSP8=NAME8(MFKCD8)]#[VNSP9=NAME9(MFKCD9)]#>
```

Calling Parameters

`1351`

Field 0, the command identifier.

Table 11-62. Command 1351: Virtual NSP System Information

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	1351

Responding Parameters

`2351`

Field 0, the response identifier.

`VNSP0 Information`

Field 1, the name and (MFK check digits) for VNSP0.

`[VNSP1 Information]`

Field 2, the name and (MFK check digits) for VNSP1.

`[VNSP2 Information]`

Field 3, the name and (MFK check digits) for VNSP2.

`[VNSP3 Information]`

Field 4, the name and (MFK check digits) for VNSP3.

`[VNSP4 Information]`

Field 5, the name and (MFK check digits) for VNSP4.

`[VNSP5 Information]`

Field 6, the name and (MFK check digits) for VNSP5.

`[VNSP6 Information]`

Field 7, the name and (MFK check digits) for VNSP6.

`[VNSP7 Information]`

Field 8, the name and (MFK check digits) for VNSP7.

`[VNSP8 Information]`

Field 9, the name and (MFK check digits) for VNSP8.

`[VNSP9 Information]`

Field 10, the name and (MFK check digits) for VNSP9.

Table 11-63. Response 2351: Virtual NSP System Information

Field #	Contents	Length (bytes)	Legal Characters
0	Response identifier	4	2351
1	VNSP0 Information	varies	0 - 9, A -Z, a -z, _
2	[VNSP1 Information]	varies	0 - 9, A -Z, a -z, _
3	[VNSP2 Information]	varies	0 - 9, A -Z, a -z, _
4	[VNSP3 Information]	varies	0 - 9, A -Z, a -z, _
5	[VNSP4 Information]	varies	0 - 9, A -Z, a -z, _
6	[VNSP5 Information]	varies	0 - 9, A -Z, a -z, _
7	[VNSP6 Information]	varies	0 - 9, A -Z, a -z, _
8	[VNSP7 Information]	varies	0 - 9, A -Z, a -z, _
9	[VNSP8 Information]	varies	0 - 9, A -Z, a -z, _
10	[VNSP9 Information]	varies	0 - 9, A -Z, a -z, _

Example

The following example illustrates the system information returned for the 5 defined virtual NSPs.

VNSP0 name : PROD

VNSP0 MFK Check Digits: B196

VNSP1 name : UAT

VNSP1 MFK Check Digits: 057A

VNSP2 name :

VNSP2 MFK Check Digits: 1234

VNSP5 name: TEST5

VNSP5 MFK Check Digits:

VNSP9 name : DEV

VNSP9 MFK Check Digits: 5678

The command looks like this:

```
<1351#>
```

The Network Security Processor returns the following response:

```
<2351#VNSP0=PROD(B196)#VNSP1=UAT(057A)#VNSP2=(1234)###VNSP5=T
EST5()####VNSP9=DEV(5678)#>
```

If the Network Security Processor is not configured to support multiple virtual NSPs the response to this command will be similar to this:

```
<2351#VNSP0=(B196)##########>
```

12 Error Messages

Application Error Messages

If the Network Security Processor encounters a command syntax error, an error response message is returned. The format of the error response is:

<00#XXYYZZ#>

The response ID of 00 indicates an error is being returned.

Table 12-1 lists the error number and its description that is returned in field XX.

Note. When xx = 04, the next two digits (yy) indicate the total number of fields that were expected in the command.

Table 12-1. Error Types (page 1 of 2)

Error #	Description
00	Response to test message
01	Length out of range
02	Invalid character
03	Value out of range
04	Invalid number of parameters
05	Parity error
06	Key usage error
07	Non-existent key
08	Execution error or self-test failure
09	Expecting 1key-3DES key
10	Key length error
11	Printing error
12	Marker string not found
20	Serial number set, cannot modify it
21	NSP is not in a Security Association, or Serial number not set
22*	Non-existent command or option
23*	Invalid command or option
24	Incorrect challenge
25	Incorrect Acknowledgement
26*	Duplicate command or option
27	No challenge to verify, a command 109 has been received without a prior command 108
28	Configuration text exceeds maximum length

Table 12-1. Error Types (page 2 of 2)

Error #	Description
29	Cannot allocate memory
41	ASRM timed out waiting for the NSP response
73	Variant mismatch
92	Security association error
93	Factory keys already generated
94	No factory keys generated

* If this error is generated when processing security policy commands, the error response will include an additional field after the XXYYZZ field. This additional field will contains the first item found in error.

YY – the first field found to be in error.

Note. Due to the Network Security Processor's parsing logic the field reported in the error response may not be the first, or only, field in the command that contains an error.

If this field returns the value 00, then any of the following may be true:

- The command specified an invalid command number.
- A necessary MFK or KEK is missing.
- The response has been sent simply as an echo of a command.

ZZ – the software revision level of the cryptographic command processor.

Detailed Errors

The detailed error is appended as a separate field after the error field (XXYYZZ). Detailed errors are only included if option 21 is enabled, see Configure Security Processor Option (Command 101) on page 11-29 for more information on enabling detailed application errors. Table 12-2 lists the detailed application error messages by number, and provides the description of each message.

Table 12-2. Detailed Application Errors (page 1 of 5)

Error #	Description
1	Invalid command string length
2	Invalid command length
3	Invalid parameter length
4	Passcode length not matched with user data
5	Non empty field - conflicts with other fields
95	Internal error
100	Invalid character error

Table 12-2. Detailed Application Errors (page 2 of 5)

Error #	Description
101	Invalid command string format
102	Invalid character
200	Value out of range
201	Invalid command
202	Invalid parameter value
203	Command not implemented
204	Invalid continue command (5B)
205	Invalid part/length for loading keys from the key loading module
206	Invalid restriction setting
207	Invalid table type specified for loading a key
208	Invalid parent key
209	Invalid key length specified
210	Invalid key name specified
211	Invalid ANSI-formatted message authentication code
214	Invalid key serial number, if new one is the same as the current one
215	Invalid checksum on string
216	Value in field is not same as other field
217	Count value not greater than zero
218	Command count table is full
220	No free key slot for RSA key
300	Invalid number of parameters
301	Too many fields
302	Too few response fields
303	Too few fields
304	Initialization vector is missing
305	Wrong combination of keys
306	Invalid number of parameters
500	Application error
501	Key table entry in use
502	Key table full
503	MFK is not valid
504	KEK is not valid
505	MFK already exists
506	KEK already exists

Table 12-2. Detailed Application Errors (page 3 of 5)

Error #	Description
507	Error during key loading process
508	KEK check digits do not match expected check digits
509	Key did not have odd parity
510	Specified variant cannot be used
511	KD1 or KD2 check digits do not match expected check digits
512	Wrong entry of 1key-3DES key
513	Command 14-5, keys have different length
514	Command 14-5, weak key
515	Any decimalization tables in the key table must be 1key-3DES
600	Non-existent key
601	Non-existent module key entry
602	Non-existent MFK
603	Non-existent KEK
604	Non-existent Pending MFK
605	Incorrect entry of 2key-3DES key slots
606	Pending MFK name is the same as the current MFK's name
607	Security violation
608	Non-existent configuration key
611	MFK name in command does not match the current or retired name in the security processor
612	MFK name in command does not match the MFK name in the security processor
613	Pending MFK name in command does not match the pending MFK name in the security processor
620	The variant is incorrect
622	The MAC of the AKB did not verify
623	Key Slot empty
700	Hardware error
701	Cannot open file
702	Problem with EDES_ENC
704	Problem in routine des_cbc_cfb8
705	Problem in routine des_ofb_cfb64
706	Hardware error
707	Fatal error
708	A routine which should always * (Return didn't *)

Table 12-2. Detailed Application Errors (page 4 of 5)

Error #	Description
709	DCP NVRAM error
710	FEB NVRAM error
711	Internal routines returned unsuccessfully
712	Wrong mode
713	Internal developer's error
714	BSAFE error
715	DUKPT error
716	Random number generator error
717	Deterministic Random Bit Generator error
718	Command not allowed in PCI-HSM mode
801	Failed hardware function
802	Failed ACE function (general)
803	Failed ACE function (command buffer too big)
804	Failed ACE function (LDM function failed)
805	Failed ACE function (Response returned smaller than minimum)
806	Failed ACE function (Response length invalid)
807	Failed ACE function (Response ID incorrect)
808	Failed ACE function (Response ID had invalid error)
809	Failed ACE function (Command had NULL error)
810	Failed ACE function (Command had NULL first item)
811	Failed ACE function (Response had NULL item)
812	Failed ACE function (Response had NULL first item)
813	Failed ACE function (Command ID was modified)
901	Expecting a 1key-3DES key and received a 2key-3DES
902	Expecting a 2key-3DES key and received a 1key-3DES
903	The 2key-3DES key is really a replicated 1key-3DES key
1100	No continuation indexes are available
1101	Specified continuation index is empty
1102	Invalid print job length
1103	Unable to obtain a socket on the printer
1104	Unable to connect to printer
1105	Unable to send print job to printer, error returned from printer
1200	Marker string not found
2000	The Serial number is already set, it cannot be modified

Table 12-2. Detailed Application Errors (page 5 of 5)

Error #	Description
2100	The Serial Number is not loaded
2101	NSP is not in a security association
2200	Non-existent command item in the configuration string
2300	Invalid command item format
2301	Command 105 must be sent first.
2400	Incorrect value in command 109
2500	The acknowledgment text is incorrect or missing
2600	Conflicting duplication of a configuration parameter
2700	Command 109 was received before command 108
2800	Configuration text exceed maximum length
2900	Unable to allocate memory
7300	The variant of the key in table incorrect
7301	The variant for a decimalization table is wrong
9200	System was not initialized
9201	RSA keys already exists
9202	Autokey global data is corrupted
9203	Can't allocate memory with mymalloc
9205	Failed signature verification
9208	Failed certificate verification
9210	Can’t sign the certificate or bad signature
9211	The NSP does not have a security association
9212	No session key present in a system
9213	MAC computation or verification failed
9214	Bad Tx buffer data length
9215	Bad data length
9216	Bad transaction function
9217	Bad transaction type
9218	Bad transaction state

Examples

Receiving Response 00 due to an Error Condition

The command being sent is 72 verify key table slot. It contains an invalid value for the key slot.

```
<72#56780#>
```

The Network Security Processor issues the following response.

```
<00#030120#>
```

This response indicates the following:

- The field's value is out of range (indicated by 03).
- Field 1 is in error (indicated by 01).
- The software's revision number is 2.00.

If the detailed error feature (option 21) is enabled the response is:

```
<00#030120#0202#>
```

This response indicates the following:

- The field's value is out of range (indicated by 03).
- Field 1 is in error (indicated by 01).
- The software's revision number is 2.00.
- The detailed error (0202) indicates an invalid parameter value.

A Introduction to Cryptography

In 1973, the National Institute of Standards Technology (NIST) approved the use of an algorithm, the Data Encryption Algorithm (DEA), for providing data security in communications systems. The algorithm is commonly known as the Data Encryption Standard (DES).

Data Encryption Standard (DES)

DES provides the data processing industry with a standard encryption technique that is acceptable in financial applications. See Federal Information Processing Standard 46-3, and American National Standard Institute standard X9.32 for more information on DES. Triple DES is defined in American National Standard Institute standard X9.52.

