FUTURE ELECTRONICS

Security Access System

Energy Harvesting Smart Label

NXP

AEG

Engineering Tomorrow's Ideas

NXP Reader, and NFC to MCU Demo

Security Access System (SAS)

LPC43S37

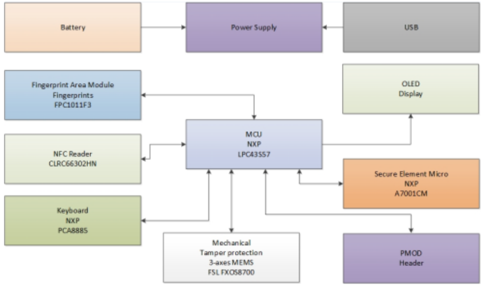
- 204 MHz ARM Cortex-M4/M0+ Dual core
- AES engine for encryption and decryption
- Extensive Communication peripherals

A7001CM

- Tamper resistant secure Micro
- Complete security platform enabling customized solutions

Capacitive Sensor (PCA8885)

- Proximity sensing
- Ruggedized keypad



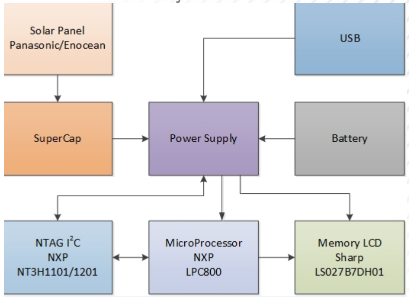
Energy Harvesting Smart Label (EnSL)

LPC824

- Ultra-efficient, 30 MHz ARM Cortex-M0+
- Versatile and Low power

NTAG I2C

- Contactless and contact interfaces
- Energy harvesting functionality



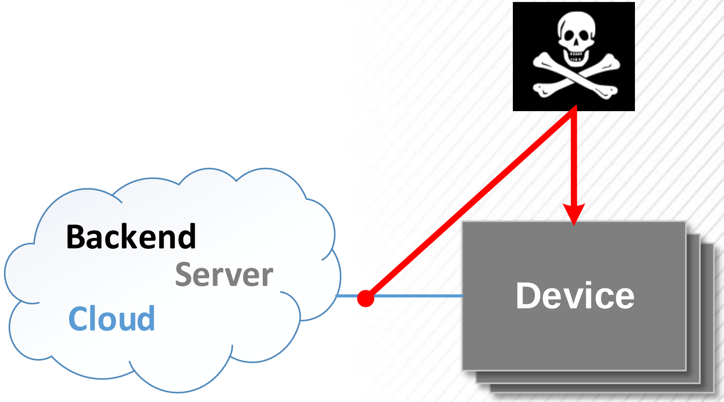
Agenda

- Motivation for Security
- What is the NXP Secure Access Demo exactly ?
- The Hardware + Firmware Implementation
- What we CAN and CANNOT offer?

Motivation for Security

Risks

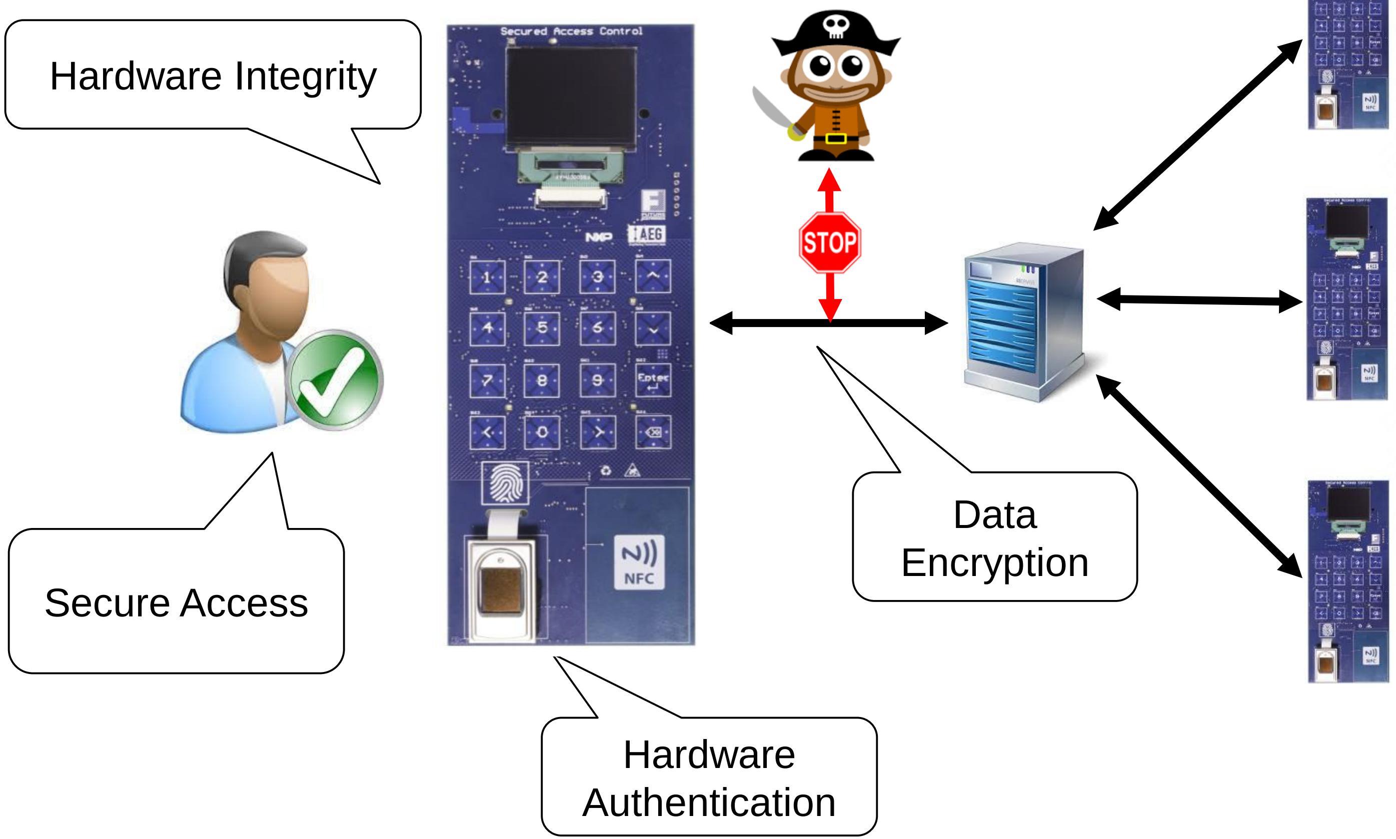
- Attacking the device
  - – Tampering with the device
  - – Counterfeit device
- Attacking the device link
  - – Stealing information (Eavesdropping)
  - – Modifying information (or Fabrication)
- Attacking the system (Denial of Service)



Requirements

- **Safety**
  - Do what you are supposed to do
- **Privacy**
  - Restrict access to user data
- **Access control**
  - Restrict access to authorized persons

