ADVANCE YOUR IoT SECURITY
LEVERAGING HARDWARE PROTECTED KEYS

DONNIE GARCIA
NXP IoT SECURITY SOLUTIONS
APRIL 2019

PUBLIC

NXP
SOLUTIONS

INTRINSIC ID

SECURE CONNECTIONS
FOR A SMARTER WORLD
Abstract

Cryptography is the basis for protecting the confidentiality, integrity and authenticity of the data within the Internet of Things ecosystem. For the IoT Edge device, the cryptographic keys used to perform the services such as encrypted boot, onboarding, and over the air updates are critical components that must be protected. Chip level hardware protected keys are the standard for achieving strong security protection for embedded designs. This session will define what a hardware protected key is and show several examples of how these keys are realized on NXP processors. Then it will dive deeper into Physical Unclonable Function (PUF) based keys that can be deployed on the vast majority of MCUs and the advantages of PUF technology. The i.MX RT 1050 family of devices will be used as a real world example of how Intrinsic ID BroadKey® SRAM based PUF can advance your IoT Security.
Agenda

- System level view of addressing IoT Security
- Hardware Protected Keys
- SRAM PUF Technology
- Implementing SRAM PUF on the i.MX RT1050 EVK
- Conclusions
IoT Security Strategies

Address the entire device lifecycle

• Once deployed processor capabilities & Cloud based monitoring ensure device lifetime integrity with hardware protected keys and secure boot for every device power up

Scale to align to end product needs

• Security technology is rooted in MCU/MPU hardware capabilities at many processor integration and performance points (NXP: A71xx, SE050, i.MX, Layerscape, Kinetis, LPC, JN)

Be easy to deploy and easy use

• Fully Documented steps and procedures from installing bootstrap through decommissioning stage (NXP: Edgescale documentation)
SYSTEM LEVEL VIEW OF ADDRESSING IoT SECURITY
Design Challenges across device lifecycle

1. **Procure**
   - Ensure design integrity of all hardware components

2. **Develop**
   - Identify all users and processes and assign roles and privileges
   - Define and create minimal secure state for secure backup functionality
   - Protect secure services with logical security
   - Develop a staged bootloader flow that authenticates all components

3. **Manufacture**
   - Ensure chips are in NXP factory configured state
   - Install secret and binding material
   - Begin device management with logging of Unique IDs
   - Apply programming and assembly policies

4. **Deploy**
   - Onboard to Device Management and Device User Cloud
   - Install first Software update/Software profile
   - Log device binding data (Region, Networks, Users)

5. **Use**
   - Enforce integrity checks for users, processes and communications
   - Enforce Logical and physical security checks with tamper and processor firewalls
   - Encrypt sensitive data
   - Log device binding data (Region, Networks, Users)

6. **Maintain**
   - Monitor audit logs and respond to any issues
   - Update firmware to mitigate any vulnerabilities
   - Maintain keys/certificates and revocation list
   - Place processing units into decommissioned state

7. **Decommission**
   - Destroy all sensitive user information and device unique keys
Security scope spans across multiple domains:
- Numerous device form factors and services
- Cloud User services and Device Management
- Certifications, regional standards and other proof points
NXP Solutions for Edge Computing

IoT Nodes

Edge Gateways

Cloud Infrastructure

NXP: SE050, LPC, Kinetis, i.MX-RT

NXP Layerscape, i.MX Family

NXP EdgeScale Suite

Data Analytics

Machine Learning

Multiple Cloud Frameworks

Application Management

Secure Device Management

Customer Solution

App

Thin Edge Agent

Edge Agent

RTOS, Linux, Android

Middleware

Middleware

RTOS, LPC, Kinetis, i.MX-RT
Can functionally overlap in runtime*  

**IDENTITY SERVICES**  
Unique ID  
Authentication  
Session establishment  

**RUNTIME SERVICES**  
Secure Boot  
Firmware Management  
Bulk Data Protection  
Application Roles  
Client side logging  

