HP-UX Kernel Cryptographic Module 1.0 User Guide

Abstract

This document describes how to install, configure, and troubleshoot HPUX-KCM on HP-UX 11i v3 platforms. It is intended for system and network administrators who have knowledge of operating system concepts, commands, and configuration.

HP Part Number: 5900-3288
Published: October 2013
Edition: 1

Contents

1 Overview

The HP-UX Kernel Cryptographic Module (HP-UX KCM) is a common cryptographic library in HP-UX Kernel. It is a library of core cryptographic algorithms, which are used by HP-UX Kernel products.

HP-UX KCM implements FIPS 140-2 compliant algorithms for commonly used cryptographic operations such as data encryption/decryption, sign/verify, digest, HMAC, and random number generation.

HP-UX KCM is available in HP-UX Kernel as a dynamically loadable library with well-defined interfaces to invoke the crypto functions. This helps to bring modularity and standardization in the usage of crypto algorithms across the HP-UX Kernel products. HP-UX KCM is available on HP Integrity platform running HP-UX 11iv3.

HP-UX KCM is undergoing FIPS 140-2 Level 1 validation and is currently in NIST *Review Pending* state.

The interfaces supported by the library follows *RSA Security Inc. PKCS#11 V.2.20* specification.

For more information on PKCS, see PKCS #11 v2.20: Cryptographic Token Interface Standard document.

NOTE: This link will take you outside the Hewlett-Packard (HP) Web site. HP does not control and is not responsible for information outside of HP.com.

Supported configuration

The supported configuration for HPUX-KCM is HP-UX 11i v3 for HP Integrity Servers.

Features provided in this release

This section discusses the new features available in the HP-UX KCM version 1.0.

The table below lists the FIPS 140-2 compliant algorithms, key lengths, modes, and operations implemented by HP-UX KCM 1.0.

FIPS algo	Key size	Operations	Purpose
AES	128, 192, and 256 Mode: CBC	Generate, Encrypt, and Decrypt	Symmetric key operations (FIPS-197 compliant)
RSA	2048	Generate key pair, Sign, Verify, Wrap key, and Unwrap key	Asymmetric key operations (FIPS 186-3 and PKCS#1 v1.5 compliant)
SHA-2	256, 384, and 512	Digest	Digest operations (FIPS 180-3 compliant)
HMAC-SHA2	256, 384, and 512	Digest (with key)	Key-Hash Message Authentication Code (HMAC)
RNG		Generate random	NIST SP800-90A compliant DRBG

HP-UX KCM also implements the following algorithms, which are required for supportability purposes even though they are not FIPS 140-2 compliant.

Non FIPS algo	Key size	Operations	Purpose
AES	128, 192, and 256 Mode: CFB	Generate, Encrypt, and Decrypt	Symmetric key operations

RSA	1024 and 1536	Generate key pair, Sign, Verify, Wrap key, and Unwrap key	Asymmetric key operations
SHA-1	160	Digest	Digest operations
HMAC-SHA1	160	Digest (with key)	Key-Hash Message Authentication Code (HMAC)

The interfaces supported by the library follows RSA Security Inc. PKCS#11 V.2.20 specification. For more information see, PKCS#11 specifications document.

PKCS #11 API considerations

Following are the API considerations for PKCS#11:

- In PKCS#11 terminology, KCM is a soft token used for software implementation. Hardware related functions, data types, and features are not implemented by default.
- There is only one conceptual slot with slotID=0 and conceptual token is assumed to be present in the slot.
- KCM does not store public or private token objects such as keys/certificates. Following are the ramifications of this consideration:
  - KCM does not implement PIN related functions or functions that require PIN (For example, C_Login) specified by PKCS#11.
  - Session type will be R/W user functions by default. There is no distinction between R/O and R/W session types.
  - No distinction is made between user session and SO session. The user is considered as logged in by default at the point of opening a session and logged out when the session is closed.
- KCM implements CK_RV type functions and does not support CK_NOTIFY type. Hence it does not support callback functions and events.
- Multiple thread access to a single PKCS#11 session is not supported.
- There will be limited support for objects and object related functions as per the scope of APIs implemented by KCM. They are used only to invoke KCM supported PKCS#11 functions and retrieve the data returned by functions.

  KCM supports the following objects:

  - Data objects – CKO_DATA
  - Key objects - CKO_PUBLIC_KEY, CKO_PRIVATE_KEY, CKO_SECRET_KEY
- Table 1 (page 5) describes the mechanisms supported by HPUX-KCM.