Electronic Code Book (ECB)

In this mode, the DES unit uses a 56-bit key to encrypt or decrypt 64 bits of data and output ciphertext or plain text. This mode can be used to encrypt small data quantities that may be fixed to 64 bits length.

Cipher Block Chaining (CBC)

In this mode, the DES unit either encrypts or decrypts long strings of data blocks in multiples of eight bytes. The MACing commands 56, 98 and 99 use the CBC mode of DES. The Data Encrypt/Decrypt command 97 also supports CBC mode of DES.

Message Authentication

DES can also be used to provide message integrity; it insures that the original message was received by the recipient, without being altered. See Federal Information Processing Standard 113, International Standards Organization 8731, and American National Standards Institute X9.9 for more information.

Triple DES (3DES)

Triple DES utilizes three DES keys for encrypting/decrypting data. The combination of all three keys is referred to as a key component. This is referred to as a triple-length key or a 3key-3DES key.

Three DES cycles are used to either encrypt or decrypt information. When encrypting, the data is first encrypted using Key 1, the result is decrypted using Key 2, and this second result is encrypted using Key 3. The name 3DES indicates that there are three DES cycles used in the process (encrypt, decrypt, encrypt).

To decrypt data, the process is reversed (decrypt, encrypt, decrypt). When decrypting, the encrypted data is decrypted using Key 1, the result is encrypted using Key 2, and this second result is decrypted using Key 3.

Many systems use only 2 keys in a 3DES process. In this instance Key 1 is equal in value to Key 3, that is this value is used in the first and third step of the process. This is referred to as a double-length key or a 2key-3DES key.

It is also possible to use just one key in a 3DES process. In this instance Key 1 is equal in value to Key 2 and Key 3, that is this value is used in all three steps of the process. The result will be the same as using DES. This is referred to as a single-length key or a 1key-3DES key.

Figure A-1. TDEA Electronic Codebook

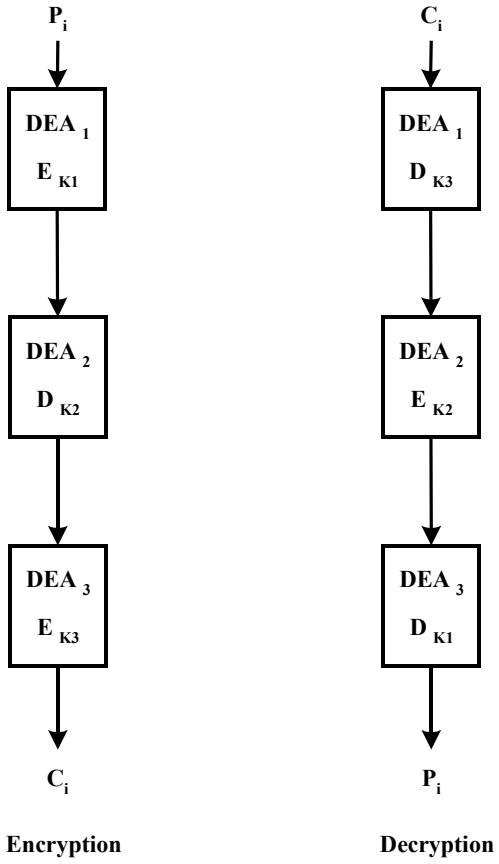


Figure A-2. TDEA Cipher Block Chaining - Encryption

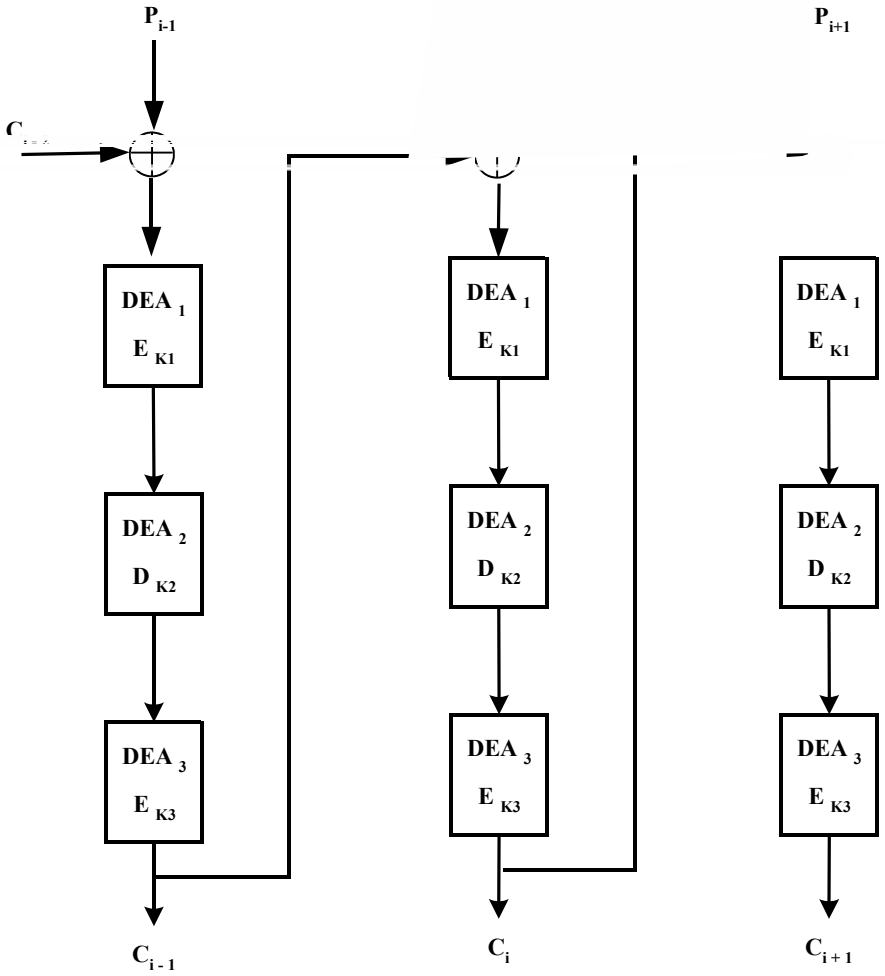
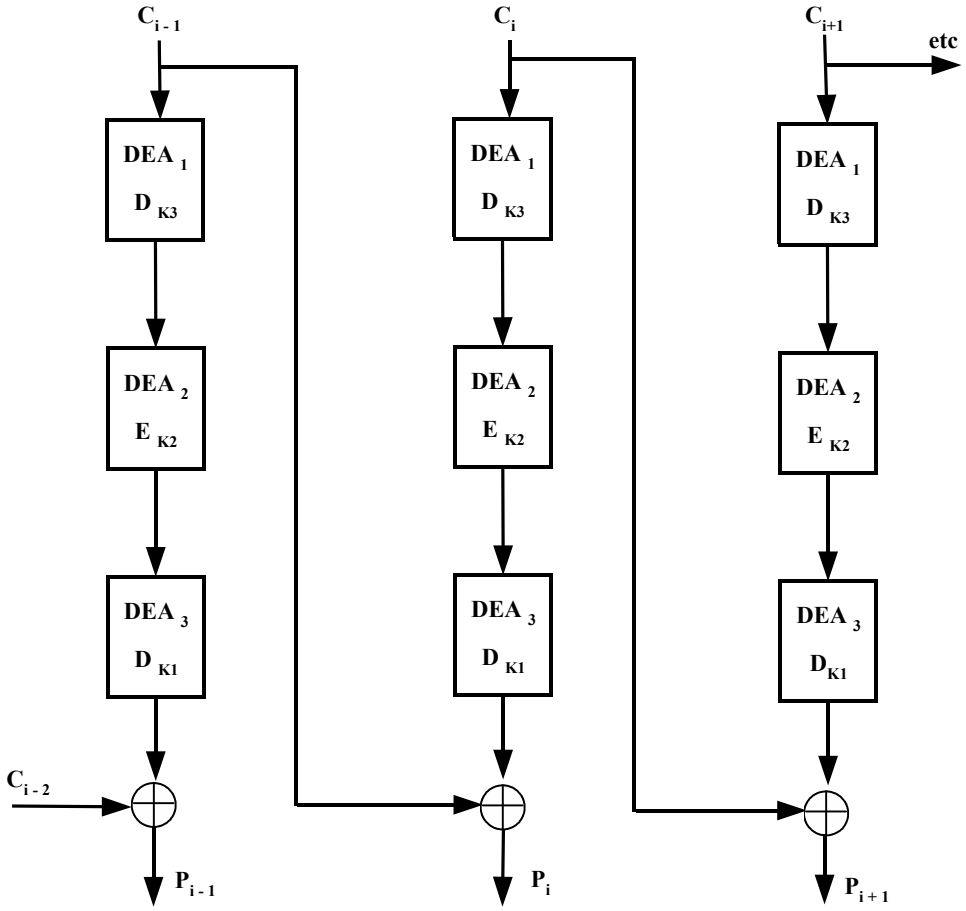


Figure A-3. TDEA Cipher Block Chaining - Decryption



Key Attributes

Key Length

The Network Security Processor supports DES keys that contain either 64 bits (single-length, 16 hexadecimal digits), 128 bits (double-length, 32 hexadecimal digits) or 192 bits (triple-length, 48 hexadecimal digits) of unique key material. DES keys use the eighth bit of each byte as a parity bit, this parity bit is not used in the encryption or decryption operation, therefore a single-length contains 56 bits, a double-length key contains 112 bits, and a triple-length key contains 168 bits, of unique keying material. See Key Parity for more information.

A triple-length key consists of three key blocks, each contain 16 hexadecimal characters, each block is a unique value, this is called a 3key-3DES key. A double-length key consists of two key blocks, each contain 16 hexadecimal characters, each block is a unique value, this is called a 2key-3DES key. A single-length key has only

one key block which contains 16 hexadecimal characters, this is called a 1key-3DES key.

The Network Security Processor requires the Master File Key, and Pending Master File Key to be either a 2key-3DES key (double-length) or a 3key-3DES key (triple-length).

Key Components

In a financial network, secret keys are used to encrypt sensitive data, such as a customer's PIN, as it flows through the network. To prevent any one individual from possessing the secret key, the secret key value is divided into key components. Key components are maintained by trusted individuals for entry into the Network Security Processor. A multi-component key increases security because a different person is assigned to create each component. This way, nobody knows the value of the entire key, reducing the possibility of a security breach. Once all the key components have been entered, the Network Security Processor combines them into a final secret key value. This secret key value can then be used to either encrypt or decrypt information as it passes through the network.

Each 3DES key can have up to four key components. When you define a 3DES key, you are prompted for the number of key components. When all key components have been entered, they are automatically combined into one secret key value. A 3key-3DES key has three key blocks for each key component. A 2key-3DES key has two key blocks for each key component. A 1key-3DES key has one key block for each component. :

3key-3DES Key (Triple-Length)			
Component (Check Digits)	**Block1**	**Block2**	**Block3**
Component 1 (35C1)	9205 48E6 FEB1 4A62	0BD1 45B6 6B72A 3BB	5865 2863 425A3 8A9
Component 2 (53B3)	B8B9 7509 BBD6 4BEB	93C1 33F3 95A1 6819	5946 6D04 CBF1 F546
Final Key (B196)	2ABC 3DEF 4567 0189	9810 7645 FED3 CBA2	0123 4567 89AB CDEF

2key-3DES Key (Double-Length)

Component	**Block 1**	**Block 2**
Component 1 (2E0D)	C8F4 BD02 FD31 FFEE	674D 1508 5489 4275
Component 2 (3178)	E248 80ED B856 FE67	FF5D 634D AA5A 89D7
Final Key (057A)	2ABC 3DEF 4567 0189	9810 7645 FED3 CBA2

1key-3DES Key (Single-Length)

Component	Block 1
Component 1 (D5D4)	0123 4567 89AB CDEF
Component 2 (8422)	DDF2 C8B7 AA6C 4DBF
Final Key (DB8E)	DCD1 8DD0 23C7 8050

Key Parity

Most cryptographic systems require keys to be odd parity. This means that when a pair of hexadecimal characters (1 byte) is converted to binary format, the result contains an odd number of one bits. Here is an example of a key that is odd parity:

Clear-text key value: 01 23 45 67 89 AB CD EF

Each byte contains an odd number of one bits.