**MANAGEMENT SERVICES**  
Device Auditing  
Software management  
Cloud to cloud onboarding  

*NXP IoT Security ICs: A100x Authenticator  
A71CH  
SE050  

Security Hardening on MCU/MPU  
NXP Processors  

Device Management  
Edgescale  

*With secure provisioning identity can be established in a processor
Device Level Security Solution

- Security scope at the device level
  - Hardware
    - SoC specific security technology
  - Software and Tools
    - Logical Security implementation
    - Trusted Execution Environment
  - Documentation
    - Security Policies
    - Internal/External Documentation
  - Certifications
    - Third-Party analysis
  - Partner/External
    - Cloud services
System Level Security Goals Depend on Cryptography

- **Cryptography is a fundamental capability needed to address edge device security**
  - Basis for protecting data at rest and in transit
  - Provides robust identity for the end device by cryptographic authentication
- **The key material used for cryptographic operations must be protected by hardware**
  - Attacks against Confidentiality/Integrity/Authenticity are aimed at attaining the Cryptographic Key
Secure Edge Architectures

**Security Architectures supported by current shipping NXP products**

Add Trusted Execution based on ARM TrustZone® and/or isolation features\(^2\) on the SoC

1) Not mandatory for MCUs/MPUs when they have embedded memory;
2) Features like RDC (Resource Domain Controller) on i.MX
HARDWARE PROTECTED KEYS
Defining Hardware Protected Keys

**protect**

*verb*

past tense: protected; past participle: protected

- keep safe from harm or injury.
- "he tried to **protect** Kelly from the attack"
- synonyms: keep safe, keep from harm, save, safeguard, shield, preserve, defend, cushion, shelter, screen, secure, fortify, guard, mount/stand guard on; More
- antonyms: expose, neglect, attack, harm
- aim to preserve (a threatened plant or animal species) by legislating against collecting or hunting.
- restrict by law access to or development of (land) so as to preserve its natural state.
- "logging is continuing in protected areas in violation of an international agreement"
- synonyms: secured, sheltered, in safe hands, safe, guarded, out of danger, safeguarded, preserved, the nation's largest protected wetland
- (of an insurance policy) promise to pay (someone) an agreed amount in the event of loss, injury, fire, theft, or other misfortune.
- "in the event of your death, your family will be protected against any financial problems that may arise" (ECONOMICS)
- shield (a domestic industry) from competition by imposing import duties on foreign goods.

- COMPUTING
  - restrict access to or use of (data or a memory location).
  - "security products are designed to protect information from unauthorized access"
Protected over the lifecycle* of the Cryptographic keys

• Key Lifecycle
  - Generation
    ▪ Who/what creates the key material
  - Establishment
    ▪ How the key material is shared or signed between entities
  - Storage
    ▪ Where the key material is placed for future access
  - Use
    ▪ How the key is utilized during the cryptographic processing
  - Decommission
    ▪ Revocation and destruction of key material

*Key Lifecycle_ [https://community.nxp.com/docs/DOC-333095](https://community.nxp.com/docs/DOC-333095)
Protected from attacks

- Malicious manufacturer steals secrets during the provisioning process
- Failure of the Certificate Authority or device database attacks
- Reverse engineering of silicon hardware or software
- Side channel attacks during cryptographic processing
- Theft of device before destruction of secrets
HW Protected Keys Example 1: Dedicated Security ICs

- **NXP IoT Security ICs:**
  - A71CH
  - A100x Secure Authenticator
  - SE050

- Premier example of a Hardware Protected Key integrated circuit

- Derived from CC certified solutions
  - Protects key generation and establishment with optional provisioning provided by NXP or qualified partners
  - Protected storage with bank grade tamper resistance in the design of the IC
  - Resistance to side channel attacks to protect the use of the keys
HW Protected Keys Example 2: MCU/MPU Security hardening

- Devices such as NXP i.MX products integrate security technology for protecting keys
  - Fuse locations for key material with read out protection for protected storage of key or key material
  - Keys/key material are passed to hardware accelerators without software interaction for protected use
  - Access to the use of keys is restricted by security state machine requiring authenticated boot
  - Zero-izable keys with tamper monitors for decommissioning
HW Protected Keys Example 3: Software PUF