Table 1 Mechanisms supported by HPUX-KCM

Mechanism	**Functions**						
	Encrypt and Decrypt	Sign and Verify	SR and VR[1]	Digest	Gen Key or Key Pair	Wrap and Unwrap	Derive
CKM_RSA_PKCS_KEY_PAIR_GEN					√		
CKM_RSA_PKCS	√	√				√	
CKM_SHA256_RSA_PKCS		√					
CKM_SHA384_RSA_PKCS		√					

Table 1 Mechanisms supported by HPUX-KCM *(continued)*

Mechanism	Functions						
CKM_SHA512_RSA_PKCS		√					
CKM_AES_KEY_GEN					√		
CKM_AES_CBC	√						
CKM_SHA_1				√			
CKM_SHA256				√			
CKM_SHA384				√			
CKM_SHA512				√			
CKM_SHA_1_HMAC		√					
CKM_SHA256_HMAC		√					
CKM_SHA384_HMAC		√					
CKM_SHA512_HMAC		√					

- HPUX-KCM implements the following PKCS#11 APIs, which are relevant for the cryptographic functions supported by KCM. Table 2 (page 6) lists the functions supported by KCM.

Table 2 Functions supported by HPUX-KCM

Category	Function	Description
General purpose functions	C_Initialize	Initializes Cryptoki
	C_Finalize	Clean up miscellaneous Cryptoki-associated resources
	C_GetInfo	Obtains general information about Cryptoki
	C_GetFunctionList	Obtains entry points of Cryptoki library functions
Slot and token management functions	C_GetSlotList	Obtains a list of slots in the system
	C_GetSlotInfo	Obtains information about a particular slot
	C_GetTokenInfo	Obtains information about a particular token
	C_GetMechanismList	Obtains a list of mechanisms supported by a token
	C_GetMechanismInfo	Obtains information about a particular mechanism
Session management functions	C_OpenSession	Opens a connection between an application and a particular token or sets up an application callback for token insertion
	C_CloseSession	Closes a session
	C_GetSessionInfo	Obtains information about the session
Object management functions	C_CreateObject	Creates an object
	C_DestroyObject	Destroys an object
Encryption functions	C_EncryptInit	Initializes an encryption operation

Table 2 Functions supported by HPUX-KCM *(continued)*

Category	Function	Description
	C_Encrypt	Encrypts single-part data
	C_EncryptUpdate	Continues a multiple-part encryption operation
	C_EncryptFinal	Finishes a multiple-part encryption operation
Decryption functions	C_DecryptInit	Initializes a decryption operation
	C_Decrypt	Decrypts single-part encrypted data
	C_DecryptUpdate	Continues a multiple-part decryption operation
	C_DecryptFinal	Finishes a multiple-part decryption operation
Message digesting functions	C_DigestInit	Initializes a message-digesting operation
	C_Digest	Digests single-part data
	C_DigestUpdate	Continues a multiple-part digesting operation
	C_DigestFinal	Finishes a multiple-part digesting operation
Signing and MACing functions	C_SignInit	Initializes a signature operation
	C_Sign	Signs single-part data
	C_SignUpdate	Continues a multiple-part signature operation
	C_SignFinal	Finishes a multiple-part signature operation
Functions for verifying signatures and MACs	C_VerifyInit	Initializes a verification operation
	C_Verify	Verifies a signature on single-part data
	C_VerifyUpdate	Continues a multiple-part verification operation
	C_VerifyFinal	Finishes a multiple-part verification operation
Key management functions	C_GenerateKey	Generates a secret key
	C_GenerateKeyPair	Generates a public-key/private-key pair
	C_WrapKey	Wraps (encrypts) a key
	C_UnwrapKey	Unwraps (decrypts) a key
Random number generation functions	C_GenerateRandom	Generates random data

For more information on APIs, see PKCS#11 specifications document.

Example usage of HPUX-KCM

```
// pkcs11 header files
#include "pkcs11_kcm.h"
#include "pkcs11.h"

// Initialize the module. Required only once during lifetime of the application
CK_RV rv = C_Initialize( NULL_PTR );
```

```
// Open session. Required for every crypto operation
CK_SESSION_HANDLE hSession;
rv = C_OpenSession( 0, 0, NULL, NULL,  );


// Set mechanism - type of crypto operation
CK_MECHANISM digestMechanism = { 0, NULL, 0 };
digestMechanism.mechanism = CKM_SHA256;

// Initialize crypto operation
rv = C_DigestInit( hSession,  );

// prepare input and output buffers
uint8_t input[] = {'a', 'b', 'c'};
uint8_t digest[64];
uint32_t inputlen = sizeof( input );
uint64_t digestlen = sizeof( digest )

// Invoke crypto operation
rv = C_Digest( hSession, input, inputlen, digest,  );

// Close crypto session
rv = C_CloseSession( hSession )

// Call this at the end of all crypto operations
rv = C_Finalize( NULL_PTR );
```

2 Installing HP-UX KCM

This chapter discusses the installation procedure for HPUX-KCM.

IMPORTANT: HP-UX KCM 1.0 requires approximately 1.5 MB of disk space after installation.

To install HP-UX KCM:

1. Log in as root.
2. Download HPUX-KCM from the HP Software Depot.
3. Save the HPUX-KCM depot as a local file on the target system.