Byte Value	Binary Value	one bits
01	0000 0001	1
23	0010 0011	3
45	0100 0101	3
67	0110 0111	5
89	1000 1001	3
AB	1010 1011	5
CD	1100 1101	5
EF	1110 1111	7

To convert an even hexadecimal byte to odd parity, adjust the least significant bit of the rightmost character of the byte. This adjusts the parity without changing the value of the key. Here is an example of adjusting a key with several even bytes to odd parity.

Key Value: 12 34 56 78 90 AB CD FF

Byte Value	Binary Value	Adjusted Binary Value	Adjusted Byte Value
12	0001 0010	0001 0011	13
34	0011 0100	odd parity	34
56	0101 0110	0101 0111	57
78	0111 1000	0111 1001	79
90	1001 0000	1001 0001	91
AB	1010 1011	odd parity	AB
CD	1100 1101	odd parity	CD
FF	1111 1111	1111 1110	FE

The parity adjusted key value is: 13 34 57 79 91 AB CD FE

Weak and Semi-weak DES Keys

Table A-1 contains a list of DES key values that are not secure. For example, a key value of all zeros cannot be used to securely encrypt. If a weak key value is entered at the Secure Configuration Assistant -3 (SCA-3) a warning message is displayed. For production systems, avoid using weak key values. Note, an even parity key is not identified as a weak key.

Table A-1. Weak and Semi-weak Keys

Weak Keys	Semi-weak Keys
0101 0101 0101 0101	E001 E001 F101 F101
FEFE FEFE FEFE FEFE	01E0 01E0 01F1 01F1
E0E0 E0E0 F1F1 F1F1	FE1F FE1F FE0E FE0E
1F1F 1F1F 0E0E 0E0E	1FFE 1FFE 0EFE 0EFE
	E01F E01F F10E F10E
	1F0E 1F0E 0EF1 0EF1
	01FE 01FE 01FE 01FE
	FE01 FE01 FE01 FE01
	011F 011F 010E 010E
	1F01 1F01 0E01 0E01
	E0FE E0FE F1FE F1FE
	FEE0 FEE0 FEF1 FEF1

Sample Clear-text Key Component Form

****

Keep a copy of this completed form for your records, store securely, in a tamper evident envelope. Write the key name, key component number, date, and author on the exterior of the envelope.

Note: This individual must not have access to any other key component for this key.

Key Name _____ Key Component Number _____

Final Key Check Digits __ __ __ __

1. Enter the Key Block1 below:

__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __

2. Enter the Key Block2 below:

(Enter a value in this field if the key component is for a 2key-3DES key or a 3key-3DES key)

__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __

3. Enter the Key Block3 below:

(Enter a value in this field if the key component is for a 3key-3DES key)

__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __

4. Enter the component check digits below:

__ __ __ __

Name of Institution: _____

Generated by: _____ Date: _____

B Understanding Financial Interchange Networks

This section introduces financial interchange networks, the networks on which secure transactions travel, and outlines the tasks involved in initializing these networks.

Overview

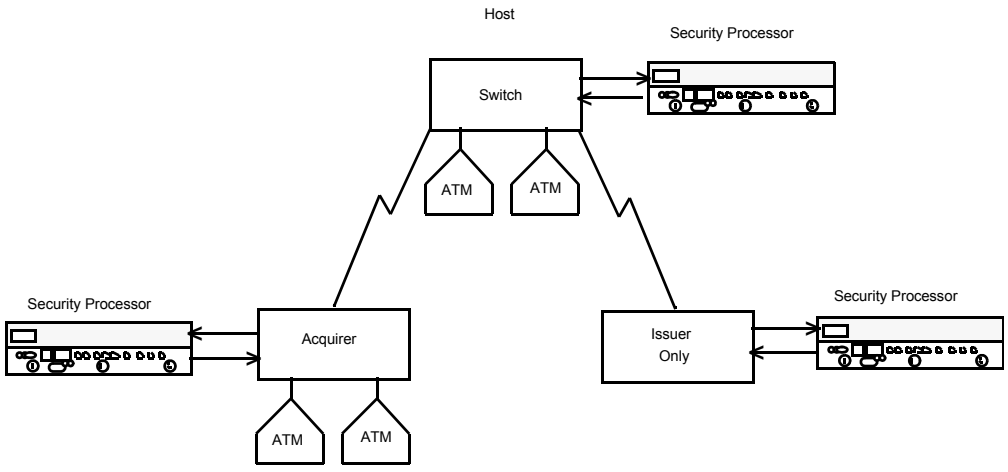
Financial interchange networks are computer networks that facilitate on-line funds transfers. This type of network consists of three primary components: acquirer nodes, issuer nodes, and switches.

- Acquirer node – the computer is attached to automated teller machines (ATMs) or PIN pads that introduce transactions into the network.
- Issuer node – the computer that belongs to the financial institution that has an account relationship with the consumer. An issuer can have ATMs or PIN pads attached to it, enabling it to act as both an issuer and an acquirer.
- Switch node – the computer that directs transactions from multiple acquirers to the appropriate issuer. A switch can have ATMs or PIN pads attached to it.

Atalla network security processors can reside with each of these nodes to ensure the security of data as it travels from point to point within the network.

Figure B-1 illustrates a simple financial interchange network.

Figure B-1. Simple Financial Interchange Network



Initializing the Financial Interchange Network

This section explains the purpose of network initialization and describes typical network initialization tasks.

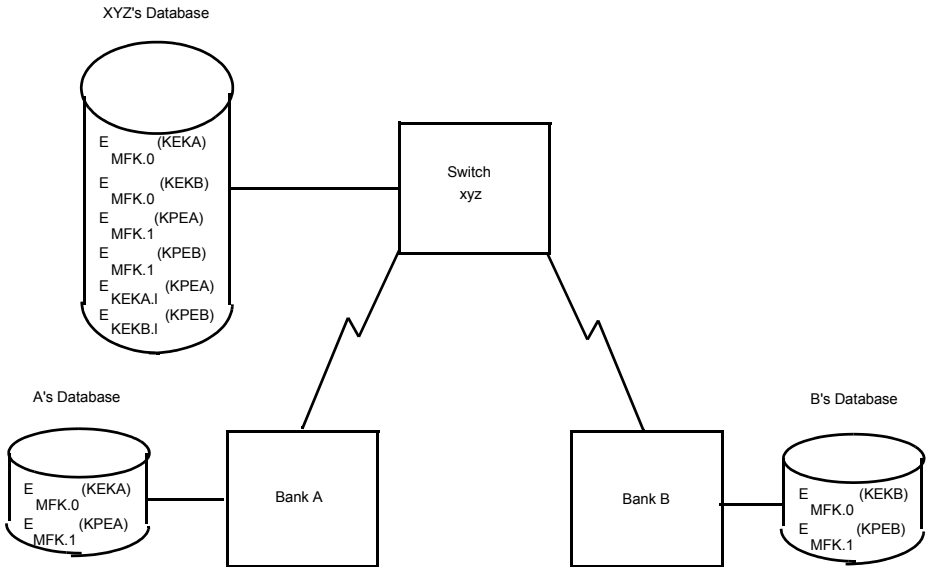
Purpose

Initialization refers to the process of establishing keys to be shared by a pair of participants on the financial network who agree to do business together. Establishing common keys facilitates transmitting data and decrypting encrypted messages.

For example, suppose that bank A decides to join switch XYZ, a switch that allows the customers of all participating financial institutions to transact business from any participant's ATM. In order for bank A customers to initiate transactions at, say, bank B, the switch must be able to translate the keys that bank B uses for encryption to keys that bank A recognizes and can decipher. To that extent, the switch must have access to both bank A's and bank B's Key Exchange and encryption keys. The switch may need to have access to other keys that each bank uses, too. The switch stores its members' Key Exchange Keys and encryption keys in its host database encrypted under its MFK.

Figure B-2 illustrates the key sharing necessary for bank A and bank B to do business on switch XYZ.

Figure B-2. Key Sharing



While this example may not resemble your network exactly, the idea of key sharing, as central to meaningful communication, is fundamental to any network that transmits secure data.

Initialization Checklist

The following list contains the tasks that are typically considered part of network initialization.

- Loading the Master File Key (MFK) into the security processor.
- Establishing the Key Exchange Key (KEK) that will be shared between nodes on the network.
- Establishing a PIN encryption key to be shared between ATMs or PIN pads and the host.
- Establishing a PIN encryption key to be shared between nodes.
- Establishing other working keys to be shared between nodes.

Establishing common PIN encryption keys and other working keys is explained below.

Establishing a PIN Encryption Key to be Shared by ATMs or PIN Pads and the Host

This subsection explains how to establish a PIN encryption key to be shared by ATMs or PIN pads and the host computer.

ATM-to-Host. The following list outlines the procedure for establishing a PIN encryption key to be shared by an ATM and a host computer.

1. Generate the ATM A key, the ATM B key, and the ATM master key.
2. Encrypt the ATM B key using the ATM A key.
3. Encrypt the ATM master key using the ATM B key.
4. Manually enter the ATM A key into the ATM.
5. Manually enter the cryptogram of the ATM B key into the ATM.
6. Program the ATM to decrypt the ATM B key using the ATM A key. When this process has finished, the clear text of the ATM B key will be in the ATM's memory.
7. Download the ATM master key from the host to the ATM, then decrypt it using the ATM B key. Now the host and the ATM share the ATM master key. When a consumer enters his or her PIN at this ATM, the PIN block will be encrypted using the ATM master key, then decrypted at the host using the same key.

PIN pad-to-host (using VISA DUKPT key management). VISA™ DUKPT key management specifies that the keys used to encrypt and transmit data be unique for every transaction. As such, the key used to initialize PIN-pad-to-host communication is used by the VISA DUKPT algorithm to derive unique transaction keys, but does not facilitate transactions on its own. The following list outlines the procedure for establishing a common PIN encryption key to be shared by a PIN pad and host computer.

1. Generate a derivation key.
2. Store the cryptogram of the derivation key in the host database.
3. Depending on the type of PIN pads being used on your network, you may want to maintain a database of each type's attributes. Alternatively, you can opt to receive this information as part of each transaction.
4. Load each PIN pad with an initial key serial number (IKSN).
5. Generate the initial PIN encryption key by encrypting the initial key serial number using the appropriate derivation key.
6. Load each PIN pad with an initial PIN encryption key.

Establishing a PIN Encryption Key to be Shared by Two Nodes

This subsection explains how to establish a PIN encryption key to be shared by two nodes. This procedure is the same for networks using VISA DUKPT key management and for those that do not.

1. The originating node generates a PIN encryption key using Command 10. The command returns two versions of the key: one to store locally, encrypted under MFK.1 and one to send to the other node, encrypted using the KEK that it shares with the receiving node. It then stores one cryptogram and sends the other cryptogram to the receiving node.
