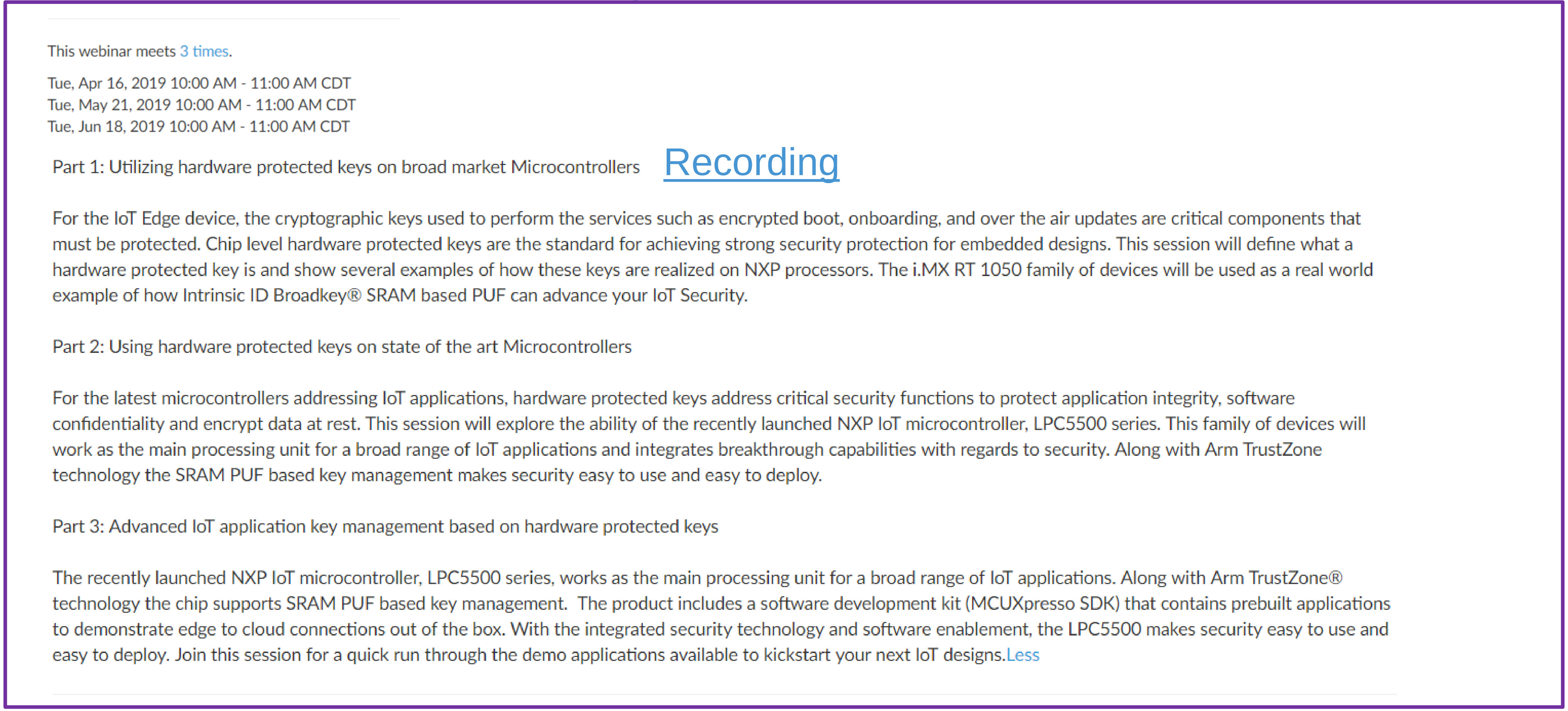
# ADVANCE YOUR IoT SECURITY LEVERAGING HARDWARE PROTECTED KEYS

DONNIE GARCIA
NXP IoT SECURITY SOLUTIONS
MAY 2019

PUBLIC

SECURE CONNECTIONS
FOR A SMARTER WORLD

# Hardware Protected Keys Webinar Series



This webinar meets 3 times.

Tue, Apr 16, 2019 10:00 AM - 11:00 AM CDT
Tue, May 21, 2019 10:00 AM - 11:00 AM CDT
Tue, Jun 18, 2019 10:00 AM - 11:00 AM CDT

Part 1: Utilizing hardware protected keys on broad market Microcontrollers Recording

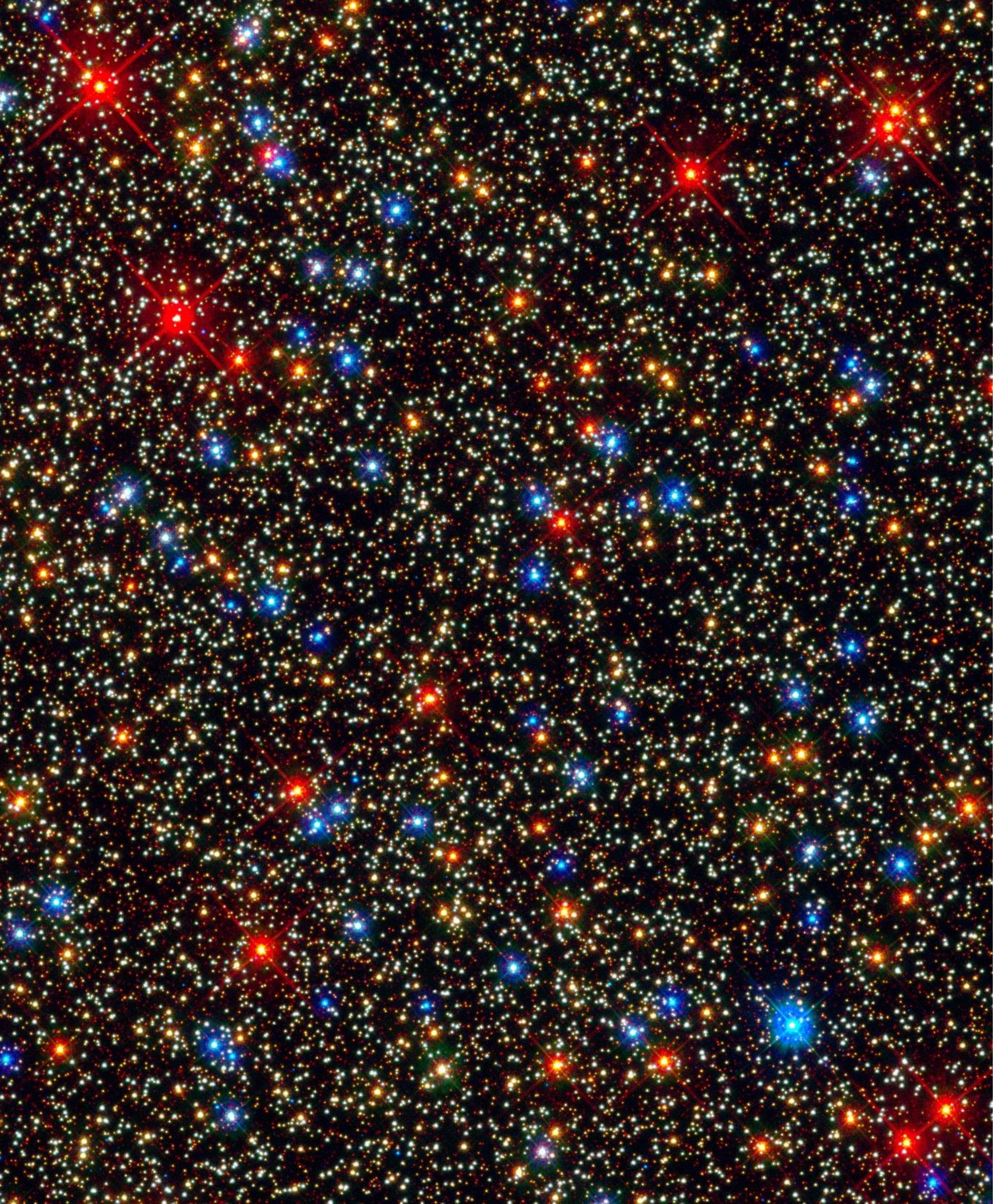
For the IoT Edge device, the cryptographic keys used to perform the services such as encrypted boot, onboarding, and over the air updates are critical components that must be protected. Chip level hardware protected keys are the standard for achieving strong security protection for embedded designs. This session will define what a hardware protected key is and show several examples of how these keys are realized on NXP processors. 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.

Part 2: Using hardware protected keys on state of the art Microcontrollers

For the latest microcontrollers addressing IoT applications, hardware protected keys address critical security functions to protect application integrity, software confidentiality and encrypt data at rest. This session will explore the ability of the recently launched NXP IoT microcontroller, LPC5500 series. This family of devices will work as the main processing unit for a broad range of IoT applications and integrates breakthrough capabilities with regards to security. Along with Arm TrustZone technology the SRAM PUF based key management makes security easy to use and easy to deploy.