Key Takeaway for Security



Hardware Integrity

Secure Access

Hardware Authentication

Data Encryption

Each measure requires <u>secure storage</u> of keys or identification assets

Hardware Security

Main Security Services

Data protection

Confidentiality
Encryption

Integrity
Hashing

Authentication Authorization

Authentication
Password
Biometry - Token
Authorization
Access rights

Software protection

Code Integrity
Code signature
Code verification

Runtime integrity

Logging & Auditing

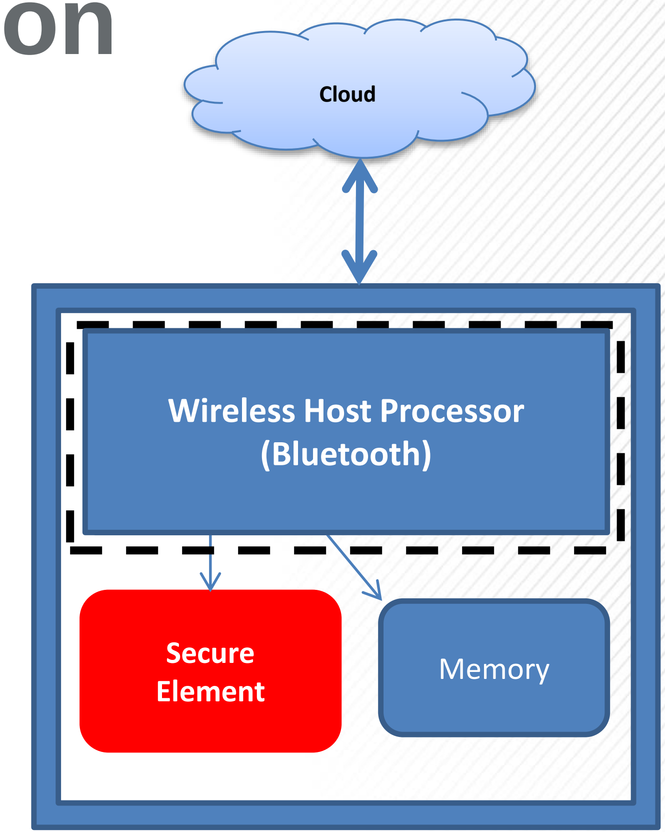
Security log
Remember actions
Auditor access
Log interpretation

Provisioning

Code Update
System upgrade
App upgrade
Bug fixing

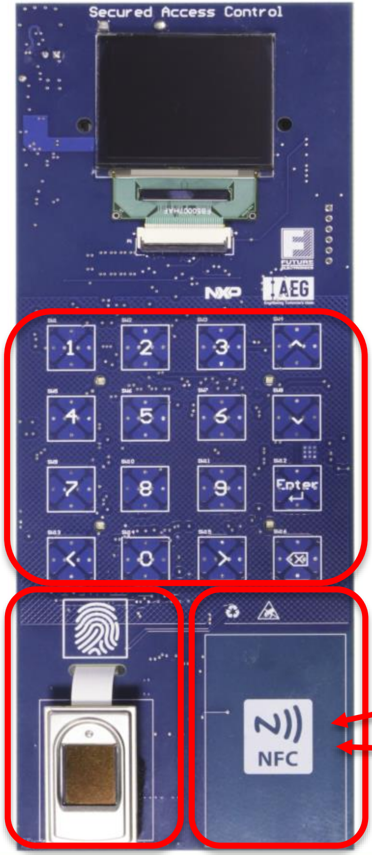
Hardware Security Solution

- Authenticate boot software
- Key storage for encrypted firmware
- Secure Firmware Update
- Node Authentication
  - Use pre-stored cert or hash to authenticate without cloud connection
- Cloud Authentication
  - Use PKI structure for mutual authentication
- Tamper resistant



Hardware Implementation

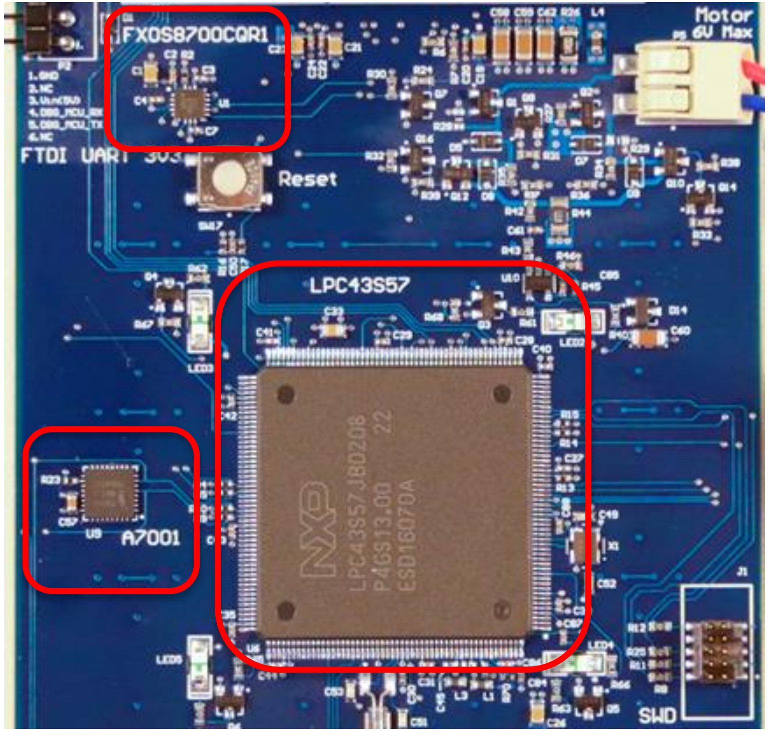
Secured Access Demo Platform



Multi-factor authentication to support user access

- ★ PIN using NXP PCA8885 (capacitive touch keyboard)
- ★ Fingerprint reader FPC1011F3 (Fingerprints security)
- ★ NXP NFC reader CLRC663