2. The receiving node stores the cryptogram on its host database.

Establishing Other Working Keys to be Shared by Nodes

The procedure for establishing any working key to be shared by two nodes is the same as the procedure for establishing a common PIN encryption key. See Establishing a PIN Encryption Key to be Shared by Two Nodes above for instructions on establishing common working keys.

C Summary of Commands and Options

Use this appendix as quick and easy way to jump to a specific command or option. Table Table C-1, Command Locator lists the Network Security Processor commands in numerical order. Table C-2, Network Security Processor Options, on page C-18 lists the Network Security Processor Options in numerical order. The Security Policy column in these tables contains one of the following three values:

- Standard - Commands and Options are ON in the Network Security Processor's default factory security policy. They can be disabled using either the SCA-3, or commands 108 and 109.
- Security Exposure - Commands and Options are OFF in the Network Security Processor's default factory security policy. They can be enabled using either the SCA-3, or commands 108 and 109.
- Premium Value - Commands and Options are OFF in the Network Security Processor's default factory security policy. They must be purchased in the form of a command 105. After the command 105 has been sent to the Network Security Processor, the security policy must be updated to include these commands and options. This is accomplished using either the SCA-3, or commands 108 and 109.

Note. Some premium value commands and options listed in the tables below were developed for specific customers. They are not available for use by any other customer. They are listed in these tables only because they are included in the response to command <9A#CONFIG-ALL#> and <9A#CONFIG-OFF#>.

If you have a need for a cryptographic command, whose syntax is not documented in this manual, please send an E-mail to AtallaOrders@hp.com. The E-mail must contain a detailed specification of your requirements including examples and test cases. Your specification will be reviewed and a cost estimate will be provided.

Table C-1. Command Locator (page 1 of 16)

Command	Name	Purpose	Security Policy	Page
00*	Echo Test Message	Tests the communications link between the host and the security processor.	Standard	11-4
10	Generate Working Key, Any Type	Generates a variety of working keys. The command returns the generated key in two forms: one for storing locally and one for transmitting to another node.	Standard	3-4
11	Translate Working Key for Distribution	Translates a working key from encryption using the Master File Key to encryption using the Key Exchange Key for transmitting to another network node.	Security Exposure	3-7
12	Translate Working Key for Local Storage	This command has been replaced by command 13.	Standard	n/a
13	Translate Working Key for Local Storage Switch-to-Switch	Translates a working key from encryption using the security processor's Key Exchange Key to encryption using the Master File Key for local storage and use.	Security Exposure	3-10
14	Load ATM Master Key – Diebold	Encrypts the ATM master key for downloading to Diebold ATMs.	Security Exposure	3-13
14	Load ATM Master Key – IBM 3624	Encrypts the ATM master key for downloading to IBM 3624 ATMs.	Security Exposure	3-16
14	Load ATM Master Key – IBM 4731	Encrypts the ATM master key for downloading to IBM 4731 ATMs.	Security Exposure	3-19
15	Change ATM Communications Key – Diebold	Encrypts a communication key for downloading to Diebold ATMS.	Security Exposure	3-23
15	Change ATM Communications Key – Docutel	Encrypts a communications key for downloading to Docutel ATMs.	Security Exposure	3-26
15	Change ATM Communications Key – IBM 3624	Encrypts a communications key for downloading to an IBM 3624 ATM.	Security Exposure	3-29

Table C-1. Command Locator (page 2 of 16)

Command	Name	Purpose	Security Policy	Page
15	Change ATM Communications Key – IBM 4731	Encrypts a communications key for downloading to an IBM 4731 ATM.	Security Exposure	3-32
16	Encrypt Financial Institution Table – Diebold	Encrypts keys for downloading to Diebold ATMs' financial institution tables.	Security Exposure	3-35
16	Encrypt Financial Institution Table – Docutel	Encrypts keys for downloading to Docutel ATMs' financial institution tables.	Security Exposure	3-38
16	Encrypt Financial Institution Table – IBM 3624	Encrypts keys for downloading to IBM 3624 ATMs' financial institution tables.	Security Exposure	3-41
17	Translate Working Key for Transfer to Primary from Secondary Node	This command has been replaced by command 11.	Standard	n/a
18	Generate VISA Working Key	Generates a working key for use with VISA security processors. The command returns the generated key in two forms: one for storing locally and one for transmitting to another node.	Security Exposure	3-44
19	Translate Communications Key for Local Storage	Translates a working key from encryption using a base key to encryption using the Master File Key for local storage.	Security Exposure	3-46
1A	Translate Working Key for Distribution to Non-Atalla Node	Translates a working key from encryption using the Master File Key to a Key Exchange Key.	Security Exposure	3-49
1C	Generate Session Key	Customer Specific Command	Premium Value	n/a
1D	Translate Communications Key for Local Storage Using a Specific Variant	Translates a working key from encryption using a base key without a variant to encryption using a Master File Key.	Security Exposure	3-52
1E	Generate New Initial Key for PIN Pad Using VISA UKPT	Re-initializes PIN Pads that perform VISA unique key per transaction key management	Premium Value	3-55
1F	Generate Token Key	Customer Specific Command	Premium Value	n/a

Table C-1. Command Locator (page 3 of 16)

Command	Name	Purpose	Security Policy	Page
30	Encrypt PIN	Encrypts a clear-text PIN.	Premium Value	4-23
31	Translate PIN	Translates a PIN from encryption under one key to encryption under ANSI format.	Standard	4-26
31	Translate PIN – VISA DUKPT	Translates an encrypted VISA DUKPT PIN to ANSI format.	Standard	4-30
32	Verify PIN – Identikey	Decrypts an incoming PIN and verifies it using the Atalla Identikey method of PIN verification.	Standard	4-35
32	Verify PIN – IBM 3624	Decrypts an incoming PIN and verifies it using the IBM 3624 method of PIN verification.	Standard	4-41
32	Verify PIN – VISA	Verifies PINs using the VISA verification method of PIN verification.	Standard	4-46
32	Verify PIN – Atalla DES Bilevel	Decrypts an incoming PIN and verifies it using the Atalla DES Bilevel method of PIN verification.	Standard	4-51
32	Verify PIN – Diebold	Decrypts an incoming PIN and verifies it using the Diebold method of PIN verification.	Standard	4-56
32	Verify PIN – NCR	Decrypts an incoming PIN and verifies it using the NCR method of PIN verification.	Standard	4-61
32	Verify PIN – Clear-PIN Comparison	Decrypts an incoming PIN and verifies it using the clear-PIN comparison method of verification.	Standard	4-67
32	Verify PIN – PIN-Block Comparison	Decrypts two incoming PIN blocks and compares their clear-text values.	Standard	4-70
32	Verify PIN - Atalla 2x2	Decrypts an incoming ANSI PIN Block and verifies it using the Atalla 2x2 PIN Verification Method	Standard	4-78

Table C-1. Command Locator (page 4 of 16)

Command	Name	Purpose	Security Policy	Page
33	Translate PIN – ANSI to PLUS and PLUS to ANSI	Translates an incoming, encrypted ANSI PIN to PLUS format. Also translates an incoming, encrypted PLUS PIN to ANSI format.	Security Exposure	4-82
33	Translate PIN – ANSI to PIN/Pad	Translates an incoming, encrypted ANSI PIN to PIN/Pad format.	Security Exposure	4-85
33	Translate PIN – ANSI to IBM 4731	Translates an incoming encrypted ANSI PIN to IBM 4731 format.	Security Exposure	4-88
33	Translate PIN – IBM 3624 to IBM 3624	Translates an incoming, encrypted IBM 3624 PIN to IBM 3624 format.	Security Exposure	4-92
33	Translate PIN – IBM 3624 to PIN/Pad	Translates an incoming, encrypted IBM 3624 PIN to PIN/Pad format.	Security Exposure	4-96
33	Translate PIN – PIN/Pad or Docutel to ANSI	Translates a PIN encrypted using PIN/Pad or Docutel format to ANSI format.	Security Exposure	4-100
33	Translate PIN – PIN/Pad or Docutel to PIN/Pad	Translates an incoming, encrypted PIN/Pad or Docutel PIN to PIN/Pad format.	Security Exposure	4-103
33	Translate PIN – PIN/Pad or Docutel to IBM 4731	Translates an incoming, encrypted PIN/Pad or Docutel PIN to IBM 4731 format.	Security Exposure	4-106
33	Translate PIN – IBM 4731 to ANSI	Translates an incoming, encrypted IBM 4731 to ANSI format.	Security Exposure	4-110
33	Translate PIN – IBM 4731 to PIN/Pad	Translates an incoming, encrypted IBM 4731 to PIN/Pad format.	Security Exposure	4-114
33	Translate PIN – IBM 4731 to IBM 4731	Translates an incoming, encrypted IBM 4731 to IBM 4731 format.	Security Exposure	4-118
34	PIN Translate	Customer Specific Command	Premium Value	n/a
35	Translate PIN, Double-Encrypted Input or Output	Decrypts and re-encrypts an encrypted PIN, where the input or output is double encrypted.	Security Exposure	4-122

Table C-1. Command Locator (page 5 of 16)

Command	Name	Purpose	Security Policy	Page
36	Verify Double-Encrypted PIN	Decrypts an incoming double-encrypted PIN and verifies it according to the specified method.	Security Exposure	4-126
37	PIN Change – Identikey	Verifies the old PIN using the Atalla Identikey method.	Premium Value	4-129
37	PIN Change – IBM 3624	Verifies the old PIN using the IBM 3624 method.	Premium Value	4-135
37	PIN Change – VISA	Verifies the old PIN using the VISA method.	Premium Value	4-141
37	PIN Change – Atalla DES BiLevel	Verifies the old PIN using the Atalla DES BiLevel method.	Premium Value	4-146
37	PIN Change – Diebold	Verifies the old PIN using the Diebold method.	Premium Value	4-152
37	PIN Change – NCR	Verifies the old PIN using the NCR method.	Premium Value	4-157
38	PIN Change	Customer Specific Command	Premium Value	n/a
39	PIN Translate and Generate MAC	PIN Translate and Generate MAC.	Security Exposure	4-163
3A	Card and PIN Verification	Customer Specific Command	Premium Value	n/a
3D	Generate PVN and Offset	Generates and Identikey PVN and IBM 3624 Offset from an encrypted PIN.	Premium Value	4-168
3F	PIN Verify	Customer Specific Command	Premium Value	n/a
55	Encrypt, Decrypt, or Translate data	Encrypts, Decrypts, or Translate Data using ECB mode of DES	Security Exposure	5-5
58	Translate MAC	Verifies a MAC and Generates a new MAC	Security Exposure	6-5
59	Generate MAC and Encrypt or Translate Data	Generate MAC and Encrypt or Translate Data	Security Exposure	6-13
5C	Verify and Generate MAC for VISA UKPT	Verifies a Message Authentication Code and generates an approval or denial Message Authentication Code.	Standard	6-25

Table C-1. Command Locator (page 6 of 16)

Command	Name	Purpose	Security Policy	Page
5D	Generate CVV/CVC	Generates a Card Verification Value/Card Validation Code	Security Exposure	7-3
5E	Verify CVV/CVC	Verifies Card Verification Value/Card Validation Code	Standard	7-6
5F	Verify MAC and Decrypt PIN	Verifies a MAC and if successful decrypts a PIN	Premium Value	6-29
70	Load Volatile Table Value	Loads a DES key or conversion table into the next available location in the table.	Standard	9-3
71	Delete Volatile Table Value	Deletes a value stored in a specific location.	Standard	9-6
72	Verify Volatile Table Value	Retrieves the check digits of a value stored in a specific location.	Standard	9-8
73	Clear Volatile Table	Clears the volatile table.	Standard	9-10
74	Load Diebold Number Table Row	Loads a row of the Diebold number table.	Standard	9-12
75	Enter Key Component	Encrypts a key under the MFK.	Premium Value	n/a
76	Import KEK	Customer Specific Command	Premium Value	n/a
77	Export KEK	Customer Specific Command	Premium Value	n/a
78	Import Operation Key	Customer Specific Command	Premium Value	n/a
79	Export Operation Key	Customer Specific Command	Premium Value	n/a
7A	Generate Check Digits	Customer Specific Command	Premium Value	n/a
7B	Verify Check Digits	Customer Specific Command	Premium Value	n/a
7E	Generate Check Digits	Generates check digits in order to confirm that two parties hold the same key value.	Standard	3-59
7F	Load Value to a Specific Location	Loads a DES key or conversion table into a specified location.	Standard	9-15

90	Decrypt PIN	Decrypts an incoming PIN block and returns the clear-text PIN.	Premium Value	4-172

Generars-24.4 a decimal or - ptMandarde
nomNnumber.