Part 3: Advanced IoT application key management based on hardware protected keys

The recently launched NXP IoT microcontroller, LPC5500 series, works as the main processing unit for a broad range of IoT applications. Along with Arm TrustZone® technology the chip supports SRAM PUF based key management.  The product includes a software development kit (MCUXpresso SDK) that contains prebuilt applications to demonstrate edge to cloud connections out of the box. With the integrated security technology and software enablement, the LPC5500 makes security easy to use and easy to deploy. Join this session for a quick run through the demo applications available to kickstart your next IoT designs.Less

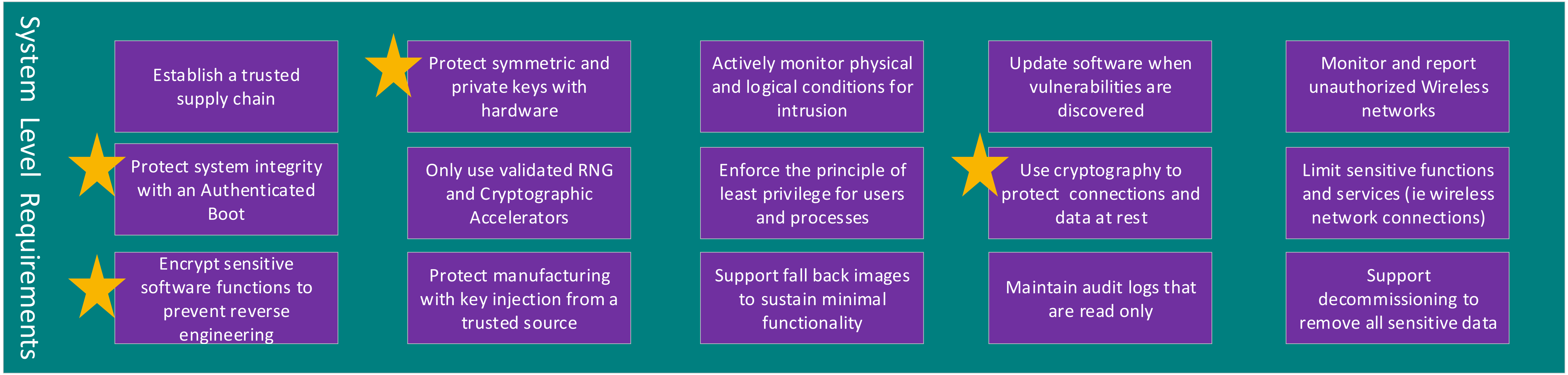


# Agenda

- Quick recap and highlights
- LPC5500 Series Overview
- Security Model & Security Technology
- LPC55S6xx Security Technology
  - Security Subsystem
  - Arm Trustzone
  - Secure Debug
- LPC5500 PUF based Key Management
- Conclusions

# QUICK RECAP & HIGHLIGHTS

# System Level Security Goals Depend on Cryptography



System Level Requirements

- Establish a trusted supply chain
- ★ Protect system integrity with an Authenticated Boot
- ★ Encrypt sensitive software functions to prevent reverse engineering
- ★ Protect symmetric and private keys with hardware
- Only use validated RNG and Cryptographic Accelerators
- Protect manufacturing with key injection from a trusted source
- Actively monitor physical and logical conditions for intrusion
- Enforce the principle of least privilege for users and processes
- Support fall back images to sustain minimal functionality
- Update software when vulnerabilities are discovered
- ★ Use cryptography to protect connections and data at rest
- Maintain audit logs that are read only
- Monitor and report unauthorized Wireless networks
- Limit sensitive functions and services (ie wireless network connections)
- Support decommissioning to remove all sensitive data

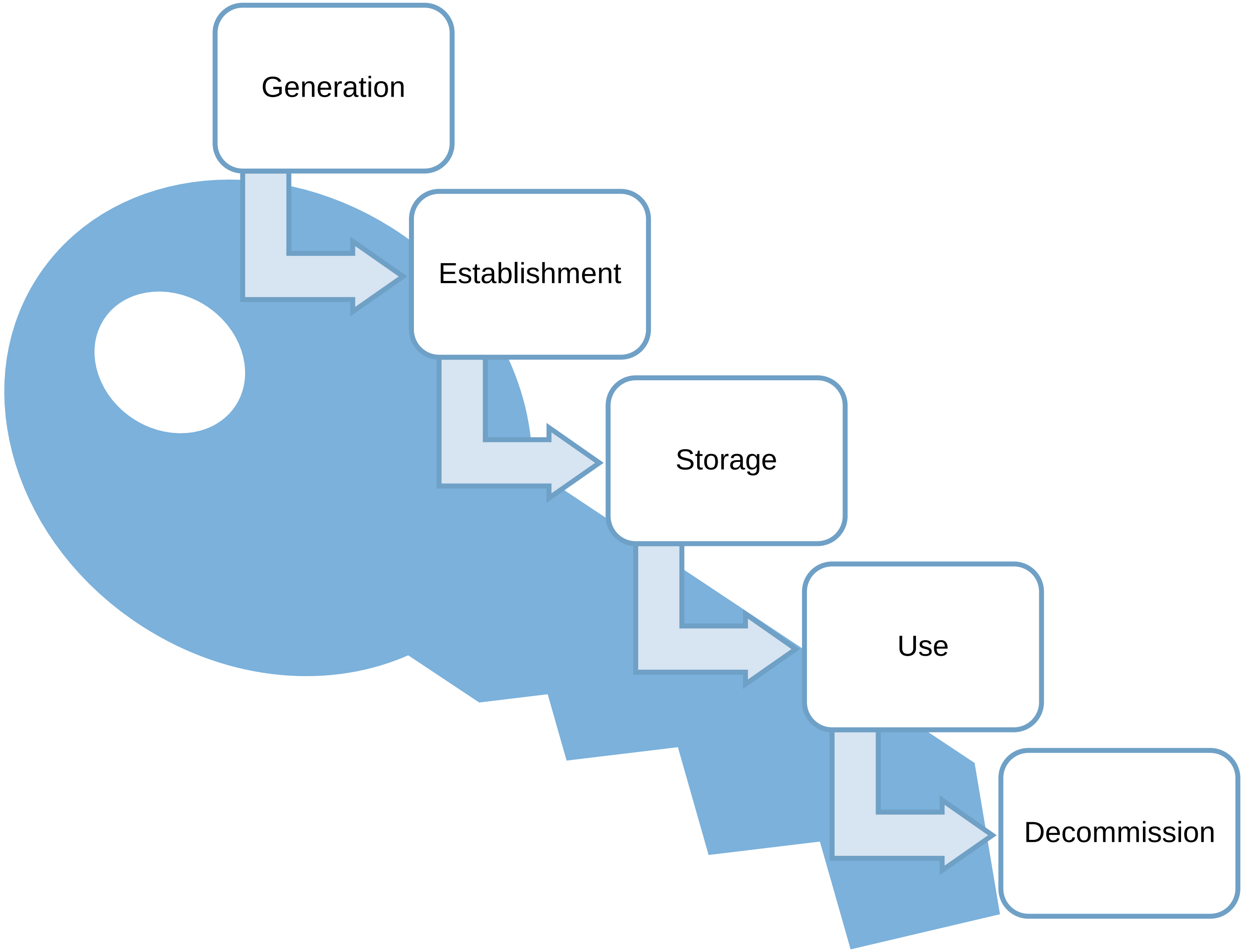
- **Cryptography is a fundamental capability needed to address edge device security**
  - Basis for protecting data at rest and in transit
  - Provides robust identity for the end device by cryptographic authentication
- **The key material used for cryptographic operations must be protected by hardware**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

★ *Requirements which depend on Cryptography*

# Protected over the lifecycle* of the Cryptographic keys

PROTECTED

- 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

# Exploring Protected Key Options

## 1 External Security IC

NXP IoT Security ICs:
A71CH
A100x Authenticator
SE050

- **Strongest protection across all key life stages**
- **Uses:**
  - Device identity and establishing TLS/onboarding
  - NXP Trust provisioning reduces overhead for key generation and establishment

Uses may overlap

## 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

Security Hardening on MCU/MPU with Software PUF (Intrinsic ID BroadKey)

- **Assist with early key life stages and improves protection for keys**
- **Uses:**
  - Key Generation and establishment
  - Device identity
  - Assist with TLS/onboarding