Core Security

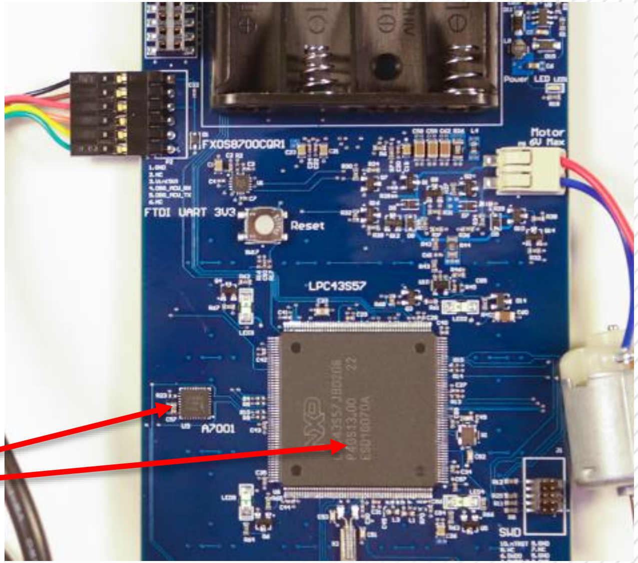


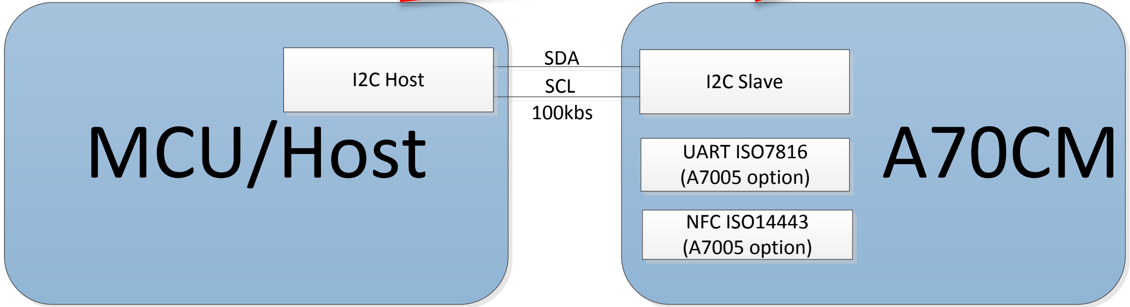
- NXP 3-Axis Accelerometer FXO8700
- NXP Secure Element A70CM
- NXP MCU with integrated security LPC43S57

Core Security

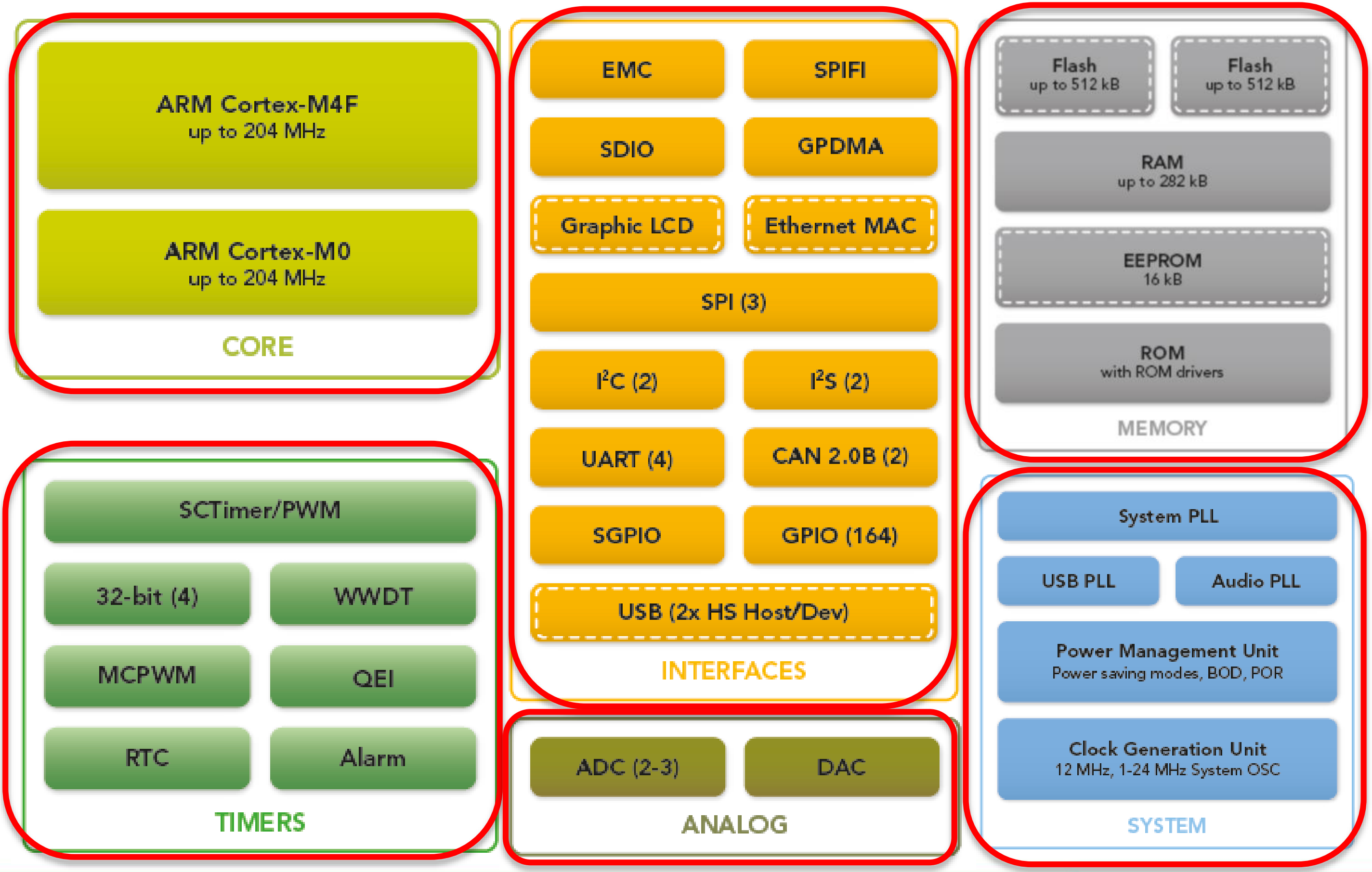
The heart of this kit is the:

1. MCU – LPC4300 Series
2. Secure Element Co-processor – A7001





LPC43S57 MCU Features



ARM Cortex-M4F
up to 204 MHz

ARM Cortex-M0
up to 204 MHz

CORE

SCTimer/PWM

32-bit (4)

WWDT

MCPWM

QEI

RTC

Alarm

TIMERS

EMC

SPIFI

SDIO

GPDMA

Graphic LCD

Ethernet MAC

SPI (3)

I^2C (2)

I^2S (2)

UART (4)

CAN 2.0B (2)

SGPIO

GPIO (164)

USB (2x HS Host/Dev)

INTERFACES

ADC (2-3)