- **Intrinsic ID** has a software based implementation of a cryptography library based on a cryptographic key derived from a patented SRAM Physical Unclonable Function
  - Key generation is device unique and unclonable based on the SRAM PUF technology
  - Key is ephemeral and not stored so is protected from physical attacks
  - BroadKey SW is developed to meet FIPS 140-2 Appendix B and applies countermeasures for side channel attacks
  - Destroying the activation code decommissions the key and protected key material
Recently launched LPC5500 family also makes use of Intrinsic ID SRAM PUF technology in the design of the microcontroller in addition to other security capabilities.

Unique Security Enhancements

A cornerstone to establishing device trustworthiness is NXP’s ROM-based secure boot process that utilizes device-unique keys to create an immutable hardware ‘root-of-trust’. The keys can now be locally generated on-demand by an SRAM-based Physically Unclonable Function (PUF) that uses natural variations intrinsic to the SRAM bitcells. This permits closed loop transactions between the end-user and the original equipment manufacturer (OEM), thus allowing the elimination of third-party key handling in potentially insecure environments. Optionally, keys can be injected through a traditional fuse-based methodology.

Furthermore, NXP’s SEE improves the symmetric and asymmetric cryptography for edge-to-edge, and cloud-to-edge communication by generating device-unique secret keys through innovative usage of the SRAM PUF. The security for public key infrastructure (PKI) or asymmetric encryption is enhanced through the Device Identity Composition Engine (DICE) security standard as defined by the Trusted Computing Group (TCG). SRAM PUF ensures confidentiality of the Unique Device Secret (UDS) as required by DICE. The newly announced solutions support acceleration for asymmetric cryptography (RSA 1024 to 4096-bit lengths, ECC), plus up to 256-bit symmetric encryption and hashing (AES-256 and SHA2-256) with MbedTLS optimized library.

“Maintaining the explosive growth of connected devices requires increased user trust in those devices,” said John Ronco, vice president and general manager, Embedded & Automotive Line of Business, Arm. “NXP’s commitment to securing connected devices is evident in its new Cortex-M33 based products built on the proven secure foundation of TrustZone technology, while incorporating design principles from Arm’s Platform Security Architecture (PSA) and pushing the boundaries of Cortex-M performance efficiency.”
Exploring Protected Key Options

1. External Security IC
   - Security Hardening on MCU/MPU with Software PUF (Intrinsic ID BroadKey)
   - Strongest protection across all key life stages
   - Uses:
     - Device identity and establishing TLS/onboarding
     - NXP Trust provisioning reduces overhead for key generation and establishment

2. Security with OTP Keys
   - Security Hardening on MCU/MPU
   - Provides runtime application security
   - Uses:
     - Secure boot
     - Bulk data protection
     - Enforces security policies (Roles)
     - Firmware updates

3. Software SRAM PUF
   - Uses Incremental
   - Assists with early key life stages and improves protection for keys
   - Uses:
     - Key Generation and establishment
     - Device identity
     - Assist with TLS/onboarding

4. Security w/SRAM PUF
   - Hardware PUF (Intrinsic ID QuiddiKey): LPC5500 Family
   - Links advantages of PUF to runtime application security
   - Uses:
     - PUF protected keys used for secure boot, etc.
     - PUF for Key generation and establishment protects early life stages
PUF TECHNOLOGY
SRAM PUF Overview

Leverages the intrinsic entropy of the silicon manufacturing process

Device unique, unclonable fingerprint derived on every activation of the PUF

PUF master key is used to protect other secrets

1. Process Variation
   Naturally occurring variations in the attributes of transistors when chips are fabricated (length, width, thickness)

2. SRAM Start-up Values
   Each time an SRAM block powers on the cells come up as either a 1 or a 0

3. Silicon Fingerprint
   The start-up values create a random and repeatable pattern that is unique to each chip

4. SRAM PUF Key
   The silicon fingerprint is turned into a secret key that builds the foundation of a security subsystem
Using PUF Technology