Uses Incremental

## 4 Security w/SRAM PUF

Hardware PUF (Intrinsic ID QuiddiKey): LPC5500 Family

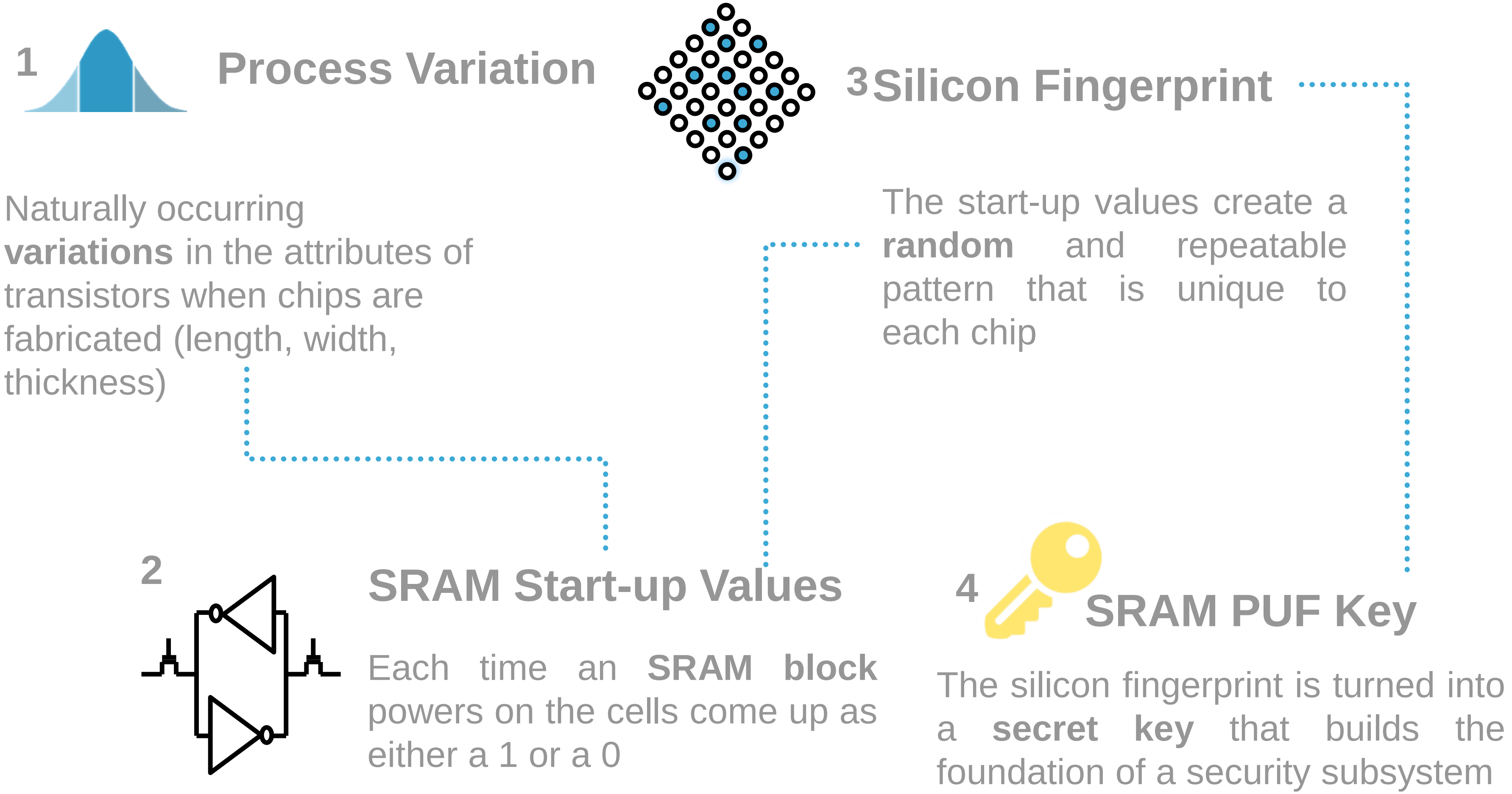
- **Links advantages of PUF to runtime application security**
- **Uses:**
  - PUF protected keys used for secure boot, etc.
  - PUF for Key generation and establishment protects early life stages

# 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**

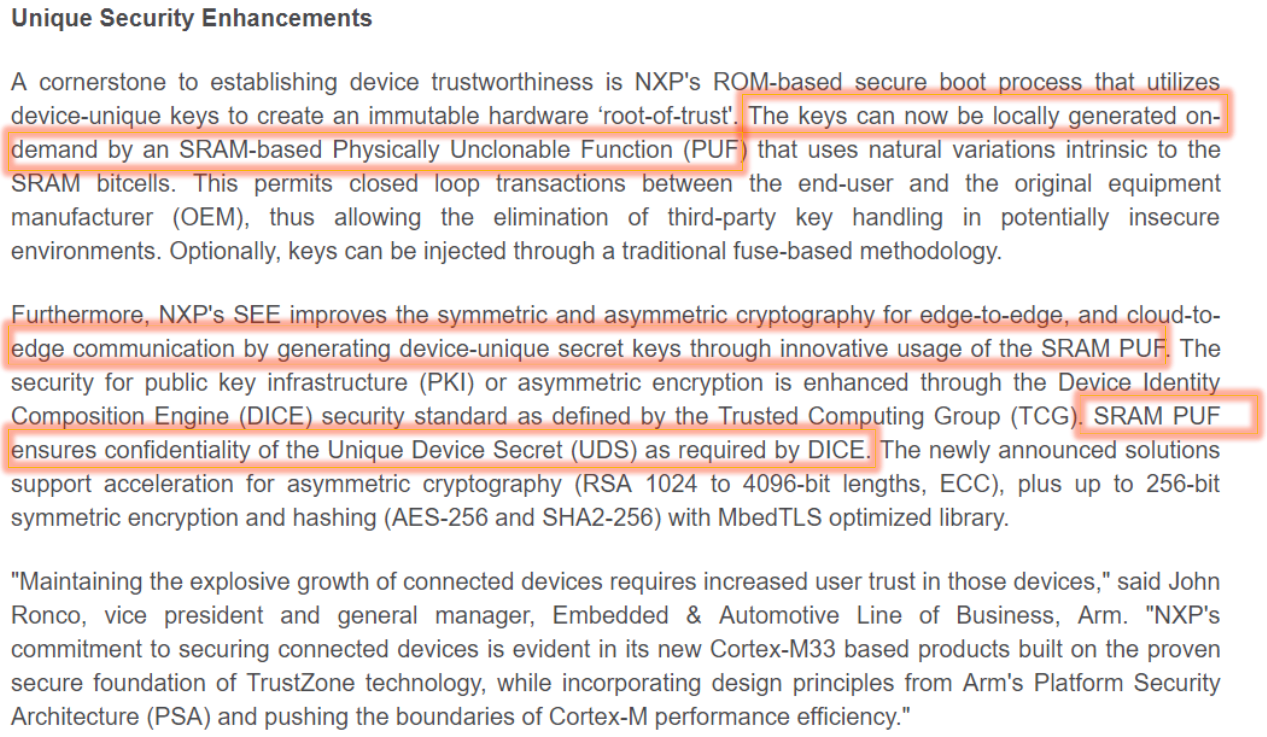
The start-up values create a **random** and repeatable pattern that is unique to each chip

4 **SRAM PUF Key**

The silicon fingerprint is turned into a **secret key** that builds the foundation of a security subsystem

# HW Protected Keys Example: Hardware PUF

Recently launched LPC5500 family also makes use of PUF technology on the microcontroller in addition to other security capabilities

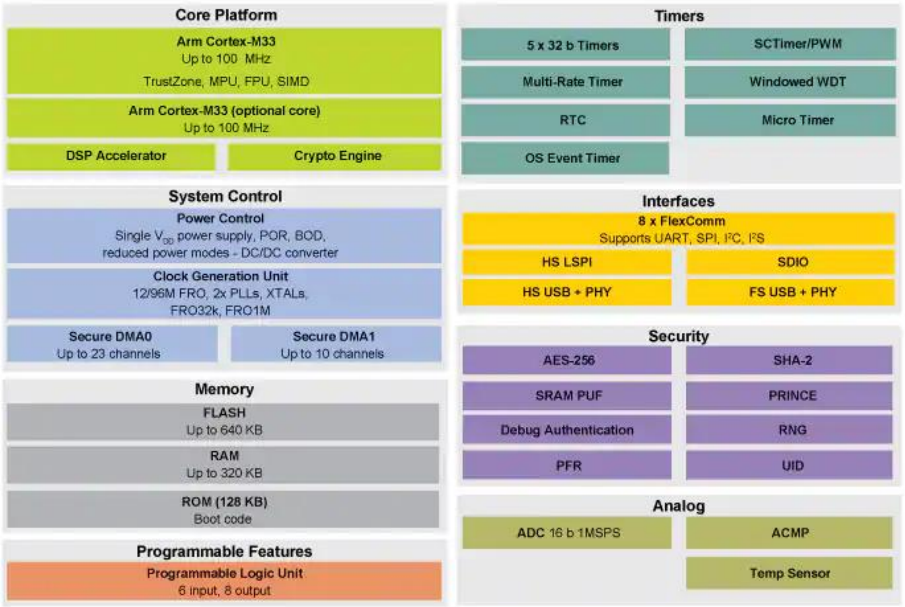


**Unique Security Enhancements**

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."