DAC

ANALOG

Flash
up to 512 kB

Flash
up to 512 kB

RAM
up to 282 kB

EEPROM
16 kB

ROM
with ROM drivers

MEMORY

System PLL

USB PLL

Audio PLL

Power Management Unit
Power saving modes, BOD, POR

Clock Generation Unit
12 MHz, 1-24 MHz System OSC

SYSTEM

★ Dual Core MCU

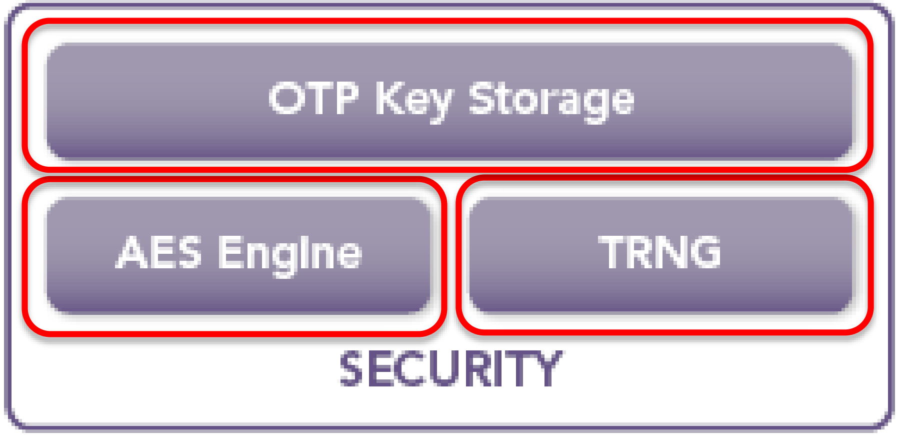
★ 1MB Flash

★ 136kB SRAM

★ High-speed Connectivity

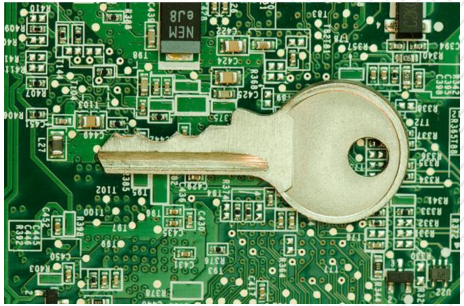
★ Advanced Peripherals

LPC43S57 Security Features



- ★ Unique Device ID
- ★ Secure Boot from encrypted image
- ★ True Random Number Generator
- ★ Hardware-accelerated AES-128 Engine
- ★ Two 128-bit nonvolatile OTP memories for encrypted keys

Private Key Storage



Where to Store Private Keys on MCU?

- **SRAM – Bad Idea …**

- **Non-Volatile Memory – Even Worse!**

- **There is NO good place to store the private key in the MCU! Especially going through a 3rd party**

- **So, the answer is …**

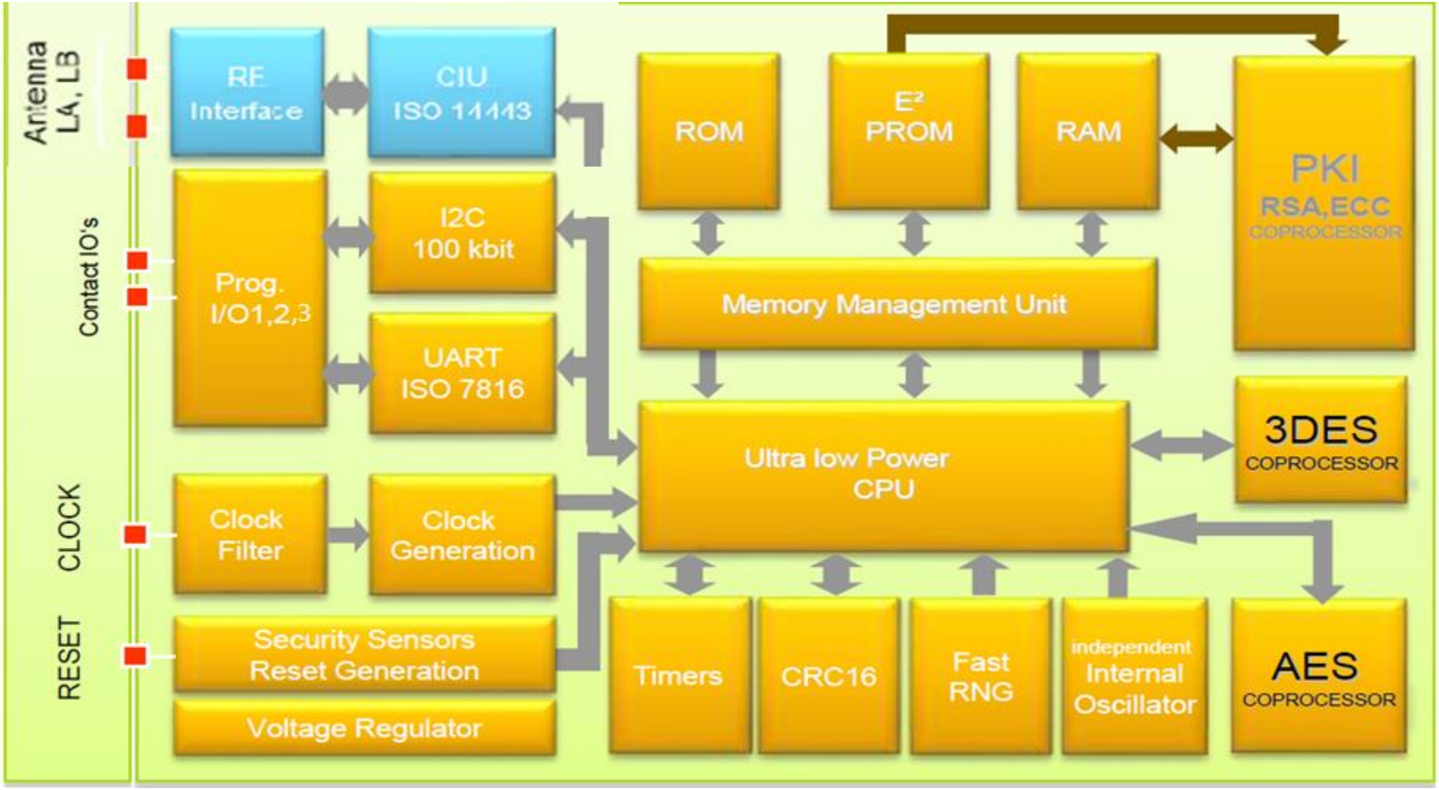
The private key MUST REMAIN in the A70CM

… NEVER store your private keys in the MCU!!

What is an A70CM?

- An integrated system with enhanced security
  - Anti-cloning
  - Key storage
  - Asymmetric/Symmetric key encryption, decryption and generation
  - Signature generation and verification
  - Authentication based on PKI
- Security OS JCOP 2.4.2 OS – Smart Card Operating System
- Card Manager Applet
  - Configuration of the cipher suites
  - Cryptographic operations
  - Trust Provisioning at different stages