   For example:

   in </tmp/HPUX-KCM>.depot
4. Verify the depot file on your system using the following command:

   ```
   $ swlist -d @ /tmp/HPUX-KCM.depot
   ```
5. If the HPUX-KCM depot file is correctly stored on the system, a message similar to the following is displayed after executing the command:

   ```
   # swlist -d @ /tmp/HPUX-KCM.depot
   # Initializing...

   # Contacting target "my_host"...

   #

   # Target: my_host:/tmp/HPUX-KCM.depot

   #

   #

   # Bundle(s):

   #

   HPUX-KCM A.01.00.00 HP-UX Kernel Cryptographic Module
   ```
6. Install HPUX-KCM using an interactive `swinstall` session or the following `swinstall` command:

   ```
   $ swinstall -s /tmp/HPUX-KCM.depot HPUX-KCM
   ```

   The `swinstall` utility will install the HPUX-KCM components.
7. Verify the installation using the following command:

   ```
   $ swverify HPUX-KCM
   ```

   If HPUX-KCM is installed correctly on the system, the `swverify` command will include the following text in the data it reports:

   ```
   * Verification succeeded
   ```

3 Configuring HP-UX KCM

The products integrated with HP-UX KCM must define the install-time and run-time dependency on HP-UX KCM. This helps to install and load KCM automatically along with the product dependent on HP-UX KCM.

NOTE:

- Before loading HPUX-KCM modules, ensure that `/stand/current/mod` and `/etc` directories are accessible.
- HPUX-KCM modules cannot be loaded as a static module as this is not a valid FIPS mode of operation.
- In case a Kernel configuration containing KCM modules are saved (by using `kconfig -s`), before loading the saved Kernel configuration, ensure that the KCM versions are consistent.

  For example, HPUX-KCM 1.0 is installed in a system and the Kernel configuration is saved as ‘backup’. Later KCM is upgraded to 2.0 on the same system. If for some reason, the ‘backup’ Kernel configuration is rebooted, then this leads to an inconsistent state as ‘backup’ contains HPUX-KCM 1.0, whereas the current installed version of HPUX-KCM is 2.0.

An example of defining dependency on HPUX-KCM is given below:

Install-time dependency:

```
myproduct.psf:
vendor

bundle

product

fileset

corequisites.HPUX-KCM.KCM.KCM-LIB,r>=A.01.00.00

end

end
```

Run-time dependency:

```
myproduct.modmeta:
module myproduct {
. . .
. . .
dependency libkcm_pkcs11
. . .
}
```

4 Troubleshooting

This chapter explains some of the problem scenarios that you might encounter while working with the HP-UX KCM.

General guidelines to troubleshoot HPUX-KCM

At the time of this release there are no issues reported with HPUX-KCM.

If any error occurs, HPUX-KCM logs the message into the `syslog` file. All the log messages by HPUX-KCM are prefixed with either `libkcm_core>` or `libkcm_pkcs11>` or `libkcm_nonfips>`.

To verify the errors reported by HPUX-KCM, run the command:

```
grep libkcm_ /var/adm/syslog/syslog.log
```

5 Removing HP-UX KCM

This chapter discusses the procedure to remove HP-UX KCM.

To remove HPUX-KCM:

1. Verify whether HPUX-KCM is already installed by running the following command:

   ```
   swlist -l bundle | grep -i kcm
   ```

   If HPUX-KCM is already installed on the system, a message similar to the following is displayed:

   ```
   HPUX-KCM A.01.00.00 HP-UX Kernel Cryptographic Module
   ```

2. Remove HPUX-KCM by running the following command:

   ```
   swremove HPUX-KCM
   ```

6 Support and other resources

Information to collect before contacting HP

Be sure to have the following information available before you contact HP:

- Software product name
- Hardware product model number
- Operating system type and version
- Applicable error message
- Third-party hardware or software
- Technical support registration number (if applicable)

How to contact HP

Use the following methods to contact HP technical support:

- See the Contact HP worldwide website
- Use the `GET HELP FROM HP` link on the
  HP Support Center website.
- In the United States, call +1 800 334 5144 to contact HP by telephone. This service is available 24 hours a day, 7 days a week. For continuous quality improvement, conversations might be recorded or monitored.

Documentation feedback

HP welcomes your feedback. To make comments and suggestions about product documentation, send a message to:

docsfeedback@hp.com

Include the document title and part number in your message. All submissions become the property of HP.

Typographic conventions

The following conventions are used in this document:

Book title The title of a book. On the web, this can be a hyperlink to the book itself.

`Command` A command name or command phrase, for example `ls -a`.

[ ] Optional content in syntax.

{ } Required content in syntax.

| Character that separates items in a list of choices.

... Indication that the preceding element can be repeated one or more times.

WARNING An alert that calls attention to important information that if not understood or followed can result in personal injury.

CAUTION An alert that calls attention to important information that if not understood or followed can result in data loss, data corruption, or damage to hardware or software.

IMPORTANT An alert that calls attention to essential information.

NOTE An alert that contains additional or supplementary information.

Index

Glossary

HP-UX Kernel Cryptographic Module (HP-UX KCM)

Public-Key Cryptography Standards (PKCS)

SO: A Security Officer user.

SR: Sign Recover

VR: Verify Recover