SRAM Start-up Pattern

NO SECRETS STORED ON CHIP

SRAM PUF IP

AC = Helper Data

NO SECRETS STORED ON CHIP
SRAM PUF Disruptive Physical protection

SRAM PUF Technology
- Key generated by device entropy
- No traces of sensitive data in the embedded system

Other Solutions
- Key programmed externally
- Permanent physical alteration
- Key visible in structure
Intrinsic ID BroadKey

- **BroadKey-Pro** (most feature complete offering)
  - Device-unique key derivation
  - Random number generation
  - Wrapping and management, including elliptic curve private key generation and storage, importing and exporting of public keys, signature generation and verification
  - Key agreement functionality and public key encryption and decryption
BroadKey-Pro API summary and Uses

• **API Summary**
  - **BASE**
    - Init, Enroll, Start, Stop
  - **Key and RNG Generation**
    - Symmetric and Asymmetric keys, Random numbers
  - **Wrap/Unwrap**
    - Handle key material
  - **Public Key Management**
    - Derive, import, export for public keys
    - Create private key code
    - ECDSA, ECDH

• **USES**
  - **Key provisioning**
    - At manufacturing or at deployment
  - **Transport Layer Security**
    - Integrated with TLS library
  - **Securing data at rest**
    - Linked to specific device
    - Binding SW
  - **OTA Firmware update**
    - Secure operation with confidentiality, authenticity and integrity
Getting BroadKey

• BroadKey Software IP is delivered as a library compiled for a specific target chip, along with interface specifications and user manual.
  - NXP Request from Intrinsic ID based BroadKey for a specific platform
    ▪ (ie. i.MX RT) and IDE (MCUXpresso IDE)
UTILIZING PUF ON THE i.MX RT1050 EVALUATION KIT
Using BroadKey: MCUXpresso Demo Application

- Steps needed to use the BroadKey delivery from Intrinsic ID
  - Install MCUXpresso IDE
  - Import the demonstration project
  - Connect the iMX RT EVK board
  - Run the demo from the debugger
  - See Output on the debug Terminal
BroadKey Demo

Readme included in the demo

---

**Terminal output**

```
COMSO-Tera Term VT

File Edit Setup Control Window Help

Found the HyperFlash by GPI

--- BK Version ---
product_id : B
major_version : 2
minor_version : 4
patch : 0
build_number : 0

--- BK Init ---
bk_init ...done!

--- BK Enroll or Start ---
bk_start ...done!

--- BK Get Random Bytes ---
bk_generate_random ...done!

The Random generated bytes are:

8xF7 8xB5 8x48 8x59 8x74 8x2F 8xC4 8x19 8x4C 8x0C 8x43 8x7E 8x05 8x31 8x83

--- BK Get Symmetric Key ---
bk_get_key ...done!

The Symmetric Key is:

033D 0xC8 0x7B 0xEE 0x31 0xF1 0x4D 0x05 0x7E 0x89 0x80 0x80 0xF1 0xE4 0xCE 0xA4

--- BK Get ECC Private Key ---
bk_get_private_key ...done!

The generated ECC Private Key is:

0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C 0x3C

--- BK Wrap and Unwrap the (user) key ---
The User Key to be wrapped is:

0x66 0x4B 0x77 0xCE 0x0E 0x51 0x71 0x6A 0x73 0x3C 0x92 0x82 0x8F 0xCB 0x0C

bk_wrap ...done!
The Key Code is:

0x66 0x36 0x85 0x18 0x80 0x8C 0x93 0x4D 0x2C 0x82 0x77 0x49 0xF3 0x27 0xE6 0xEC
```
BroadKey Documentation

- Robust and detailed documentation covering all APIs
  - Great for understanding the life stages of PUF keys
  - Includes performance benchmarking for Arm Cortex-M devices
  - Must read document
BroadKey Demo Summary

- Demo utilizes External Flash Memory to store the PUF activation code and has linker file configuration aligned to BroadKey requirements
- Demo is executed from internal SRAM
- Demo provides the base functionality to see BroadKey across the key life cycle
  - BroadKey is initialized, enrolled to generate AC (if needed), then used
- Demonstration of key wrap and unwrap functions showing protected key storage
- Demo allows erasure of the AC (Activation Code) to Decommission the Cryptographic Context
**BroadKey Demo (API Example Only)**