A7005 Product Features

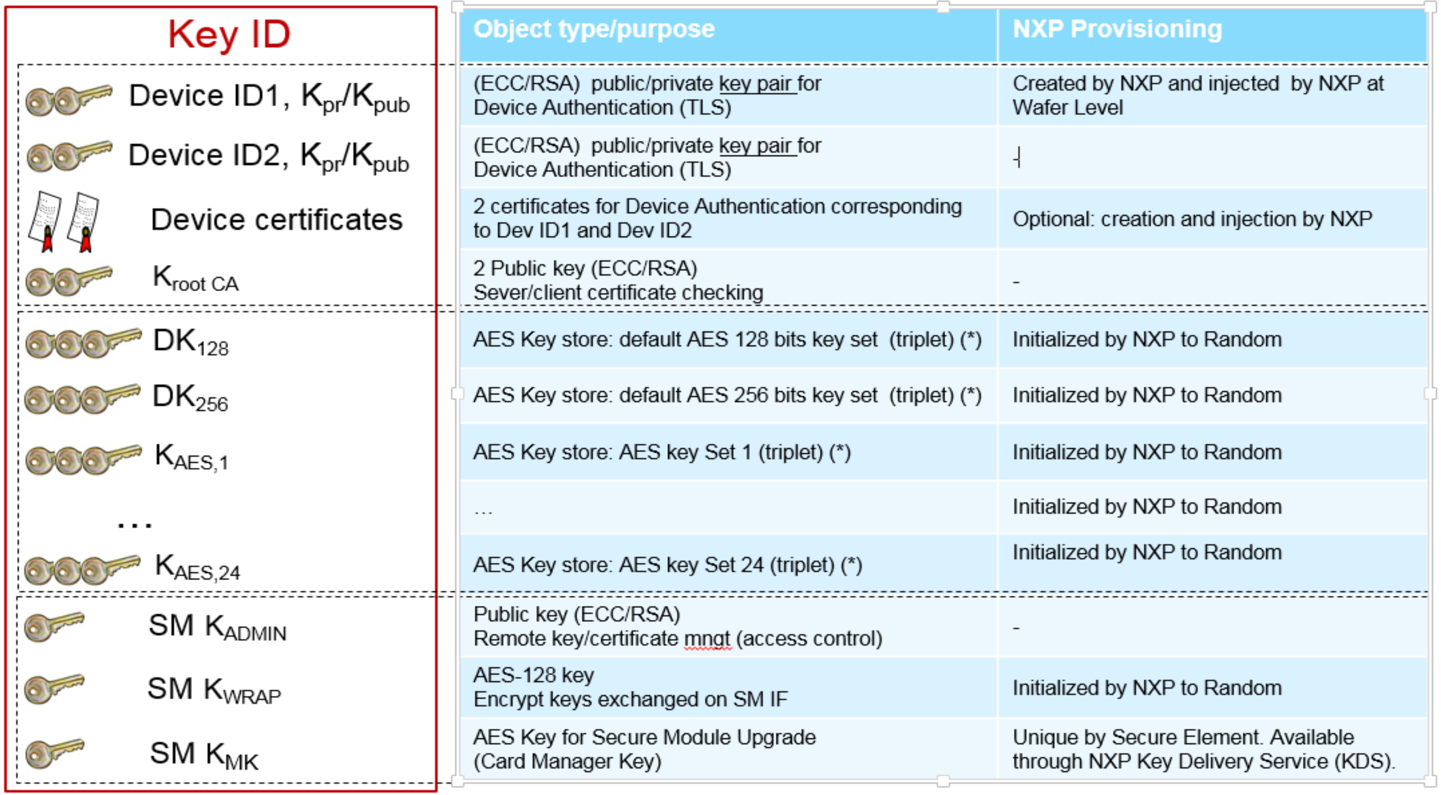


MIFARE on A7005/6 depending on the configuration

A70CM Key Features

- Public Key Infrastructure (PKI) authentication to support TLS session
- RSA/ECC key-pair generation and signature generation/verification
- RSA encryption/decryption
- AES algorithm: AES-128/256
- **Total 78 AES keys in the key store**.
  - 26 Key sets in the key store. Accessible to users
  - 1 default key-wrapping key. Invisible to user
  - 1 local encryption key. Invisible to users
- Key wrapping
- Two formats of key set
- Secure remote key management
- Trust Provisioning service in NXP certified and secure environment

A70CM Keys and Certificates

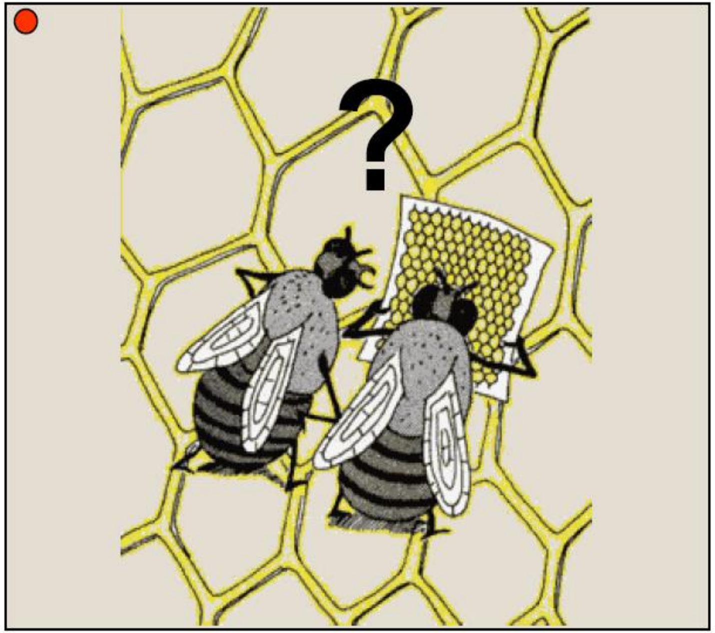


Key ID	Object type/purpose	NXP Provisioning
Device ID1, K_{pr}/K_{pub}	(ECC/RSA) public/private key pair for Device Authentication (TLS)	Created by NXP and injected by NXP at Wafer Level
Device ID2, K_{pr}/K_{pub}	(ECC/RSA) public/private key pair for Device Authentication (TLS)	-
Device certificates	2 certificates for Device Authentication corresponding to Dev ID1 and Dev ID2	Optional: creation and injection by NXP
$K_{root\ CA}$	2 Public key (ECC/RSA) Sever/client certificate checking	-
DK_{128}	AES Key store: default AES 128 bits key set (triplet) (*)	Initialized by NXP to Random
DK_{256}	AES Key store: default AES 256 bits key set (triplet) (*)	Initialized by NXP to Random
$K_{AES,1}$	AES Key store: AES key Set 1 (triplet) (*)	Initialized by NXP to Random
…	…	Initialized by NXP to Random
$K_{AES,24}$	AES Key store: AES key Set 24 (triplet) (*)	Initialized by NXP to Random
SM K_{ADMIN}	Public key (ECC/RSA) Remote key/certificate mngt (access control)	-
SM K_{WRAP}	AES-128 key Encrypt keys exchanged on SM IF	Initialized by NXP to Random
SM K_{MK}	AES Key for Secure Module Upgrade (Card Manager Key)	Unique by Secure Element. Available through NXP Key Delivery Service (KDS).