Core Platform

Arm Cortex-M33
Up to 100 MHz
TrustZone, MPU, FPU, SIMD

Arm Cortex-M33 (optional core)
Up to 100 MHz

DSP Accelerator

Crypto Engine

System Control

Power Control
Single $V_{DD}$ power supply, POR, BOD,
reduced power modes - DC/DC converter

Clock Generation Unit
12/96M FRO, 2x PLLs, XTALs,
FRO32k, FRO1M

Secure DMA0
Up to 23 channels

Secure DMA1
Up to 10 channels

Memory

FLASH
Up to 640 KB

RAM
Up to 320 KB

ROM (128 KB)
Boot code

Programmable Features

Programmable Logic Unit
6 input, 8 output

Timers

5 x 32 b Timers

SCTimer/PWM

Multi-Rate Timer

Windowed WDT

RTC

Micro Timer

OS Event Timer

Interfaces

8 x FlexComm
Supports UART, SPI, $I^2C$, $I^2S$

HS LSPI

SDIO

HS USB + PHY

FS USB + PHY

Security

AES-256

SHA-2

SRAM PUF

PRINCE

Debug Authentication

RNG

PFR

UID

Analog

ADC 16 b 1MSPS

ACMP

Temp Sensor

LPC MICROCONTROLLERS

IoT MCUs Based on Arm® Technology

LPC Cortex-M MCUs

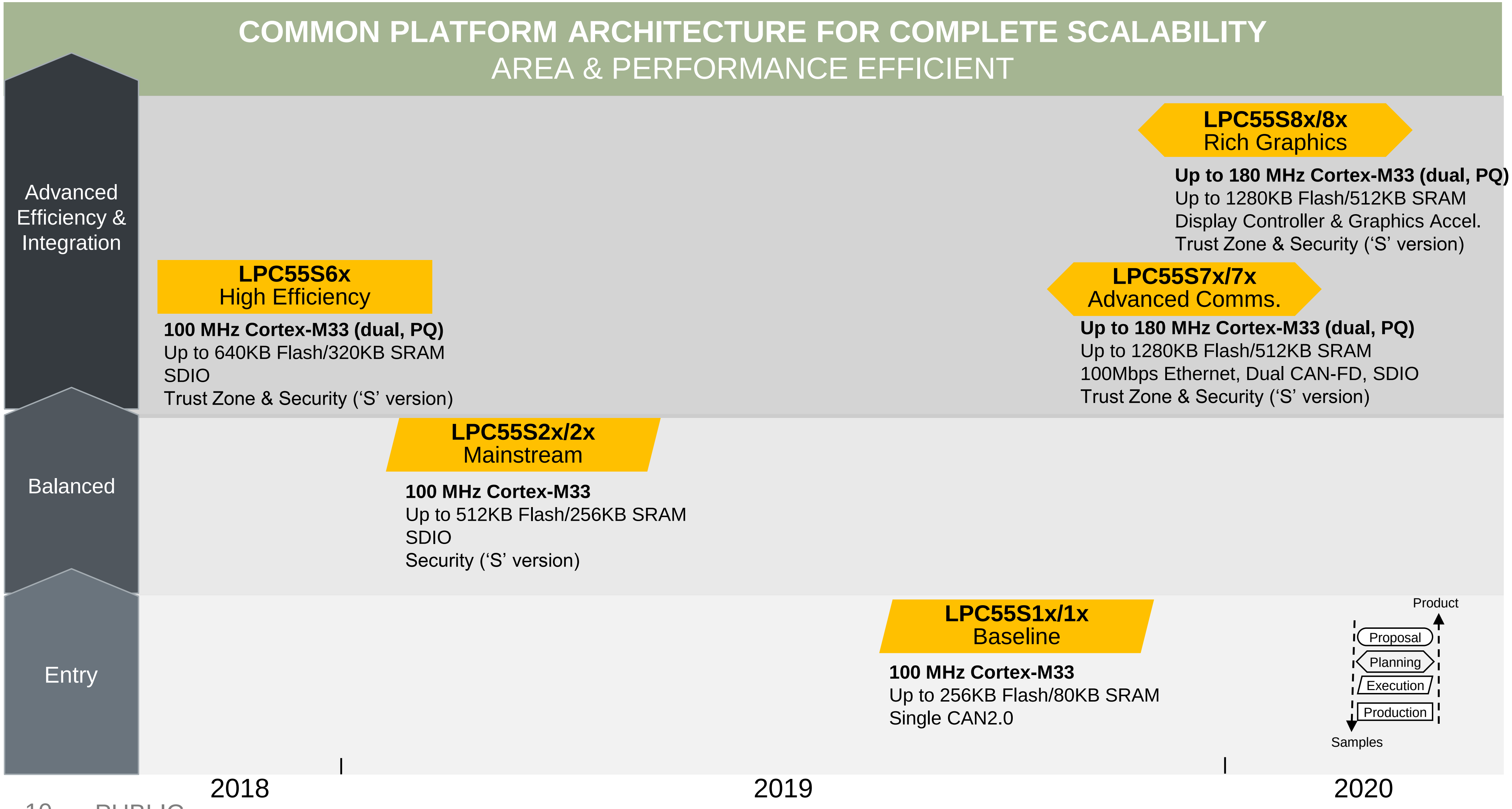
- LPC5500 Cortex-M33
- LPC54000 Cortex-M4
- LPC1100 Cortex-M0+/M0
- LPC800 Series Cortex-M0+ MCUs
- LPC1200 Cortex-M0
- LPC1300 Cortex-M3
- LPC1500 Cortex-M3
- LPC1700 Cortex-M3
- LPC1800 Cortex-M3
- LPC4000 Cortex-M4
- LPC4300 Cortex-M4/M0
- LPC2000 Arm7 MCUs
- LPC3000 Arm9 MPUs

# LPC5500 SERIES OVERVIEW

NXP LPC5500 MCU SERIES

# ROADMAP

**Subject to Change**



## COMMON PLATFORM ARCHITECTURE FOR COMPLETE SCALABILITY

AREA & PERFORMANCE EFFICIENT

Advanced Efficiency & Integration

**LPC55S8x/8x**
Rich Graphics

**Up to 180 MHz Cortex-M33 (dual, PQ)**
Up to 1280KB Flash/512KB SRAM
Display Controller & Graphics Accel.
Trust Zone & Security (‘S’ version)

**LPC55S6x**
High Efficiency

**100 MHz Cortex-M33 (dual, PQ)**
Up to 640KB Flash/320KB SRAM
SDIO
Trust Zone & Security (‘S’ version)

**LPC55S7x/7x**
Advanced Comms.

**Up to 180 MHz Cortex-M33 (dual, PQ)**
Up to 1280KB Flash/512KB SRAM
100Mbps Ethernet, Dual CAN-FD, SDIO
Trust Zone & Security (‘S’ version)