(and-112106(Nam8n)847-08(Purpose)TJ3-196720-1.1858 TD0006 TcSecuritym)Tj0 -1.1858 TD0094

Table C-1. Command Locator (page 8 of 16)

Command	Name	Purpose	Security Policy	Page
9F	Replace the Current MFK with the Pending MFK	Replaces the current Master File Key with the pending one.	Standard	3-66
B1	Generate PIN	Customer Specific Command	Premium Value	n/a
B2	Generate Token Response	Customer Specific Command	Premium Value	n/a
B3	Verify Token Response	Customer Specific Command	Premium Value	n/a
B4	Generate MD4 Hash	Customer Specific Command	Premium Value	n/a
B5	Verify Signature	Customer Specific Command	Premium Value	n/a
B6	One Way Encryption	Customer Specific Command	Premium Value	n/a
BA	PIN Translate ANSI to PIN Pad and MAC Verify	Translates an ANSI PIN Block to PIN Pad and Verifies a MAC.	Security Exposure	4-175
BB	PIN Translate ANSI to Plus an MAC Verify	Translates an ANSI PIN Block to Plus and Verifies a MAC.	Security Exposure	4-179
BD	PIN Translate and Generate MAC	Translates a PIN and Generates a MAC	Security Exposure	4-183
BE	Verify VSVC S1 Signature	Used to validate the S1 signature and generate the S2 signature for VSVC cards.	Premium Value	8-5
BF	Verify VSVC S3 Signature	Used to validate the S3 signature for VSVC cards.	Premium Value	8-10
D0	Verify Clear PIN	Verifies a clear-text PIN according to the specified verification method.	Premium Value	4-191
D1	Verify Password	Customer Specific Command	Premium Value	n/a
D2	Modify Password	Customer Specific Command	Premium Value	n/a
D3	Generate Initial Password Offset	Customer Specific Command	Premium Value	n/a
D4	Verify Password	Customer Specific Command	Premium Value	n/a
D5	Verify Signature	Customer Specific Command	Premium Value	n/a

Table C-1. Command Locator (page 9 of 16)

Command	Name	Purpose	Security Policy	Page
D6	Modify Signature	Customer Specific Command	Premium Value	n/a
D7	Initial Signature	Customer Specific Command	Premium Value	n/a
D8	Generate Hash	Customer Specific Command	Premium Value	n/a
D9	Verify Hash Signature	Customer Specific Command	Premium Value	n/a
DA	Verify MAC	Customer Specific Command	Premium Value	n/a
101*	Configure Security Processor Options	Lets you enable and disable the security processor's options.	Standard	11-29
102	Command Monitoring	Counts verification failures and commands.	Standard	11-32
105	Enable Premium Value Commands and Options	Enables Premium Value Commands and Options in a Network Security Processor	Standard	11-37
106	Define Temporary Serial Number	Allows the entry of a temporary serial number into a Network Security Processor.	Standard	11-41
107	Confirm Temporary Serial Number	Activates a temporary serial number into a Network Security Processor	Standard	11-44
108*	Define Security Policy	Lets you enable and disable the security processor's commands and security related options.	Standard	11-48
109*	Confirm Security Policy	Implements the security policy defined in command 108.	Standard	11-55
110	Generate KSM Key	CSM command	Security Exposure	n/a
111	Process KSM Key	CSM command	Security Exposure	n/a
112	Generate CSM MAC Key	CSM command	Security Exposure	n/a
113	Translate Key	Translates a key - ECB to CBC mode, or CBC to ECB mode.	Standard	3-69

Table C-1. Command Locator (page 10 of 16)

Command	Name	Purpose	Security Policy	Page
114	Import Key	Customer Specific Command	Premium Value	n/a
115	Generate Key	Customer Specific Command	Premium Value	n/a
11D	Generate ATM MAC or Data Encryption Key	Generates an ATM MAC or Data Encryption Key	Premium Value	3-72
11E	Generate Atalla 2x2 PVN	Generates a PIN Verification Number using the Atalla 2x2 method.	Premium Value	4-194
15E	Combine Key Components	Combine key component to form a key	Security Exposure	10-6
160	Generate PIN Printing Key	Generate a PIN Printing Key	Security Exposure	10-10
161	Print PIN Letter	Print a PIN Letter	Security Exposure	10-13
162	PIN Issuance: IBM 3624 Method	Generate a PIN and/or Offset	Security Exposure	10-19
163	PIN Issuance: Visa Method	Generate a PIN and PVV	Security Exposure	10-26
16E	Divide a key into components	Divide an existing encrypted key into multiple components	Security Exposure	10-31
16F	Print Component Letter	Print a key component letter	Security Exposure	10-36
301	Verify MAC	Customer Specific Command	Premium Value	n/a
302	Generate MAC	Customer Specific Command	Premium Value	n/a
306	Generate Cryptogram	Customer Specific Command	Premium Value	n/a
307	Generate APRC	Customer Specific Command	Premium Value	n/a
308	Generate MAC	Customer Specific Command	Premium Value	n/a
309	Verify MAC	Customer Specific Command	Premium Value	n/a
30A	Calculate PIN Offset	Generates a new Offset based on the old Offset	Premium Value	4-197

Table C-1. Command Locator (page 11 of 16)

Command	Name	Purpose	Security Policy	Page
30B	Verify MAC	Customer Specific Command	Premium Value	n/a
30C	Generate MAC	Customer Specific Command	Premium Value	n/a
30D	Translate MAC	Customer Specific Command	Premium Value	n/a
30E	Verify MAC	Customer Specific Command	Premium Value	n/a
30F	Generate MAC	Customer Specific Command	Premium Value	n/a
319	Generate Cryptogram	Customer Specific Command	Premium Value	n/a
31A	Verify Check Value	Customer Specific Command	Premium Value	n/a
31B	Decrypt Data	Customer Specific Command	Premium Value	n/a
31C	Encrypt Password	Customer Specific Command	Premium Value	n/a
31D	Verify Password	Customer Specific Command	Premium Value	n/a
31E	Generate Key	Customer Specific Command	Premium Value	n/a
31F	Verify Key	Customer Specific Command	Premium Value	n/a
321	Verify PIN	Customer Specific Command	Premium Value	n/a
328	Verify PIN	Customer Specific Command	Premium Value	n/a
32A	Verify PIN	Customer Specific Command	Premium Value	n/a
32B	Import Key	Customer Specific Command	Premium Value	n/a
32C	Verify ePIN Offset	Verifies an ePIN using an Offset	Premium Value	4-202
332	PIN Translate	Customer Specific Command	Premium Value	n/a
333	PIN Translate	Customer Specific Command	Premium Value	n/a

Table C-1. Command Locator (page 12 of 16)

Command	Name	Purpose	Security Policy	Page
334	PIN Translate	Customer Specific Command	Premium Value	n/a
335	PIN Translate	Translates a PIN into a variety of PIN block types.	Standard	4-205
336	PIN Translate	Customer Specific Command	Premium Value	n/a
337	PIN Translate	Customer Specific Command	Premium Value	n/a
338	Export PIN	Customer Specific Command	Premium Value	n/a
339	Generate PIN Offset	Customer Specific Command	Premium Value	n/a
33A	PIN Translate	Customer Specific Command	Premium Value	n/a
33B	Translate Response	Customer Specific Command	Premium Value	n/a
33C	Generate Key	Customer Specific Command	Premium Value	n/a
33D	Derive Key	Customer Specific Command	Premium Value	n/a
33E	Data Decrypt and PIN Translate	Customer Specific Command	Premium Value	n/a
33F	Data Encrypt and PIN Translate	Customer Specific Command	Premium Value	n/a
348	Verify DUKPT Message Authentication Code	Verifies a Message Authentication Code that was generated using a Derived Unique Key per Transaction Key.	Standard	6-51
349	Generate Terminal Master Key	Customer Specific Command	Premium Value	n/a
34A	Calculate PIN	Customer Specific Command	Premium Value	n/a
34B	Translate Key	Customer Specific Command	Premium Value	n/a
34C	Generate PIN	Customer Specific Command	Premium Value	n/a
34D	Derive Key	Customer Specific Command	Premium Value	n/a

Table C-1. Command Locator (page 13 of 16)

Command	Name	Purpose	Security Policy	Page
34E	Derive Terminal Master Key	Customer Specific Command	Premium Value	n/a
34F	Generate Terminal Personalization Key	Customer Specific Command	Premium Value	n/a
350	Verify ARQC and return ARPC	Verifies an Authorization Request Cryptogram, and returns an Authorization Response Cryptogram, using either the Visa or Europay/Mastercard algorithms.	Standard	8-14
351	EMV PIN Change	Facilitates the functions required when performing an EMV PIN Change without using the current PIN.	Premium Value	8-23
352	Generate EMV MAC	Generates a MAC using either the Visa or Europay/Mastercard algorithms.	Standard	8-29
354	Generate EMV ICC Master Key	Generates an EMV ICC Master Key	Standard	8-38
356	Validate CAP Token	Verifies an application cryptogram (AC) or signs transaction data.	Standard	8-42
357	Verify dCVV	Verifies a VISA dynamic Card Verification Value	Standard	7-9
359	Verify dynamic CVC3	Verifies a MasterCard dynamic Card Validation Code 3	Standard	7-12
35A	Verify AMEX CSC	Verifies an American Express Card Security Code	Standard	7-16
35B	Generate AMEX CSC	Generates an American Express Card Security Code	Security Exposure	7-20
35C	Cardholder Authentication Value	Customer Specific Command	Premium Value	n/a
35E	Derive Terminal Master Key	Customer Specific Command	Premium Value	n/a
35F	Verify DCVV	Verifies a Discover card Dynamic Card Verification Value	Standard	7-23
360	Generate Card Key	Customer Specific Command	Premium Value	n/a

Table C-1. Command Locator (page 14 of 16)

Command	Name	Purpose	Security Policy	Page
361	Generate Dynamic PAN	Customer Specific Command	Premium Value	n/a
362	Translate Dynamic PAN	Customer Specific Command	Premium Value	n/a
363	Generate DTC	Customer Specific Command	Premium Value	n/a
364	Verify DTC	Customer Specific Command	Premium Value	n/a
36A	Verify AMEX Expresspay - Magstripe	Verifies an AMEX Expresspay value using the Magstripe mode	Standard	7-26
370	Validate PIN	Customer Specific Command	Premium Value	n/a
371	Change PIN	Customer Specific Command	Premium Value	n/a
372	Translate Reference PIN Block	Customer Specific Command	Premium Value	n/a
37A	Change PIN	Customer Specific Command	Premium Value	n/a
37B	Generate ePIN Offset	Generates an ePIN Offset	Premium Value	4-212
381	Verify MAC	Customer Specific Command	Premium Value	n/a
382	Generate MAC	Customer Specific Command	Premium Value	n/a
386	Generate DUKPT Message Authentication	Generates a Message Authentication Code using a Derived Unique Key per Transaction Key.	Security Exposure	6-55
388	3DES DUKPT Encrypt/Decrypt Data	Encrypts or Decrypts data	Security Exposure	5-26
390	Encrypt/Decrypt Data	Customer Specific Command	Premium Value	n/a
391	Generate Terminal Key	Customer Specific Command	Premium Value	n/a
392	Generate Check Digits	Customer Specific Command	Premium Value	n/a