Note: Device = OEM product

(*) Eg DLMS keys (K_{MK} K_{AK} K_{EK}) or Mbus keys (K_M K_C -)

A70CM Security in Hardware



Memory:

holding secret data

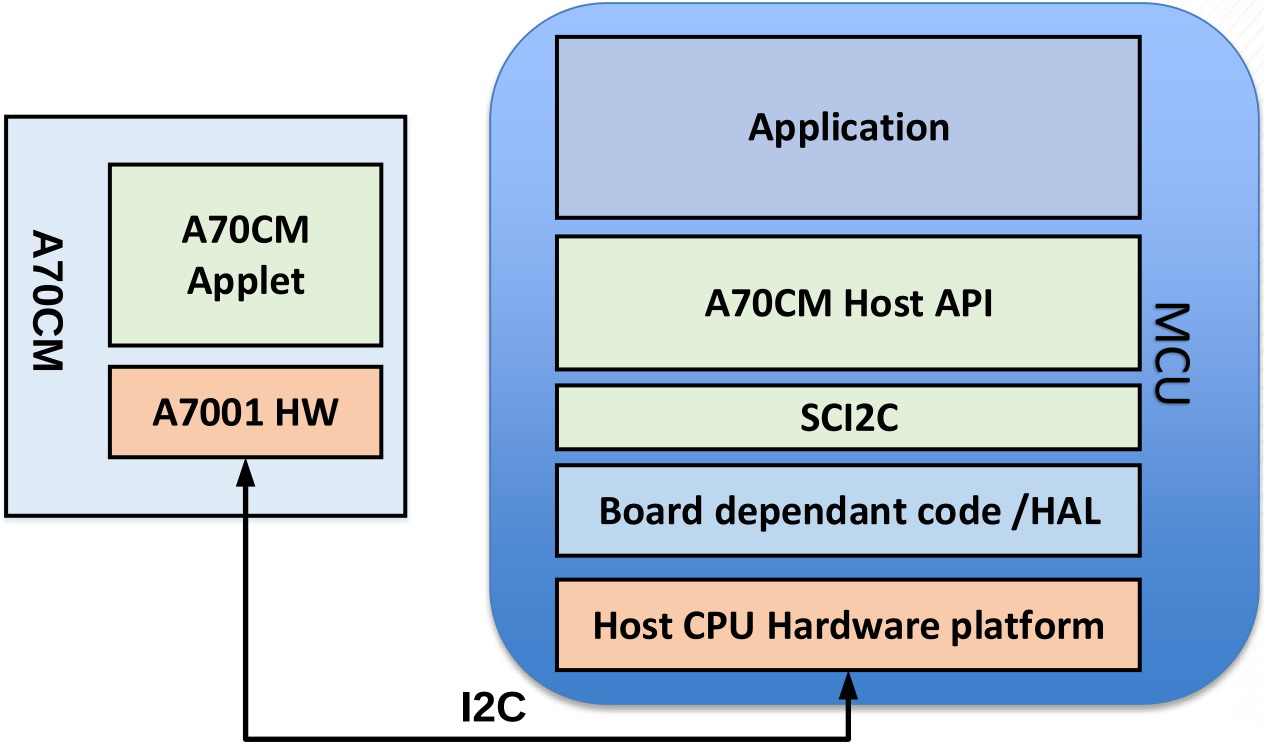
CPU with Glue Logic
+ Memory Scrambling
+ Active Shield

Active Shields

(Co-)Processor, Logic:

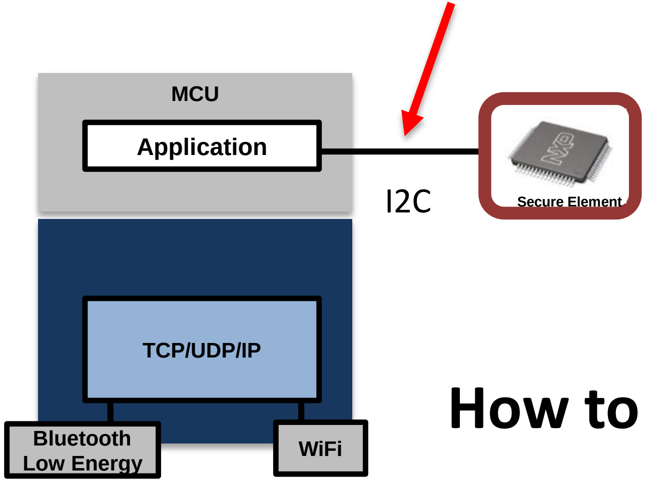
operating on secret data

A70CM: System Implementation



Transmit Keys Securely

Recall: I2C bus between LPC43S & A70CM is NOT secure



How to send keys securely on I2C?

Solution:

- Use **Key-Wrapping Key** to encrypt keys before transmitting!

Key-Wrapping Key

How to Use Key-Wrapping Key?

- Save Symmetric Key-Wrapping Key to:
  - ✓ A70CM
  - ✓ OTP key in LPC43S

Configure

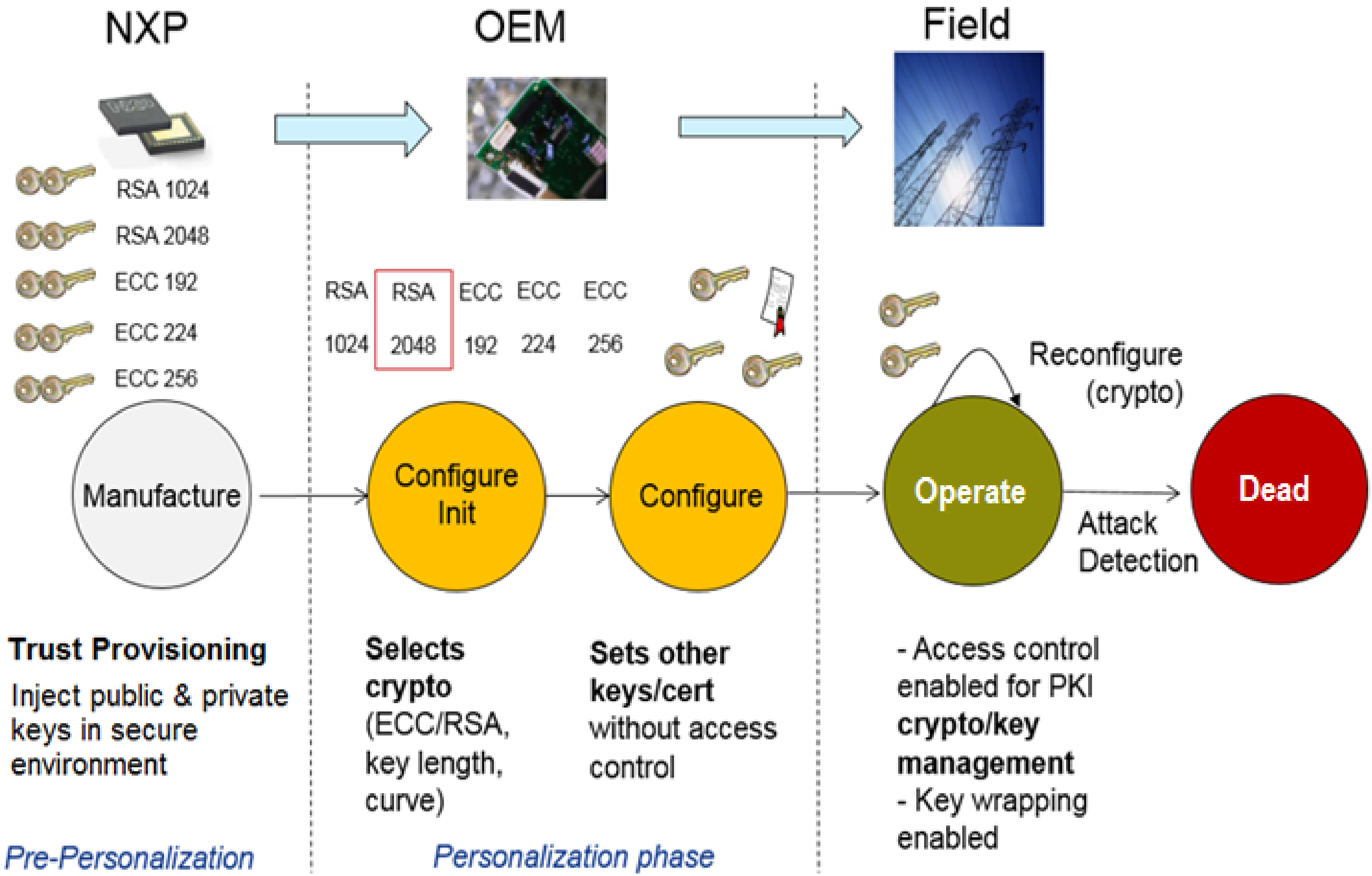
- LPC43S requests AES key from A70CM
- A70CM Key-Wraps AES key & sends to LPC43S