- Currently the demo resides in SRAM, but the predominant use case for i.MX RT series is **Execute in Place (XiP)**
  - Performing XiP and writing an Activation Code (AC) to the external flash needs special care at the application level
  - Intrinsic ID has 2 versions of Broadkey, one for provisioning and one for OEM application use
- Currently the demo completely shows the BroadKey API
  - Application cases such as OTA and Cloud connection to AWS IoT/Google/MS Azure core are future work
- i.MX RT security features add security protections to the system using BroadKey
  - Secure Boot, Encrypted Boot, and encrypted XIP ensure the integrity and confidentiality of Broadkey
  - Hardware firewalls could establish trusted execution of BroadKey
CONCLUSION
Why Intrinsic ID BroadKey?

• Breakthrough technology aligned to IoT Security Strategies for scalability and ease of use
  - Protection of keys throughout the key lifecycle
  - APIs to support a broad range of uses

• Alignment to strategic needs when addressing IoT
  - Portable to many MCU/MPU types
  - Scalable key strength and functionality
  - Easy to deploy and use

<table>
<thead>
<tr>
<th>BroadKey Configurations</th>
<th>Safe</th>
<th>Plus</th>
<th>Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Strength (bits)</td>
<td>128/256</td>
<td>128/256</td>
<td>128/256</td>
</tr>
<tr>
<td>PUF (KB) related to Security Strength</td>
<td>0.7/1</td>
<td>0.7/1</td>
<td>0.7/1</td>
</tr>
<tr>
<td>Code Size (KB)</td>
<td>8</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>Generate Device Keys and Random Values</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Wrap and Unwrap Application Keys</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Public Key Management and Crypto Operations</td>
<td></td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
Why NXP i.MX RT Series?

High Performance
Real-Time Processing
• Cortex-M7 up to 600MHz (50% faster than current existing M7 products)
• 20ns interrupt latency
• Up to 1MB Tightly Couple Memory

Low BOM Cost
• Competitive pricing starting @ $1.48 10k RSL
• Fully integrated PMIC with DC-DC
• Low cost package, 10x10 BGA, enabling 4 Layer PCB design
• SDRAM interface

High level of Integration
• High Security enabled by AES-128, HAB and On-the-fly QSPI Flash Decryption
• 2D graphics acceleration engine
• Parallel camera sensor interface
• LCD display controller up to WXGA (1366x768)
• Audio interface with three I2S for multichannel high performance audio

Easy to Use
• MCU customers can leveraging their current toolchain (MCUXpresso, IAR, Keil)
• Rapid and easy prototyping and development with NXP FreeRTOS, SDK, ARM mbed and the global ARM ecosystem
• Single voltage input simplifies power circuit design
• Scalability to Kinetis & i.MX products
ECDSA P256 Sign and Verify Times on i.MX RT

- For 256 bit curve strength ECDSA sign and verify complete in less than 4 million CPU cycles
  - For 600MHz CPU clock that results in sign and verify times <7milliseconds
HW Protected Keys Example 4: Hardware PUF

- Recently launched LPC5500 family also makes use of Intrinsic ID SRAM PUF technology in the design of the microcontroller in addition to other security capabilities
Conclusions

• In today’s threat landscape, all IoT devices must address security
• Cryptography is a common component in securing IoT Designs
• NXP device options exist to protect the cryptographic keys for embedded designs
• Intrinsic ID BroadKey on the i.MX RT working together combine to create a solution aligned to the need for addressing lifecycle, scalability and ease of use
Conclusions

• In today’s threat landscape, all IoT devices must address security
• Cryptography is a common component in securing IoT Designs
• NXP device options exist to protect the cryptographic keys for embedded designs
• Intrinsic ID BroadKey on the i.MX RT working together combine to create a solution aligned to the need for addressing lifecycle, scalability and ease of use

Questions & Answers Session