Balanced

**LPC55S2x/2x**
Mainstream

**100 MHz Cortex-M33**
Up to 512KB Flash/256KB SRAM
SDIO
Security (‘S’ version)

Entry

**LPC55S1x/1x**
Baseline

**100 MHz Cortex-M33**
Up to 256KB Flash/80KB SRAM
Single CAN2.0

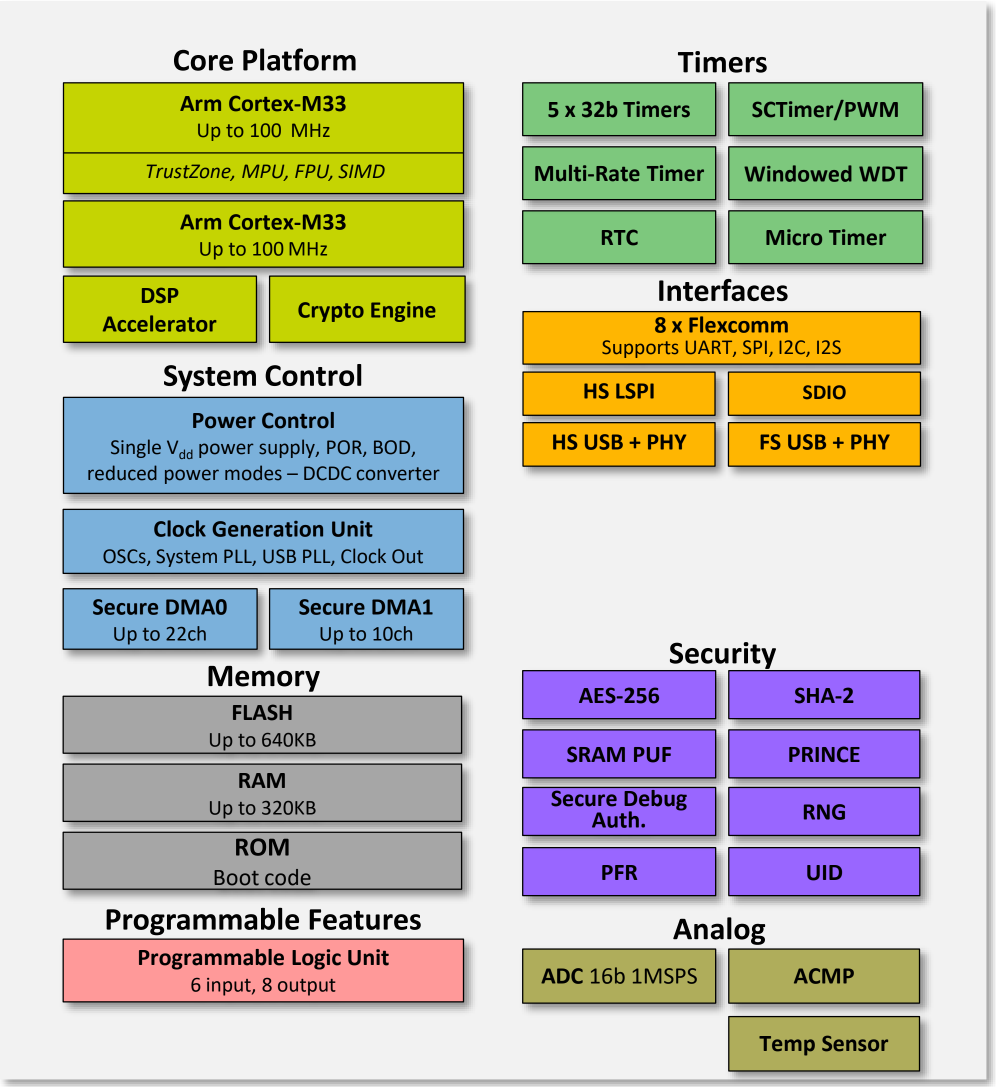
**Common features across families,**

- FS USB (wo xtal)
- HS USB with PHY*
- 50MHz SPI,
- Up to 8/10 Serial Interfaces (FlexComm)
- I3C interface (LPC557x/8x families)
- Up to 2Msps 16-bit SAR ADC
- Comparator
- Temperature Sensor & RTC
- 1.8 to 3.6V
- -40 to 105 °C

**not available in all packages*

**10Ku S/R is budgetary range; will vary for specific package/memory variants**

# LPC55S6x Product Overview



Core Platform

Arm Cortex-M33
Up to 100 MHz
*TrustZone, MPU, FPU, SIMD*

Arm Cortex-M33
Up to 100 MHz

DSP Accelerator

Crypto Engine

System Control

Power Control
Single $V_{dd}$ power supply, POR, BOD, reduced power modes – DCDC converter

Clock Generation Unit
OSCs, System PLL, USB PLL, Clock Out

Secure DMA0
Up to 22ch

Secure DMA1
Up to 10ch

Memory

FLASH
Up to 640KB

RAM
Up to 320KB

ROM
Boot code

Programmable Features

Programmable Logic Unit
6 input, 8 output

Timers

5 x 32b Timers

SCTimer/PWM

Multi-Rate Timer

Windowed WDT

RTC

Micro Timer

Interfaces

8 x Flexcomm
Supports UART, SPI, I2C, I2S

HS LSPI

SDIO

HS USB + PHY

FS USB + PHY

Security

AES-256

SHA-2

SRAM PUF

PRINCE

Secure Debug Auth.

RNG

PFR

UID

Analog

ADC 16b 1MSPS

ACMP

Temp Sensor

## Core Platform

- Up to 100MHz Cortex-M33
  - TrustZone, MPU, FPU, SIMD
- Up to 100MHz Cortex-M33
- Coprocessors
  - DSP Accelerator
  - Crypto Engine
- Multilayer Bus Matrix

## Memory

- Up to 640KB FLASH (includes PFR)
- Up to 320KB RAM
- 128KB ROM

## Timers

- 5 x 32b Timers
- SCTimer/PWM
- Multi-Rate Timer
- OS Timer
- Windowed Watchdog Timer
- RTC
- Micro Timer

## Interfaces

- USB High-speed (H/D) w/ on-chip HS PHY
- USB Full-speed (H/D), Crystal-less
- SDIO, Support 2 cards
- 1 x High-Speed SPI up to 50MHz
- 8 x Flexcomms support up to 8x SPI, 8x I2C, 8x UART, 4x $I^2S$ channels (total 8 instances)

## Advanced Security Subsystem

- Protected Flash Region (PFR)
- AES-256 HW Encryption/Decryption Engine
- SHA-2
- SRAM PUF for Key Generation support
- PRINCE – On-The-Fly Encrypt/Decrypt for flash data
- Secure debug authentication
- RNG

## Analog

- 16b ADC, 16ch, 1MSPS
- Analog Comparator
- Temperature Sensor

## Packages

- LQFP100
- VFBGA98
- LQFP64 or QFN64

## Other

- Programmable Logic Unit
- Buck DC-DC
- Operating voltage: 1.8 to 3.6V
- Temperature range: -40 to 105 °C

# NXP’s LPC5500 Product Spotlight

## Bringing Intelligence & Efficiency to the Edge

### Single & Dual-core Cortex-M33 MCU Series

- 755 CoreMarks[1] and 32uA/MHz[2] for leading performance efficiency
- 10x improvement for signal processing & cryptography
- TrustZone + Secure Execution Environment (SEE)
- Rich integration to connect and control
- MCUXpresso Ecosystem with HW & SW scalability

**1:** 2xCM33 @ 100MHz, **2:** 1xCM33 @ 100MHz

NXP LPC5500 MCU SERIES

# MCUXPRESSO SOFTWARE & TOOLS ECOSYSTEM

## Complimentary with Extensive Support

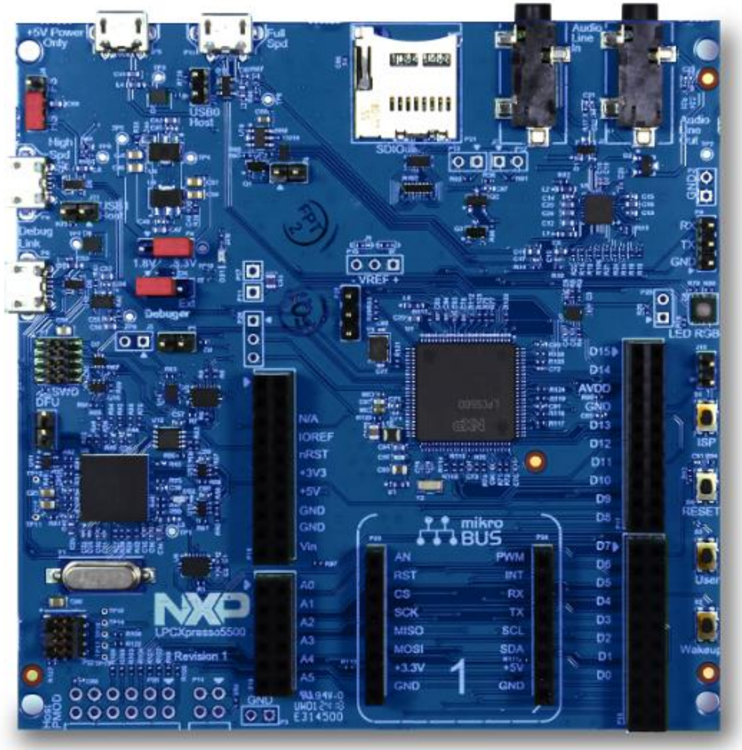
SDK MCUXpresso SDK

IDE MCUXpresso IDE

CFG MCUXpresso Config Tools

## Hardware Platform for Ease of Development

- On-board debug circuit
- PCB Layout, Schematic and Board Files Available



LPCXpresso55S69: LPC55S69-EVK

ARM KEIL Microcontroller Tools

IAR SYSTEMS

SEGGER

Simplify secure embedded development; Reduce time to market.