Operate

- LPC43S decrypts AES key with Key-Wrapping Key in OTP
- AES engine uses decrypted AES key to encrypt/decrypt

A70CM
Life Cycle and Firmware Implementation

A70CM Life Cycle



NXP

OEM

Field

RSA 1024

RSA 2048

ECC 192

ECC 224

ECC 256

RSA	RSA	ECC	ECC	ECC
1024	2048	192	224	256

Manufacture → Configure Init → Configure → Operate → Dead

Reconfigure (crypto)

Attack Detection

Trust Provisioning

Inject public & private keys in secure environment

Selects crypto (ECC/RSA, key length, curve)

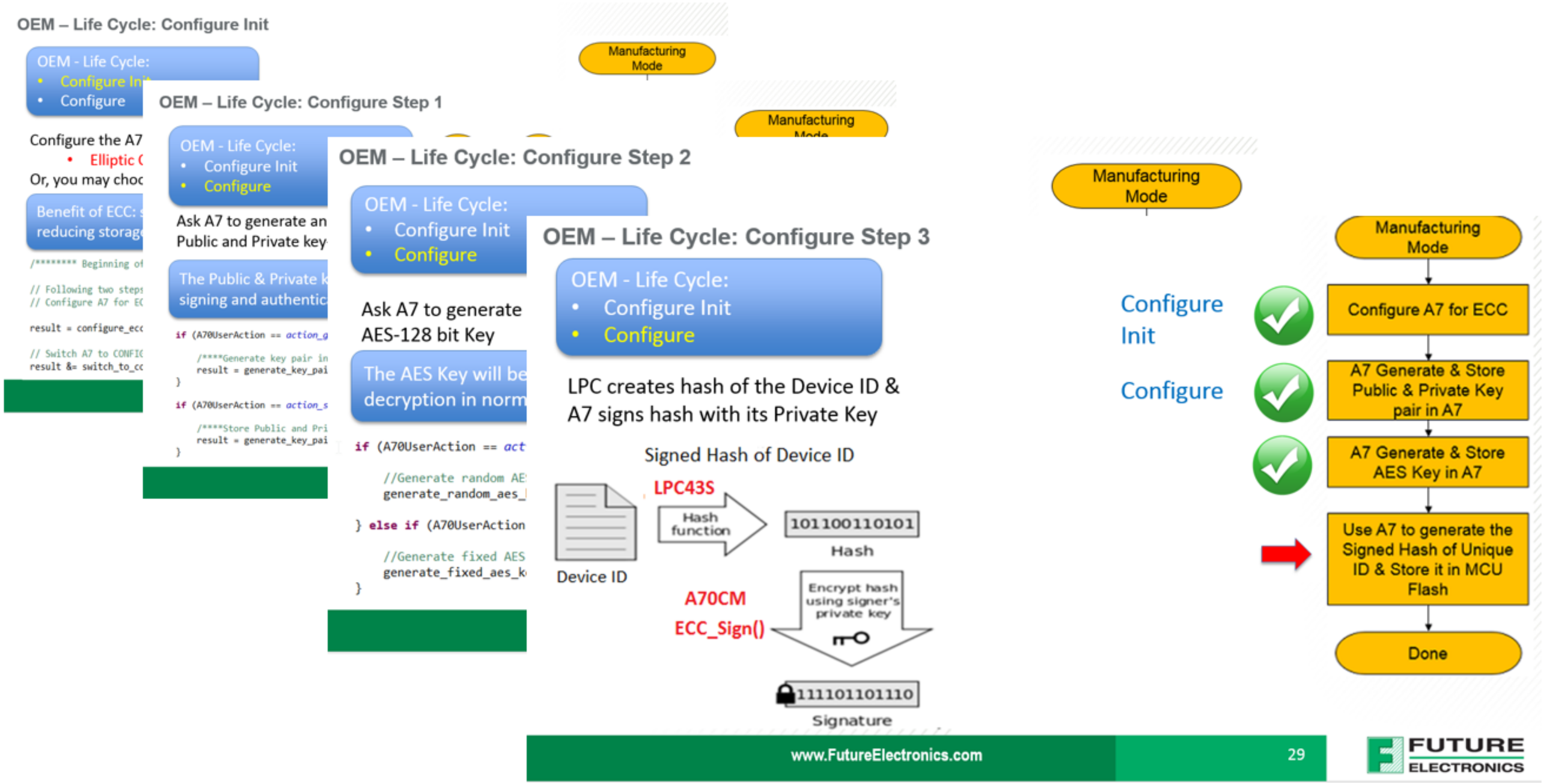
Sets other keys/cert without access control

- Access control enabled for PKI **crypto/key management**
- Key wrapping enabled

Pre-Personalization

Personalization phase

Firmware Implementation



OEM – Life Cycle: Configure Init

OEM - Life Cycle:
- Configure Ini
- Configure

Configure the A7
- Elliptic C

Or, you may choc

Benefit of ECC: s reducing storag

```
/******** Beginning of

// Following two steps
// Configure A7 for EC

result = configure_ecc

// Switch A7 to CONFIG
result &= switch_to_cc
```