3A1	PIN Verify	Customer Specific Command	Premium Value	n/a

Table C-1. Command Locator (page 15 of 16)

Command	Name	Purpose	Security Policy	Page
3A2	Generate PRV	Customer Specific Command	Premium Value	n/a
3A3	Verify PRV	Customer Specific Command	Premium Value	n/a
3A4	PIN Translate	Customer Specific Command	Premium Value	n/a
3B2	PIN Translate	Customer Specific Command	Premium Value	n/a
3B3	PIN Translate	Customer Specific Command	Premium Value	n/a
3B4	PIN Translate	Customer Specific Command	Premium Value	n/a
3B5	PIN Translate	Customer Specific Command	Premium Value	n/a
3EA	Derive Encrypted PIN	Customer Specific Command	Premium Value	n/a
3FA	Generate PIN and PVV	Customer Specific Command	Premium Value	n/a
1101*	Get Image ID	Returns the image version information of the cryptographic command processor.	Standard	11-58
1102*	Get Virtual NSP Information	Returns the number of the virtual NSP that the host application is connected to, the name of the virtual NSP, and number of virtual NSPs defined.	Standard	11-60
1104*	Get Temporary Serial Number Information	Returns the temporary serial number and remaining hours.	Standard	11-62
1105*	License Premium Commands/Options	Licenses premium value commands/option in all virtual NSPs.	Standard	11-64
1110*	Get System Configuration Information	Returns the version information of all components in the Network Security Processor.	Standard	11-67
1111*	Get System Date and Time	Returns the Network Security Processor's system date and time.	Standard	11-69

Table C-1. Command Locator (page 16 of 16)

Command	Name	Purpose	Security Policy	Page
1113*	Get CPU Utilization	Returns average CPU utilization.	Standard	11-71
1120*	Get System Information	Returns the NSP serial number, product ID, system software information, and a personality version field.	Standard	11-73
1204*	Get Log Signing Key Certificate	Returns the certificate which contains the public key to verify the log signature	Standard	11-75
1216*	Get Battery Life Remaining	Returns the number of days remaining before the battery expiration messages start appearing in the log.	Standard	11-78
1221*	Return IP Address of NSP	Returns the IP Address of the Network Security Processor.	Standard	11-80
1223*	TCP/IP Socket Information	Returns information on the number of TCP/IP sockets available on the Network Security Processor.	Standard	11-82
1226*	Get Check Digits	Returns check digits of keys in the non-volatile key table.	Standard	11-85
1227*	Reset to Factory State Part 1	Used to reset the Network Security Processor to factory state.	Standard	11-87
1228*	Reset to Factory State Part 2	Used to reset the Network Security Processor to factory state.	Standard	11-89
1350*	Select Virtual NSP	Choose which virtual NSP should process subsequent commands	Standard	11-91
1351*	Virtual NSP System Information	Returns the name and MFK check digits of all virtual NSPs.	Standard	11-93

* This command is not controlled by the security policy it is always enabled.

Network Security Processor Options

Table C-2. Network Security Processor Options (page 1 of 7)

Option	Name	Purpose	Security Policy
20	Append MFK Name to all responses	Append the Master File Key name to all responses except the response of the status command, 9A; default – do not append name. Enable this option using either the SCA-3 or command 101.	Standard Default is Off
21	Append Detailed Error information	Append the detailed error information to the error response, 00; default – do not append detailed error. Enable this option using either the SCA-3 or command 101.	Standard Default is Off
23	Remove CR/LF from responses	Remove the carriage return and line feed from all responses; default – CR/LF appended to all responses. Enable this option using either the SCA-3 or command 101.	Standard Default is Off
27	Use the rightmost 4 PIN digits for Diebold PIN verification. (Default uses leftmost 4 PIN digits)	Use the rightmost 4 PIN digits for Diebold PIN verification; default is to use the leftmost 4 PIN digits. Enable this option using either the SCA-3 or command 101.	Standard Default is Off
44	Record Network Security Processor Command and Error Response to System Log	When this option is enabled, any host application command sent to the Network Security Processor that results in an error will be logged to the system log along with the corresponding error response. This option is useful for troubleshooting host applications. Be sure to disable this option after the problem has been identified. Enable this option using either the SCA-3 or command 101.	Security Exposure Default is Off

Table C-2. Network Security Processor Options (page 2 of 7)

Option	Name	Purpose	Security Policy
46	Restrict PIN block types in PIN translate commands	See Option 46 - ANSI and ISO-3 PIN block	Security Exposure Default is Off
47	Restrict outgoing PIN block types in PIN translate commands	See Option 47 - ANSI and ISO-3 Outgoing PIN block	Security Exposure Default is Off
48	Require encrypted conversion tables	See Option 48 - Encrypted Conversion Tables	Security Exposure Default is Off
49	Outgoing PIN Encryption Key (KPEo) length check	See Option 49 - Outgoing PIN Encryption Key length	Security Exposure Default is Off
4B	Prevent PIN block attack	See Option 4B - Modified PIN Sanity Test	Security Exposure Default is Off
4C	Validation digits match ANSI PAN digits	See Option 4C - Validation Data equals ANSI PIN block data	Security Exposure Default is Off
4D	CVV/CVCs length check	See Option 4D - CVV/CVC length	Security Exposure Default is Off
4E	Conversion Table Restriction	See Option 4E - Conversion Table restrictions	Security Exposure Default is Off
4F	Check Digit Methods in command 7E.	If this option is disabled method R is disabled and method I is enabled. If this option is enabled method R is enabled and method I is disabled.	Security Exposure Default is Off
60	Clear PIN Compare	Controls the ability to verify clear PINs. See Verify PIN – Clear-PIN Comparison (Command 32) on page 4-67. If this option is enabled, the clear PIN Compare command will be enabled.	Premium Value

Table C-2. Network Security Processor Options (page 3 of 7)

Option	Name	Purpose	Security Policy
61	Encrypted PIN Compare	Controls the ability to verify encrypted PINs by comparison. See Verify PIN – PIN-Block Comparison (Command 32) on page 4-70. If this option is enabled, the encrypted PIN Compare command will be enabled.	Premium Value
62	Allow command 31 DUKPT	Controls the ability to translate PINs that have been encrypted using the Visa DUKPT PIN encryption key. See Translate PIN – VISA DUKPT (Command 31) on page 4-30. If this option is enabled, the PIN translate command 31 will allow an incoming PIN block to be in the Visa DUPKPT format.	Standard Default is On
63	Allow command 32 and 36 DUKPT	Controls the ability to verify PINs that have been encrypted using the Visa DUKPT PIN encryption key. See Verify PIN – Atalla 2x2 (Command 32) on page 4-78 and Verify Double-Encrypted PIN (Command 36) on page 4-126 If this option is enabled, the PIN Verify commands 32 and 36 will allow an incoming PIN block to be in the Visa DUPKPT format.	Standard Default is On
65	Allow Commands 13 and 19 to use variant 0	This option is required If importing a KEK.	Premium Value
66	Do not validate old PIN for command 37	Controls the ability to validate the old PIN in command 37. If this option is enabled, command 37 will not check the old PIN before processing the new PIN.	Premium Value

Table C-2. Network Security Processor Options (page 4 of 7)

Option	Name	Purpose	Security Policy
68	Allow Command 32 option E	Controls the ability to verify EBCDIC PINs. This option must be purchased in the form of a command 105 and enabled in the Network Security Processor's security policy. If this option is enabled, command 32#E will be processed.	Premium Value
69	Allow Command 37A to not verify the old PIN	This is an option for a custom command.	Premium Value
6A	Allows both halves of a 2key-3DES key to have the same value	Allow both halves of a 2key-3DES (double-length) key to be the same value supported in commands: 30. 97, 98, 99, 113, 30A, 31E, 32C, 335, 344, 350, 352, 354, 356, 357, 359, 35A, 35B, 370, 371, 37B. If this option is enabled, both halves of a 2key-3DES (double-length) key can be the same value.	Security Exposure Default is Off
6B	Requires the Incoming PAN/ICV to match the outgoing PAN/ICV and be non-zero	Requires the Incoming PAN/ICV to match the outgoing PAN/ICV and be non-zero. This option is only used in PIN and PIN-Block Translate (Command 335) on page 4-205. If this option is enabled, command 335 will verify that the incoming and outgoing PAN/ICV match and are non-zero.	Security Exposure Default is Off
6C	Allows commands to accept single-length incoming keys	Allows certain commands to accept a 1key-3DES (single-length) working keys. If this option is enabled, some commands will allow 1key-3DES (single-length) working keys.	Security Exposure Default is Off

Table C-2. Network Security Processor Options (page 5 of 7)

Option	Name	Purpose	Security Policy
6E	Disable sequence number validation for command 109	Controls the ability to validate the sequence number when accepting a new security policy. If this option is enabled, the sequence number will not be validated when accepting a new security policy.	Security Exposure Default is Off
6F	Disable serial number validation for command 109	Controls the ability to validate the serial number when accepting a new security policy. If this option is enabled the serial number will not be validated when accepting a new security policy.	Security Exposure Default is Off
80	Command 3F, option 1-6	This is an option for a custom command.	Premium Value
81	Command 3F, option 7	This is an option for a custom command.	Premium Value
82	Command 3F, option 8 and 9	This is an option for a custom command.	Premium Value
83	Command 3F, option SG, SA, and SC	This is an option for a custom command.	Premium Value
84	Command 32A, option B	This is an option for a custom command.	Premium Value
87	Enable NIC2	When this option is enabled, the NSP will enable NIC2 per the keyword/value pairs present in the config.prm file.	Default is Off.