**LPC5500 MCU Series**

# LPC5500 Series Security Resources (as of 4/2019)

Embedded World: LPC5500 Security white paper

Arm+NXP Webinar on LPC5500

LPC55S6x Secure GPIO and Usage

Element14 Secure your Sensor with LPC5500 series

LPC55S69 Security Solutions for IoT

LPC55Sxx usage of the PUF and Hash Crypt to AES coding

LPC55Sxx Secure Boot

# SECURITY MODEL & SECURITY TECHNOLOGY

# Security Model

## Policies

The rules in place that identify the data that should be protected

For example
The management of firmware, secret keys, user and application data Passwords, personal information, network credentials

## Threat landscape

The definition of the attacks and attackers that the end device will face and protect against**.** Considers the access to the device, and cost of the attack

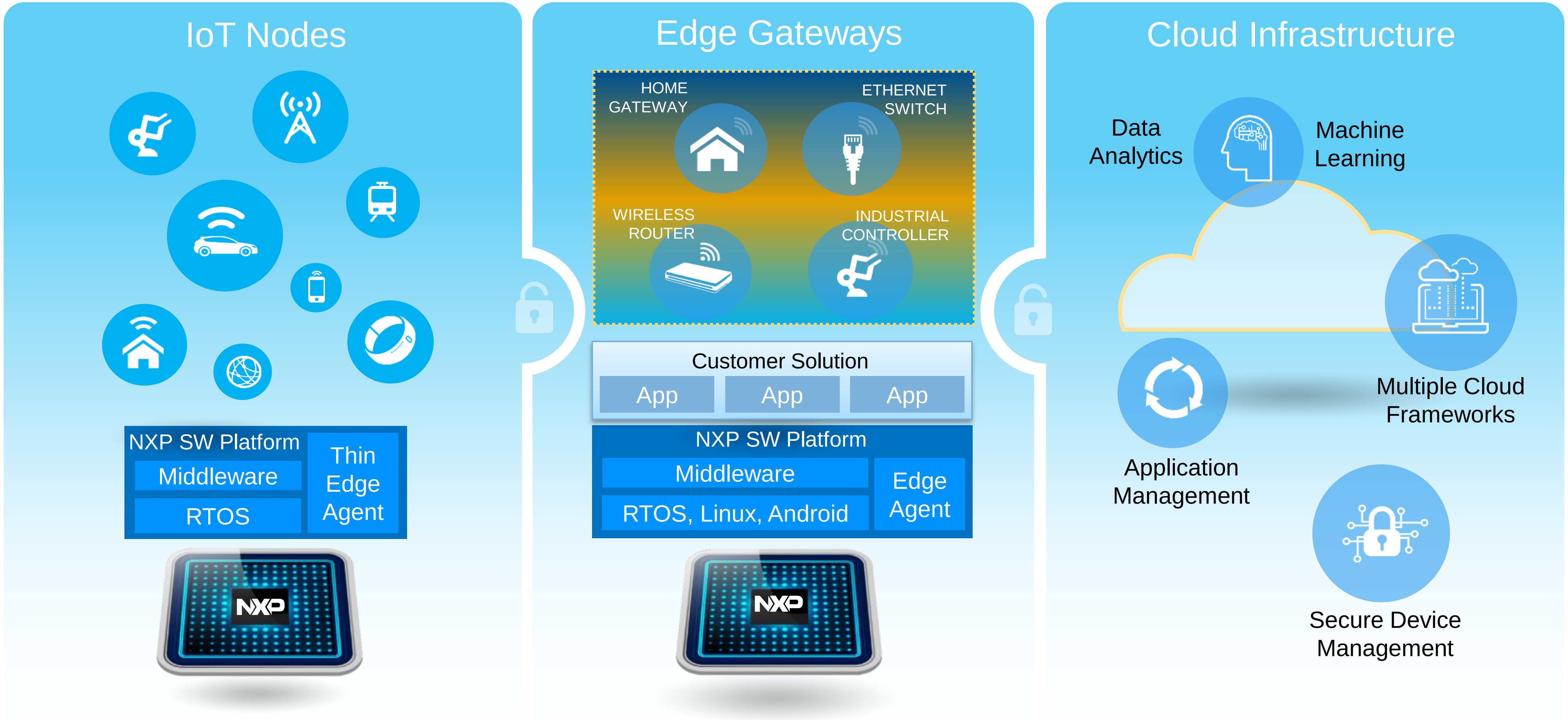
For example
Expert attackers who will use off the shelf tools to gain access and insert malware

## Methods

The means by which the policies for the device are enforced. Involves the application of security technology to achieve product goals

For example
Disabling debug access to restrict the availability of secret data on a processor