OEM – Life Cycle: Configure Step 1

OEM - Life Cycle:
- Configure Init
- Configure

Ask A7 to generate an Public and Private key

The Public & Private k signing and authentic

```
if (A70UserAction == action_g

    /****Generate key pair in
    result = generate_key_pai
}

if (A70UserAction == action_s

    /****Store Public and Pri
    result = generate_key_pai
}
```

OEM – Life Cycle: Configure Step 2

OEM - Life Cycle:
- Configure Init
- Configure

Ask A7 to generate AES-128 bit Key

The AES Key will be decryption in norm

```
if (A70UserAction == act

    //Generate random AE
    generate_random_aes_

} else if (A70UserAction

    //Generate fixed AES
    generate_fixed_aes_k
}
```

OEM – Life Cycle: Configure Step 3

OEM - Life Cycle:
- Configure Init
- Configure

LPC creates hash of the Device ID & A7 signs hash with its Private Key

Signed Hash of Device ID

Device ID
LPC43S
Hash function
101100110101
Hash
A70CM
ECC_Sign()
Encrypt hash using signer's private key
111101101110
Signature
Manufacturing Mode
Configure Init
Configure
Configure A7 for ECC
A7 Generate & Store Public & Private Key pair in A7
A7 Generate & Store AES Key in A7
Use A7 to generate the Signed Hash of Unique ID & Store it in MCU Flash
Done

www.FutureElectronics.com 29

FUTURE ELECTRONICS

WE CAN

- We can share the schematic on a request
- We can share the gerbers files on request
- We can share the BOM on request
- We can help our customers with their designs needs

WE CANNOT DO:

- Provide this demo board to our customer. As we are using a “debug version” of the secure element (A7001) which customer will need a NDA with NXP to proceed.
- Give away the library for the finger print sensor. The library is not free and customer will need to license it from Fingerprints
- Give away any source code of the demo application as mentioned secure element need NDA with NXP and in some case some of our IP is in the code as well

Latest NFC Frontend Evaluation Board

NXP PN5180 NFC Frontend Evaluation Board

Target Application:

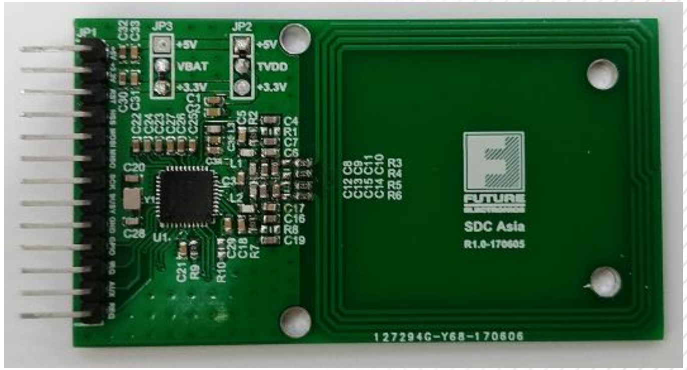
- Payment (e.g. Point-of-Sales Terminals), Physical-access, eGov, Industrial

Key Components:

- **NXP – PN5180A0HN**

Features:

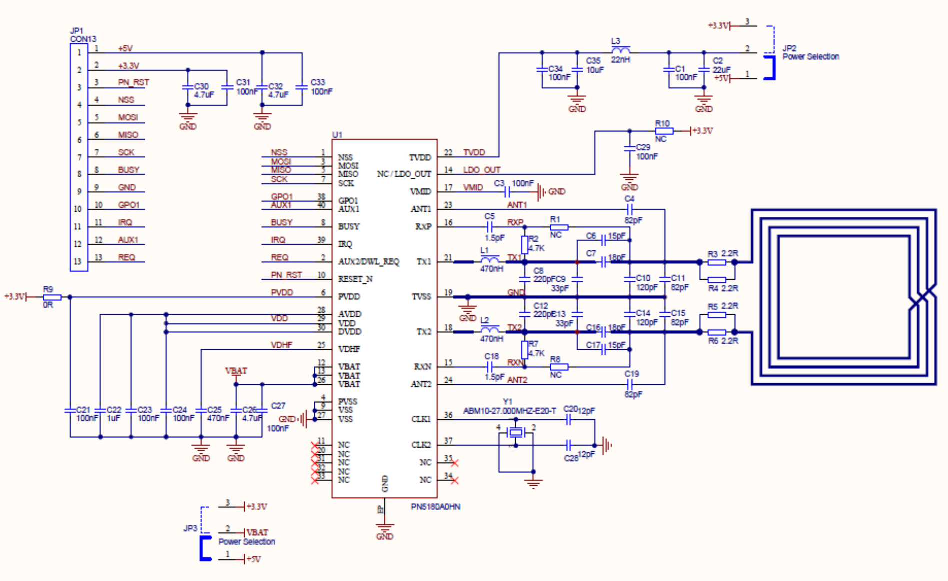
- Based on NXP PN5180 Eval Board and removed the MCU and reduce the size of the antenna to fit smaller dimension.
- PCB Dimension – 68mm x 38mm (FR-4, 2-layers board)
- Highly integrated high performance full NFC Forum-compliant frontend for contactless communication at 13.56 MHz
- Transmitter current up to 250 mA
- Dynamic Power Control (DPC) for optimized RF performance, even under detuned antenna conditions
- Adaptive Receiver Control (ARC) automatically adjusts the receiver parameters for always reliable communication
- Includes NXP ISO/IEC14443-A, Innovatron ISO/IEC14443-B and NXP MIFARE Crypto 1 intellectual property
- Full compliancy with all standards relevant to NFC, contactless operation and EMVCo
- Active load modulation supports smaller antenna in Card Emulation Mode
- Automatic EMD handling performed without host interaction relaxes the timing requirements on the Host Controller
- Low-power card detection (LPCD) minimizes current consumption during polling
- Automatic support of system LDO or system DC/DC power-down mode during LPCD
- Zero-Power-Wake-up
- One host interface based on SPI



NXP PN5180 NFC Frontend Evaluation Board

The PN5180 frontend supports the following RF operating modes:

- Reader/Writer mode supporting ISO/IEC 14443-A up to 848 kBit/s, MIFARE
- Reader/Writer mode supporting ISO/IEC 14443-B up to 848 kBit/s
- Reader/Writer mode supporting JIS X 6319-4 (comparable with FeliCa scheme)
- Supports reading of all NFC tag types (type 1, type 2, type 3, type 4A and type 4B)
- Reader/Writer mode supporting ISO/IEC 15693
- Reader/Writer mode supporting ISO/IEC 18000-3 Mode 3
- ISO/IEC 18092 (NFC-IP1)
- ISO/IEC 21481 (NFC-IP-2)
- ISO/IEC 14443-type A Card emulation up to 848 kBit/s



Item	Description	P/N	DTR #	Brand	Ref.	Qty
1	IC, PN5180A0HN/HVQFN40//C1/REEL 13 Q	PN5180A0HN		NXP	U1	1

AEG

Engineering Tomorrow's Ideas

Thank you

FUTURE ELECTRONICS