88	Return 6 check digits	If this option is enabled the Network Security Processor will return six check digits instead of four for certain commands.	Security Exposure Default is Off

Table C-2. Network Security Processor Options (page 6 of 7)

Option	Name	Purpose	Security Policy
89	Controls Command 32#E	This is an option for a custom command. Use this option along with option 68 to control the operation of command 32#E as follows: If options 68 and 89 are OFF, command 32#E will not be accepted. If options 68 and 89 are ON, the EBCDIC form of command 32#E will be accepted. If option 68 is ON and option 89 is OFF, the EBCDIC form of command 32#E will be accepted. If option 68 is OFF and option 89 is ON, the EssoPAC form of command 32#E will be accepted.	Premium Value
8A	Enable commands 32#G and 32#H	This is an option for custom commands.	Premium Value
8B	Enable command 14#5#	This option is used to enable command 14#5#.	Premium Value
8D	Allow variant 0 in 14#5#	This option is used in command 14#5#.	Premium Value
A0	Minimum PIN length	The minimum PIN length can be from 0 to 12. Minimum PIN lengths of 10, 11, or 12 are defined as A, B, and C, respectively. This option replaces option 40 that was set using command 101.	Default is 4

Table C-2. Network Security Processor Options (page 7 of 7)

Option	Name	Purpose	Security Policy
A1	Defines the Sanity indicator	Controls the value of the sanity indicator. The default value is “S”; returns a sanity error if the PIN length is out of range or the decrypted PIN block is invalid. When this option is set to “L”, the Network Security Processor will perform a PIN length test before the PIN sanity test. When a PIN length error is detected, the sanity indicator, returned in the response, will be the letter “L”. This option replaces option 24 that was set using command 101.	Default is S
A2	3DES DUKPT session key length	This option controls the length of the generated session key in commands 1E, 31, 32, and 5C, and also in custom commands 308 and 309. The default value is “S” 1key-3DES (single-length). For 2key-3DES (double-length), set this option to “D”. To allow the host application to specify the length of the session key, set this option to “B”. The length of the Base Derivation Key must be greater than or equal to the length of the session key.	Default is S
C1	SCA-3 screen control	This option when enabled, instructs the SCA-3 to display specific screens.	Premium Value

Recommended settings for security options

These options give security officers the ability to restrict the input data supplied in PIN processing commands, as well as in the Verify Visa Card Verification Value or MasterCard Card Validation Code command. The default value may not be the most secure choice. Carefully review each of these options and then decide which ones should be enabled in the Network Security Processor's security policy.

Option 46 - ANSI and ISO-3 PIN block

This option is used to restrict PIN block types in the following PIN translate commands: 31, 33#11, 33#13, 33#19, 33#22, 33#23, 33#33, 33#39, 33#91, 33#93, 33#99, 35, 39, BA, BB, BD and 335.

The default setting for this option is disabled (OFF), which means that PIN translate commands will allow all PIN block types supported by that command.

When this option is enabled (ON), the Network Security Processor enforces these two requirements:

- Only ANSI (also referred to as ISO-0) or ISO-3 PIN blocks are allowed in the PIN translate command. This requirement is enforced for both the incoming and outgoing PIN block.
- In PIN translate commands 33#11, 35#...#...#11#, BB and 335 which contain both an incoming and an outgoing ANSI or ISO-3 Primary Account Number (PAN) field, both the incoming and outgoing ANSI or ISO-3 PAN values must be identical.

Recommendation: Review your PIN processing environment to determine what PIN translate commands are in use; disable all unnecessary PIN translate commands. Check with your processing partners to determine what types of PIN blocks should be allowed, and if the incoming ANSI or ISO-3 PAN data should be different than the outgoing ANSI or ISO-3 PAN data. Enable this option if only ANSI or ISO-3 PIN blocks should be allowed, and there is no legitimate business reason to support different values for the incoming and outgoing ANSI or ISO-3 PAN data.

Option 47 - ANSI and ISO-3 Outgoing PIN block

This option is used to restrict the types of outgoing PIN blocks in the following PIN translate command: 33#11, 33#13, 33#19, 33#22, 33#23, 33#33, 33#39, 33#91, 33#93, 33#99, 35, BA, BB and 335.

The default setting for this option is disabled (OFF), which means that PIN translate commands will allow all outgoing PIN block types supported by that command.

When this option is enabled (ON), the Network Security Processor enforces these two requirements:

- Only ANSI (also referred to as ISO-0) or ISO-3 outgoing PIN blocks are allowed in PIN translate commands. All incoming PIN block types supported by the PIN translate command are allowed.

- In PIN translate commands 33#11, 35#...#...#11#, BB and 335, which contain both an incoming and an outgoing ANSI or ISO-3 Primary Account Number (PAN) field, both the incoming and outgoing ANSI or ISO-3 PAN values must be identical.

Note. When option 46 is enabled, it supersedes this option.

Recommendation: Review your PIN processing environment to determine what PIN translate commands are in use; disable all unnecessary PIN translate commands. Check with your processing partners to determine what types of PIN blocks should be allowed, and if the incoming ANSI or ISO-3 PAN data should be different than the outgoing ANSI or ISO-3 PAN data. Enable this option if only ANSI or ISO-3 outgoing PIN blocks should be allowed, and there is no legitimate business reason to support different values for the incoming and outgoing ANSI or ISO-3 PAN data.

Option 48 - Encrypted Conversion Tables

This option affects PIN verification and PIN change commands that support conversion tables. The commands affected by this option are: 32#2, 32#6, 36, 37#2, 37#6, 3D, D0#2 and 30A.

The default setting for this option is disabled (OFF), which means that only clear-text conversion tables or volatile table locations that contain encrypted conversion tables are supported in PIN verification and PIN change commands.

When this option is enabled (ON), the Network Security Processor enforces either of these two requirements:

- Conversion tables must be supplied encrypted under variant 6 of the MFK.
- The volatile table location that contains the conversion table must be provided in the command. The conversion table must be loaded into the volatile table using command 70 or 7F.

Recommendation: Review your PIN processing environment to determine if conversion tables are in use. Confirm that your host application can support encrypted conversion tables, and if so, enable this option. If the host application cannot support encrypted conversion tables consider enabling option 4E.

Option 49 - Outgoing PIN Encryption Key length

This option is used to restrict the length of the outgoing PIN encryption key in the following PIN translate commands: 31, 33#11, 33#13, 33#19, 33#22, 33#23, 33#33, 33#39, 33#91,33#93, 33#99, 35, 39, BA, BB, BD and 335.

The default setting for this option is disabled (OFF), which means that the length of the outgoing PIN Encryption Key (KPEo) is only restricted by the option 6C.

When this option is enabled (ON), the length of the KPEo must be equal to, or greater than, the length of the incoming PIN Encryption Key (KPEi). This option does not restrict the length of the KPEi.

Recommendation: Review your PIN processing environment to determine what PIN translate commands are in use; disable all unnecessary PIN translate commands. Enable this option if all your processing partners require PIN blocks encrypted under 2key-3DES (double-length) keys.

Option 4B - Modified PIN Sanity Test

This option affects PIN change, PIN translate, and PIN verification commands that support ANSI (also referred to as ISO-0) and ISO-3 PIN blocks. The commands affected by this option are: 31, 32#1, 32#2, 32#3, 32#4, 32#5, 32#6, 32#7, 32#F, 32#I, 33#11,33#13, 33#19, 35, 36, 37#1, 37#2, 37#3, 37#4, 37#5, 37#6, 39, 3D, BA, BB, BD and 335.

The default setting for this option is disabled (OFF), which means that all 16 hexadecimal characters of the decrypted ANSI or ISO-3 PIN block will be evaluated for sanity. And in the case of a PIN translate command, if the decrypted PIN block fails the sanity test, the value of the decrypted PIN block will be encrypted under the outgoing PIN Encryption Key (KPEo) and returned in the response.

When this option is enabled (ON), a modified PIN sanity test, which checks a subset of the decrypted PIN block, is performed. And in the case of a PIN translate command, if the decrypted PIN block fails the modified sanity test, 16 zeros will be returned in the response. This modified PIN sanity test block does not reveal information about the PIN when the Primary Account Number (PAN) digits, used to form the ANSI or ISO-3 PIN block, are manipulated.

Since all of the decrypted PIN block digits are not checked for sanity, there is the potential that in certain rare key synchronization conditions a PIN will not verify, or in the case of a PIN translate command an incorrect encrypted PIN will be returned in the response.

Recommendation: Enable this option so no useful information about a PIN is returned when incorrect primary account numbers are sent to the Network Security Processor in a PIN change, translate, and PIN verification command.

Option 4C - Validation Data equals ANSI PIN block data

This option affects PIN change and PIN verification commands that support ANSI (also referred to as ISO-0) PIN blocks. In IBM3624, NCR, and Visa PIN change and PIN verification commands, customer specific account digits are used in the PIN validation process. These same digits may, or may not, be used to form the ANSI PIN block. The commands affected by this option are: 32#2, 32#3, 32#6, 36, 37#2, 37#3, 37#6, and 3D.

The default setting for this option is disabled (OFF), which means that no comparison of the PIN validation data and ANSI PAN digits is performed.

When this option is enabled (ON), the IBM3624 and NCR validation digits must match the 12 digits used to form the ANSI PIN block. If there are less than 12 validation data digits an error response will be returned. When there are more than 12 validation

digits, the Network Security Processor will compare the rightmost 12 digits of the validation data to the ANSI PAN digits. If these digits do not match, the Network Security Processor will discard the rightmost digit of the validation data and the comparison is performed again. An error is returned If both comparisons fail. In a Visa PIN change or PIN verification command, the 11 digits of the verification data are compared to the rightmost 11 ANSI PAN digits. If that test fails, the 10 rightmost ANSI PAN digits are compared to the 10 leftmost digits of the VISA verification data. An error is returned if both comparisons fail.

Recommendation: Review your PIN processing environment to determine what validation data lengths are supported. Enable this option if the validation data is 12 (11 digits for VISA) or more digits, and the validation data is equal to the digits used to form the ANSI PIN block.

Option 4D - CVV/CVC length

This option affects Card Verification Value and Card Validation Code (CVV/CVC) verification command 5E.

The default value for this option is OFF, which means that the CVV/CVC to be verified can be 1 - 8 digits in length.

When this option is enabled (ON), the CVV/CVC to be verified must be 3 - 8 digits in length.

Note. In command 5E, this option applies only when field one (algorithm identifier) is set to 3.

Recommendation: Review your CVV/CVC processing environment to determine the lengths of CVV/CVCs to be verified. Enable this option if all CVV/CVCs to be verified are at least three digits in length.

Option 4E - Conversion Table restrictions

This option affects PIN change and PIN verification commands that support either the IBM3624 or NCR PIN algorithms. The commands affected by this option are: 32#2, 32#6, 36, 37#2, 37#6, 3D, D0#2 and 30A.

The default setting for this option is disabled (OFF), which means that any clear-text conversion table is allowed.

When this option is enabled (ON), two restrictions are placed on the conversion table:

- The numeric conversion table must contain at least eight unique digits.
- No single digit can occur more than four times.

Example conversion tables that adhere to these restrictions:

0123456789012345, 987654321054321, 8351296477461538

Example conversion tables that do not adhere to these restrictions:

- 1234567123456712, does not contain eight unique digits

- 2437528797671271, the number 7 appears more than 4 times.

Note. When both options 48 and 4E are enabled, the conversion table restrictions are applied to the decrypted conversion table.

Recommendation: Review your PIN processing environment to determine what, if any, conversion tables are used. Enable this option if all conversion tables meet the requirements mentioned above.

D Contacting Atalla

Before contacting Atalla Technical Support, please read this manual. Many of the common installation, key loading, and product questions are covered in detail in this guide. If you are still unable to find answers to your questions, contact Atalla Technical Support. Atalla Technical Support's normal working hours are 8 am to 5 pm, Pacific Standard Time, Monday through Friday. Atalla Technical Support provides assistance for customers and field personnel who have questions or problems with the installation, setup and use of Atalla equipment or products. When requesting support, please have the following information available as it will enable us to quickly and efficiently answer your question or solve the problem you are encountering:

- Type and model of Atalla equipment.
- Type of system it is attached to or installed in.
- Exact nature of the problem, provide as much detail as possible.

Customers can contact Atalla Technical Support by:

By e-mail:

`atalla.support@hp.com`

By telephone:

`800-500-7858` (U.S. only)

or

`916-414-0216` (outside U.S.)