# NXP Solutions for Edge Computing



IoT Nodes

NXP SW Platform

Middleware

RTOS

Thin Edge Agent

Edge Gateways

HOME GATEWAY

ETHERNET SWITCH

WIRELESS ROUTER

INDUSTRIAL CONTROLLER

Customer Solution

App App App

NXP SW Platform

Middleware

RTOS, Linux, Android

Edge Agent

Cloud Infrastructure

Data Analytics

Machine Learning

Multiple Cloud Frameworks

Application Management

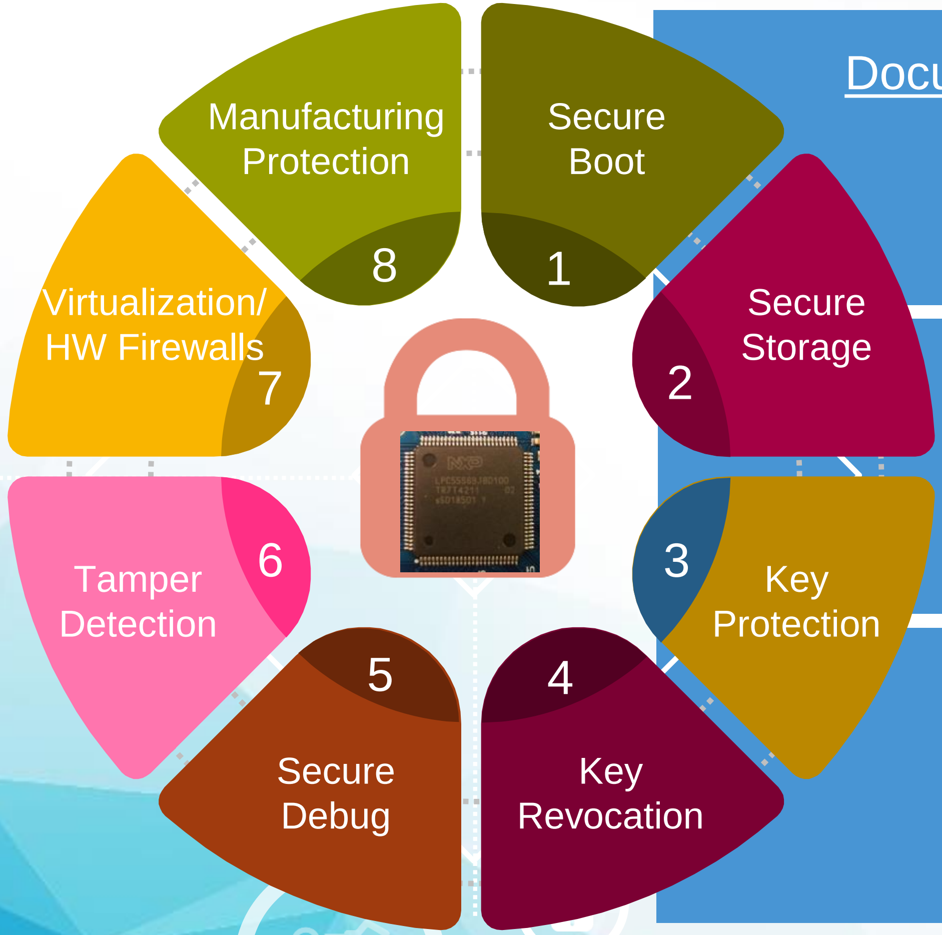
Secure Device Management

NXP Kinetis, LPC, i.MX-RT Family

NXP Layerscape, i.MX Family

NXP EdgeScale Suite

# NXP Security Technology

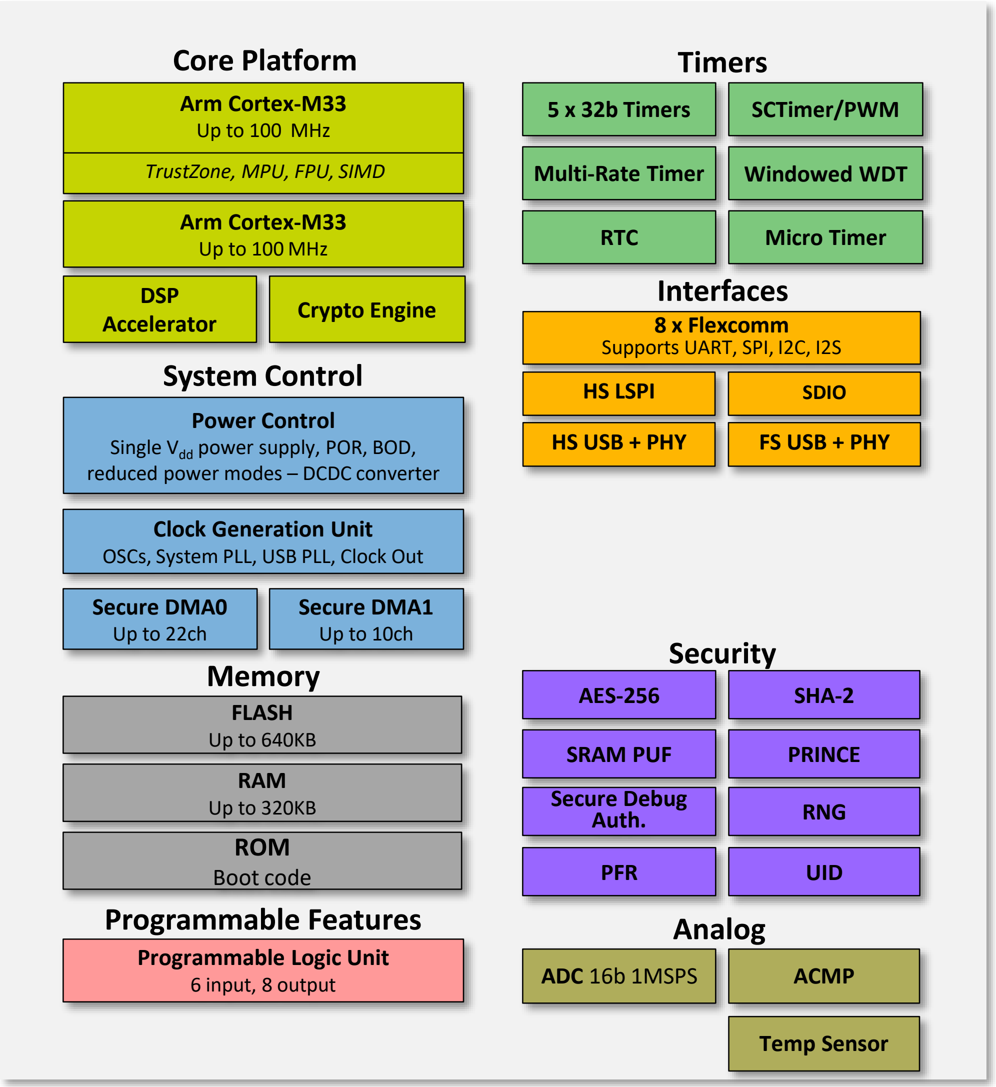


**Documentation & Certifications**
Security Users Manual
App. Notes
Community

**Software &Tools**
MCUXpresso
Code Signing Tools
Manufacturing Tools
Serial Download Tools

**Hardware**
Casper, Prince
MPC, PPC
PUF

# LPC55S6x Product Overview



Core Platform

Arm Cortex-M33
Up to 100  MHz

*TrustZone, MPU, FPU, SIMD*

Arm Cortex-M33
Up to 100 MHz

DSP Accelerator

Crypto Engine

System Control

Power Control
Single $V_{dd}$ power supply, POR, BOD, reduced power modes – DCDC converter

Clock Generation Unit
OSCs, System PLL, USB PLL, Clock Out

Secure DMA0
Up to 22ch

Secure DMA1
Up to 10ch

Memory

FLASH
Up to 640KB

RAM
Up to 320KB

ROM
Boot code

Programmable Features

Programmable Logic Unit
6 input, 8 output

Timers

5 x 32b Timers

SCTimer/PWM

Multi-Rate Timer

Windowed WDT

RTC

Micro Timer

Interfaces

8 x Flexcomm
Supports UART, SPI, I2C, I2S

HS LSPI

SDIO

HS USB + PHY

FS USB + PHY

Security

AES-256

SHA-2

SRAM PUF

PRINCE

Secure Debug Auth.

RNG

PFR

UID

Analog

ADC 16b 1MSPS

ACMP

Temp Sensor

## Core Platform

- Up to 100MHz Cortex-M33
  - TrustZone, MPU, FPU, SIMD
- Up to 100MHz Cortex-M33
- Coprocessors
  - DSP Accelerator
  - Crypto Engine
- Multilayer Bus Matrix

## Memory

- Up to 640KB FLASH (includes PFR)
- Up to 320KB RAM
- 128KB ROM

## Timers

- 5 x 32b Timers
- SCTimer/PWM
- Multi-Rate Timer
- OS Timer
- Windowed Watchdog Timer
- RTC
- Micro Timer

## Interfaces

- USB High-speed (H/D) w/ on-chip HS PHY
- USB Full-speed (H/D), Crystal-less
- SDIO, Support 2 cards
- 1 x High-Speed SPI up to 50MHz
- 8 x Flexcomms support up to 8x SPI, 8x I2C, 8x UART, 4x $I^2S$ channels (total 8 instances)

## Advanced Security Subsystem

- Protected Flash Region (PFR)
- AES-256 HW Encryption/Decryption Engine
- SHA-2
- SRAM PUF for Key Generation support
- PRINCE – On-The-Fly Encrypt/Decrypt for flash data
- Secure debug authentication
- RNG

## Analog

- 16b ADC, 16ch, 1MSPS
- Analog Comparator
- Temperature Sensor

## Packages

- LQFP100
- VFBGA98
- LQFP64 or QFN64

## Other

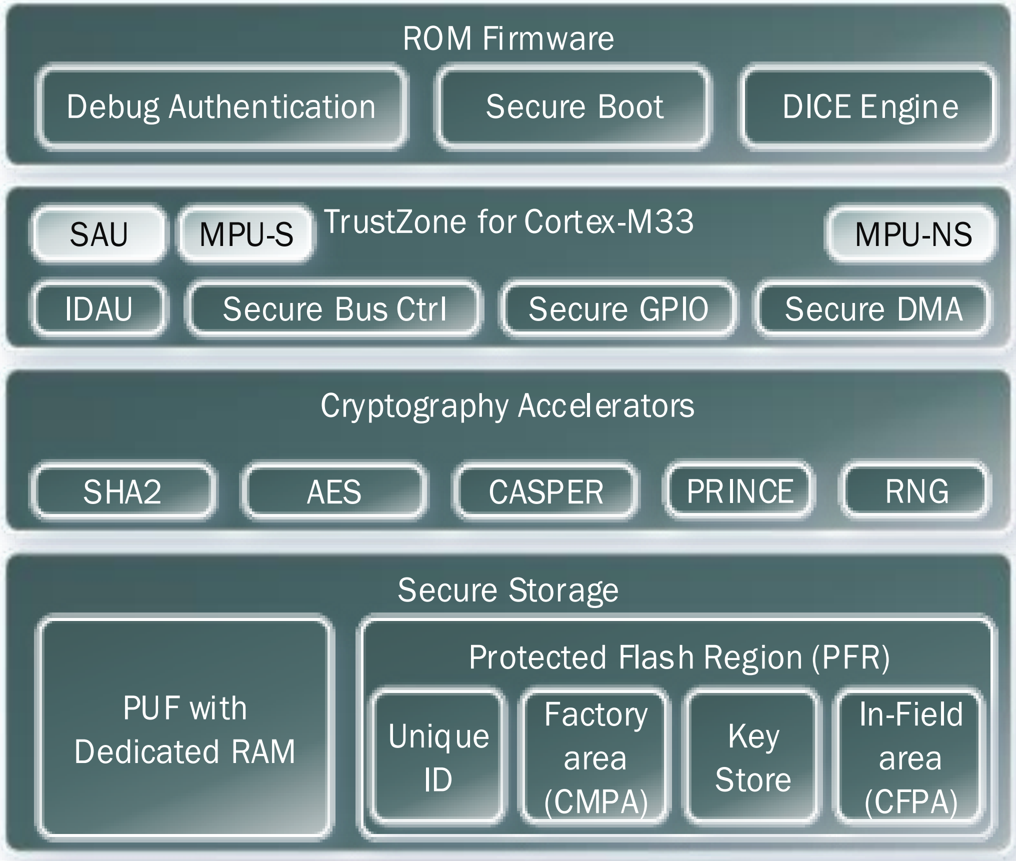
- Programmable Logic Unit
- Buck DC-DC
- Operating voltage: 1.8 to 3.6V
- Temperature range: -40 to 105 °C

# LPC55S6XX SECURITY TECHNOLOGY

NXP LPC5500 MCU SERIES

# SECURITY SUBSYSTEM OVERVIEW

- **ROM supporting**
  - Secure Boot, Debug Authentication & DICE Engine
- **TrustZone for Cortex-M33**
  - Arm's Security Attribution Unit (SAU)
  - Arm's Memory Protection Unit (MPU): Secure & Non-Secure
  - NXP's (implementation) Defined Attribution Unit (using IDAU interface)
  - NXP's Secure Bus, Secure GPIO & Secure DMA Controllers
- **Cryptography Accelerators**
  - Symmetric (AES-256) & Hashing (SHA2) engine
  - On-the-fly flash encryption/decryption engine (PRINCE)
  - Asymmetric engine for RSA and ECC (CASPER)
  - Random Number Generator (RNG)
- **Secure Storage**
  - Physically Unclonable Function (PUF)
    - Device unique root key (256 bit strength), 64-4096 bit key size
  - Protected Flash Region
    - RFC4122 compliant 128-bit UUID per device
    - Customer Manufacturing Programable Area (Boot Configuration, RoT key table hash, Debug configuration, Prince configuration)
      - PUF Key Store (Activation code, Prince region key codes, FW update key encryption key, Unique Device Secret)
    - Customer Field Programable Area (Monotonic counter, Prince IV codes)



# Virtualization/Hardware Firewalls

Protect from Software & Remote Attacks

## Challenges

- Protect from software attacks
  - Buffer overflow
  - Interrupt/Starvation
  - Malware Injection
- Meet minimum latency requirements of real time systems while crossing boundaries