24-hour Support

Atalla Technical Support provides 24-hour emergency coverage for those customers who have valid service contracts. Use this service for Atalla product and system emergencies that occur after normal working hours or on weekends and U.S. holidays. Questions about Atalla product installation and setup are supported during normal working hours.

NonStop Service Contracts

Customers with NonStop service contracts can reach Atalla Technical Support through the Global Customer Support Center (GCSC). Within the U.S. the GCSC can be contacted by calling:

`800-255-5010`

Customers located outside the U.S. can obtain local GCSC contact information from the **Country phone numbers** section of this document:

`http://h20195.www2.hp.com/V2/GetPDF.aspx/c02083951.pdf`

HP CarePack Service Contracts

Customers with HP CarePack contracts can reach Atalla Technical Support by contacting HP Technical Support. To efficiently route your call to the appropriate support organization use any of these key words: HP Atalla, Network Security Processor, or NSP.

Customers will be asked to provide their 12 digit contract Service Agreement Identification Number (SAID) and the product serial number.

Within the U.S call :

```
800-633-3600
```

Customers located outside the U.S. should contact their local HP Technical Support organization or call:

```
+1 770-343-5002
```

On-site Support

On-site assistance for Atalla equipment, products, and training is provided for a fee. For more information on Atalla Professional Services, contact the Atalla Sales Department:

By e-mail:

```
AtallaOrders@hp.COM
```

By telephone:

```
800-523-9981 (U.S. only)
```

or

```
916-414-0217 (outside U.S.)
```

Glossary

AATMKEY. This working key is used to encrypt the ATM B key before the B key's cryptogram is loaded into an ATM machine.

Acquirer Node. The computer that has attached to it, automatic teller machines or PIN pads that introduce transactions into the network.

ANSI. American National Standards Institute.

ATM. Automated Teller Machine.

Authenticate. To establish the validity of a claimed identify.

BATM. ATM B-Key. This working key is used to encrypt the ATM master key before it is transmitted to an ATM machine.

CTK. Configuration Table Key. A single-length key, stored in the non-volatile key table. It is used to process a special configuration command 100.

Check Digit. An ending digit that is derived from the preceding digits in a number using an algorithm. Usually appended to the Primary Account Number (PAN).

Check Digits. A four to six hexadecimal character value used to ensure both entities have the same secret value without knowing the actual value.

Clear-Text. Data or a key value in unencrypted form.

CMDID. The two, three, or four-character Command ID.

CRLF. Carriage Return Line Feed. Added to the end of the response. It can be removed by enabling option 23 in the CONFIG_COMMANDS section of the CONFIG.PRM file.

CONFIG.PRM. The file used to configure the NSP. It resides on the System Image CD-ROM and must be copied to the USB flash memory device.

CVC. Card Validation Code. Check values that confirm the validity of a MasterCard bankcard's magnetic stripe.

CVV. Card Verification Value. Check value that confirms the validity of a VISA bankcard's magnetic stripe.

Decryption. The process of using a key to unscramble data.

DES. Data Encryption Standard. A cryptographic algorithm which employs a 56-bit secret key, adopted by the National Bureau of Standards for data security.

DK. Derivation Key. A working key which is used in a cryptographic process to derive other keys.

DNT. Diebold Number Table.

Double-Length Key. A DES Key that contains a 128 bits, consisting of a left half and right half.

DUKPT. Derived Unique Key Per Transaction. A key management scheme developed by Visa, used in Point-Of-Sale devices. As the name implies the key used to encrypt is derived by the host security module based on data sent from the device.

Encryption. The process of using a data encryption key to scramble data so that it cannot be read by someone who does not have the key.

Exclusive Or. A process of combining two values on a bit-by-bit basis. If both values contain a one bit, the resulting bit will be zero. If only one of the values contain a one bit the resulting bit will be one. If neither of the values contain a one bit the resulting value will be zero.

Hexadecimal. The character set of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. When representing a DES key or encrypted DES key (cryptogram), each character contains 4 bits. Therefore a single-length DES key with contains 64 bits is represented as 16 hexadecimal characters.

IKSN. Initial Key Serial Number. A non-secret value loaded by the acquirer into each PIN Pad to generate the initial PIN encryption key. Each PIN Pad has a unique initial key serial number.

IPEK. Initial PIN Encryption Key. The result of encrypting the IKSN with the Derivation Key.

Issuer Node. The computer that belongs to the financial institution that has an account relationship with the consumer. An issuer can have ATMs or PIN Pads attached to it, enabling it to act as both an issuer and an acquirer.

IV. Initialization Vector. A value that is Exclusive OR'd with data.

IVN. Input Verification Number.

KC. Communications Key. Used in ATMs to encrypt information, such as a PIN.

KD. Data Encryption Key. Used to encrypt or decrypt transaction data.

KEK. Key Exchange Key. A cryptographic key used to encrypt working keys. It can be either single or double-length. For most commands the KEK is provided encrypted under variant zero of the MFK.

Key Table. An area of RAM memory used to hold up to 9999 single-length working keys. The contents of this key table are not maintained during a power outage.

KI. Initial Master Key.

KM. ATM Master Key.

KMAC. Message Authentication Code Key. Used to generate or verify the integrity of transmitted data.

KMATM. ATM Master Key. This key is downloaded to an ATM machine from the host computer during initialization to facilitate PIN encryption.

KPE. PIN Encryption Key. Used to encrypt or decrypt PINs.

KPEn. Unique Transaction Key. The key that encrypts the PIN from all but the first transaction.

KPV. PIN Verification Key. Used in an algorithm to verify PINs.

KSN. Key Serial Number. A non-secret value generated from the initial key serial number and an encryption counter, used in the Visa DUKPT key management scheme.

KX. Exchange Key. Another term for Key Exchange Key (KEK),

MAC. See Message Authentication Code.

Master File Key (MFK). The double-length cryptographic key under which all working keys are protected. It is stored in the Network Security Processors non-volatile key table. It is not erased if power is removed.

Message Authentication Code. A code derived from applying the DES algorithm and cryptographic key to a message to protect it from alteration.

MFK. Master File Key.

MFK Check Digits. The Master File Key's check digits. Are produced by encrypting zeros with the MFK. The check digits are the leftmost four characters of the result.

Non-volatile Key Table. An area of battery backed up memory used to hold the Master File Key and the Pending Master File Key. The contents of this non-volatile key table are maintained during a power outage.

NSP. Network Security Processor. A hardware security module used to perform cryptographic operations.

PAN. Primary Account Number.

Pending Master File Key. A double-length key that is stored in the non-volatile key table. It is promoted to the current MFK using command 9F.

PIN. Personal Identification Number.

PMK. PIN Master Key

POS. Point Of Sale.

PVN. PIN Verification Number. The result of processing a PIN through the Identikey algorithm.

PVV. PIN Verification Value.The result of processing a PIN through the Visa algorithm.

Replicated Single-Length Key. IA double-length key where both the left (key1) and right (key2) halves contain the same value.

Sanity Indicator. A flag returned in a response to indicate where or not the decrypted PIN block is in a valid format. When processing an encrypted PIN, the encrypted PIN block is decrypted inside the Network Security Processor. If the Network Security Processor determines that this decrypted PIN block is invalid it will set the sanity indicator to N. If the Network Security Processor determines that the decrypted PIN block is valid it will set the sanity indicator to Y.

Security Policy. The definition of commands and options that are enabled in the Network Security Processor.

Security Processor. See Network Security Processor (NSP).

Single-Length Key. IA DES key that contains 64 bits.

Switch Node. The computer that directs transactions from multiple acquirers to the appropriate issuer. A switch can have ATMs or PIN Pads attached to it.

TMK. Terminal Master Key.

Variant. A value, based on the type of key that is being encrypted, that is X-OR'd with the encrypting key.

Working Keys. A category of keys used to perform specific cryptographic operations. Every working key is encrypted by a unique version of the MFK or a KEK. They are not stored in the non-volatile key table. However they may be stored in the volatile table.

XOR'd. See Exclusive Or.

ZCMK. Zone Control Master Key. A Visa term for a Key Exchange Key.

Index

A

Application errors 12-1
  detailed 12-2
ATM Networks 4-2
Authentication 6-2
  all at once 6-2
  in batches 6-2

C

Card Security Codes 7-1
Card Validation Codes 7-1
Card Verification Values 7-1
CBC A-1
Cipher Block Chaining
  see CBC
Commands C-1
Command, Response
  introduction 1-1
CRLF 1-2
CVC
  See Card Validation Code
CVV
  See Card Verification Values
CVV/CVC 7-1
CVV/CVC Commands 7-2

D

Data Authentication 6-1
Data Authentication Commands 6-3
Data Authentication Tasks 6-1
  generating the MAC 6-1
  verifying the MAC 6-1
Data Encryption Algorithm
  see DEA
Data Encryption Standard
  see DES
Data Processing Commands 5-3
Data Processing Tasks 5-1
  decrypting 5-1
  encrypting data 5-1
  establishing a data encryption key 5-1
  part of message to be encrypted 5-1
  transmitting data 5-1
DEA A-1
Derivation Key B-4
DES
  description A-1
  key components A-5
  weak keys A-7
Diebold
  number table 9-12

E

ECB A-1
Electronic Code Book
  see ECB
Encrypting and Decrypting Data 5-1
Even Parity Keys A-6

F

Financial Interchange Networks B-1
  acquirer node B-1
  issuer node B-1
  switch node B-1

I

IKSN B-4
Initialization Checklist B-3
Initialization Vector 5-2

K

Key Variants 2-2
Keys

components A-5

L

List of Commands C-1
List of Options C-18

M

MAC Type 6-35, 6-41
MasterCard CVCs 7-1
Message Authentication Code 6-34, 6-40

N

National Institute of Standards Technology
see NIST
Network Initialization B-2
Network Initialization Commands 3-1
NIST A-1

O

Operating Overview 1-1
Options C-18

P

PIN Block Formats 4-4
ANSI 4-4
Burroughs 4-4
Diebold 4-4
Docutel 4-4
IBM 3624 4-4
IBM 4731 4-4
IBM Encrypting PIN Pad 4-4
VISA 4-4
PIN Blocks 4-2
translating 4-2
verifying 4-2
PIN Encryption 4-2
PIN Encryption Key B-3
PIN Pad 4-9
character format 4-9
PIN Processing 4-1
tasks 4-1
PIN Processing Commands 4-21
PMFK 2-1
Printing Commands 10-1
Programming Guidelines 1-5
closing socket 1-6
connecting socket 1-6
opening socket interface 1-6
receiving response 1-6
sending command 1-6
setting up application 1-5

T

Technical support
contacting D-1
Typographic Conventions -xxiv

U

Utility Commands 11-1

V

Verify and Generate MAC for VISA UKPT (Command 5C) 6-25
VISA CVV 7-1
defined 7-1
VISA DUKPT B-3
VISA DUKPT Networks 4-2
VISA UKPT 6-3
Volatile Table 9-1
commands 9-2
defined 9-1
deleting entries 9-2
loading 9-1
tasks 9-1
verifying entries 9-2
VSVC
data elements 8-3

defined 8-1
DES key management 8-2
signatures defined 8-1

W

Weak Keys A-7
Working Keys 2-1, B-4
Generate 3-4
Generate Visa 3-44
Translate for distribution 3-7
Translate for distribution to non-Atalla node 3-49
Translate for local storage 3-10