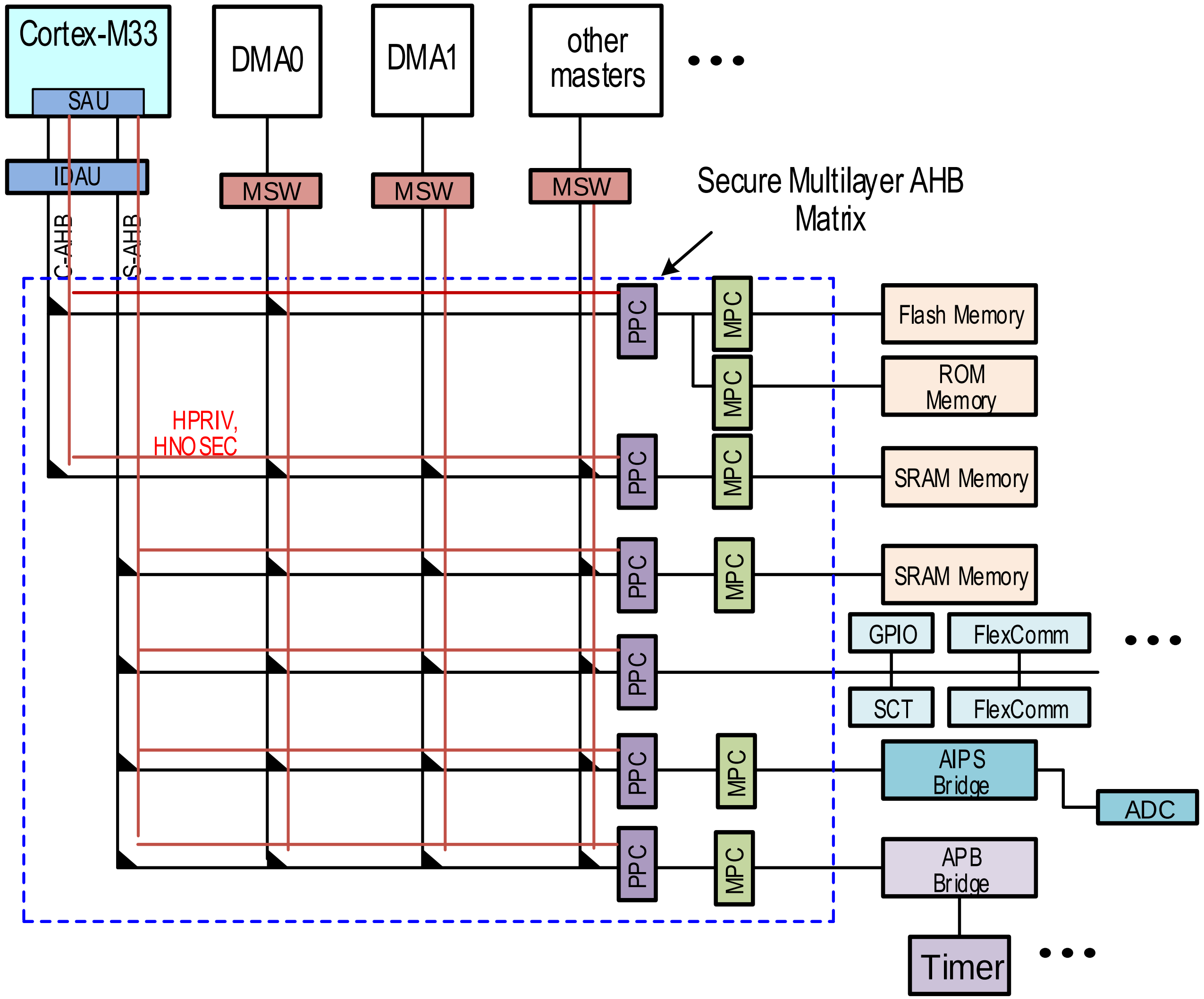
## LPC55S69 solution

- Based on Cortex-M33 with ARM's Trustzone technology
- NXP's Light weight device attribution unit to simplify setup process
- Two factor isolation protection built in AHB secure bus control with
  - Peripheral Protection Checkers
  - Memory Protection Checkers
- GPIO Masking/isolation
- Interrupt Masking/isolation
- Master Security Wrapper for other masters
- Secure configuration locking

# Virtualization/Hardware Firewalls

## Secure AHB bus matrix

- Has Security side band signals
  - HPRIV, HNONSEC
    - Pole and anti-pole version of signals used for tamper detection
- PPC per AHB slave port
  - Default security level checking
  - Provision to check both security & privilege levels
- MPCs for memories and bridge ports
  - Default security level checking
  - Provision to check both security & privilege levels
- Each master has separate security wrapper (MSW)

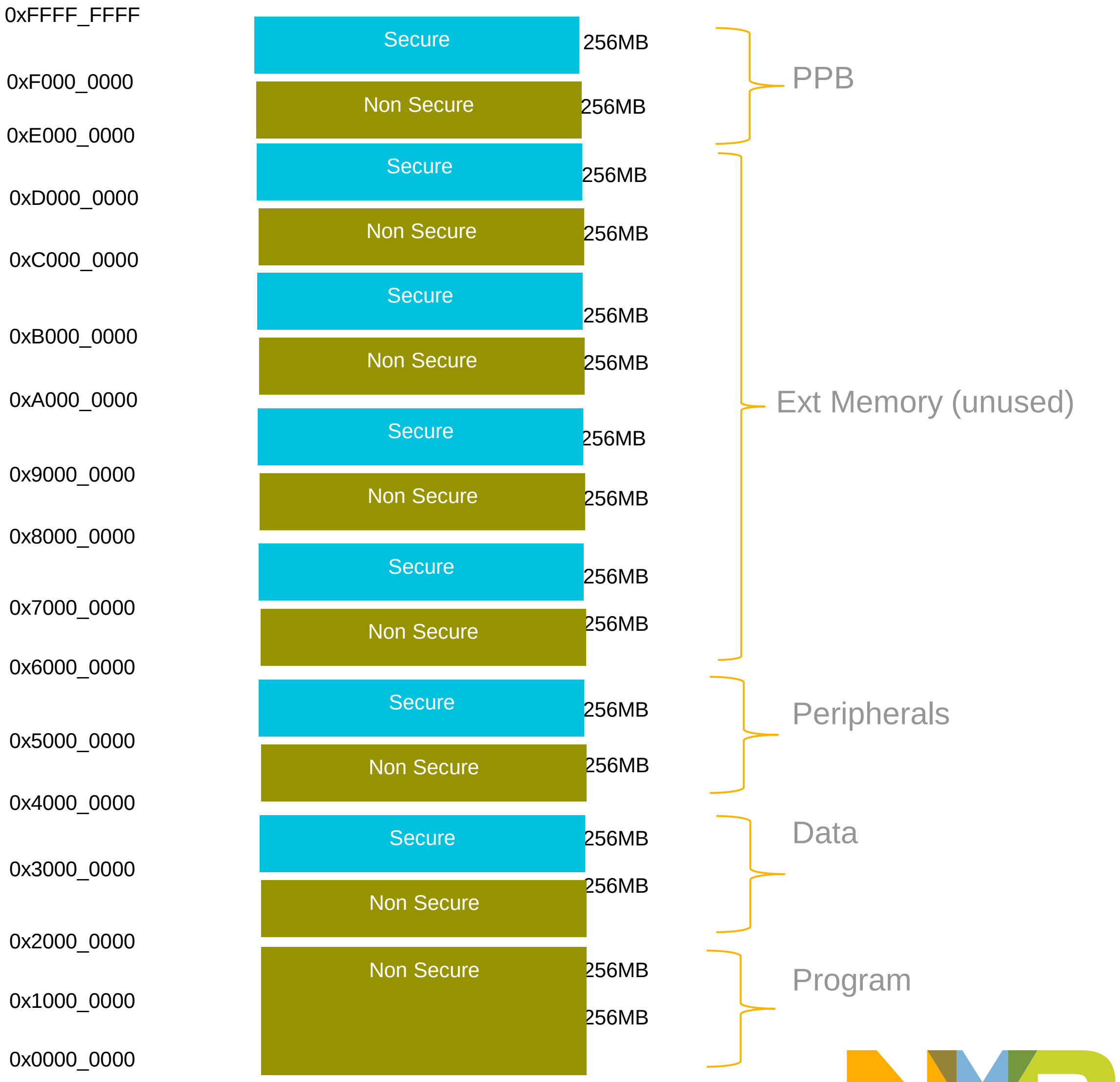


# Virtualization/Hardware Firewalls

## Memory attribution

- NXP’s Light weight device attribution unit
  - Address range 0x0000_0000 to 0x1FFF_FFFF is Non-Secure
  - Address range 0x2000_0000 to 0xFFFF_FFFF
    - If Address Bit_28 = 0  Non-Secure
    - If Address Bit_28 = 1  Secure
  - All peripherals and memories are aliased at two locations
- LPC55S69 supports 8 SAU regions

## Lightweight Device Arbitration Unit

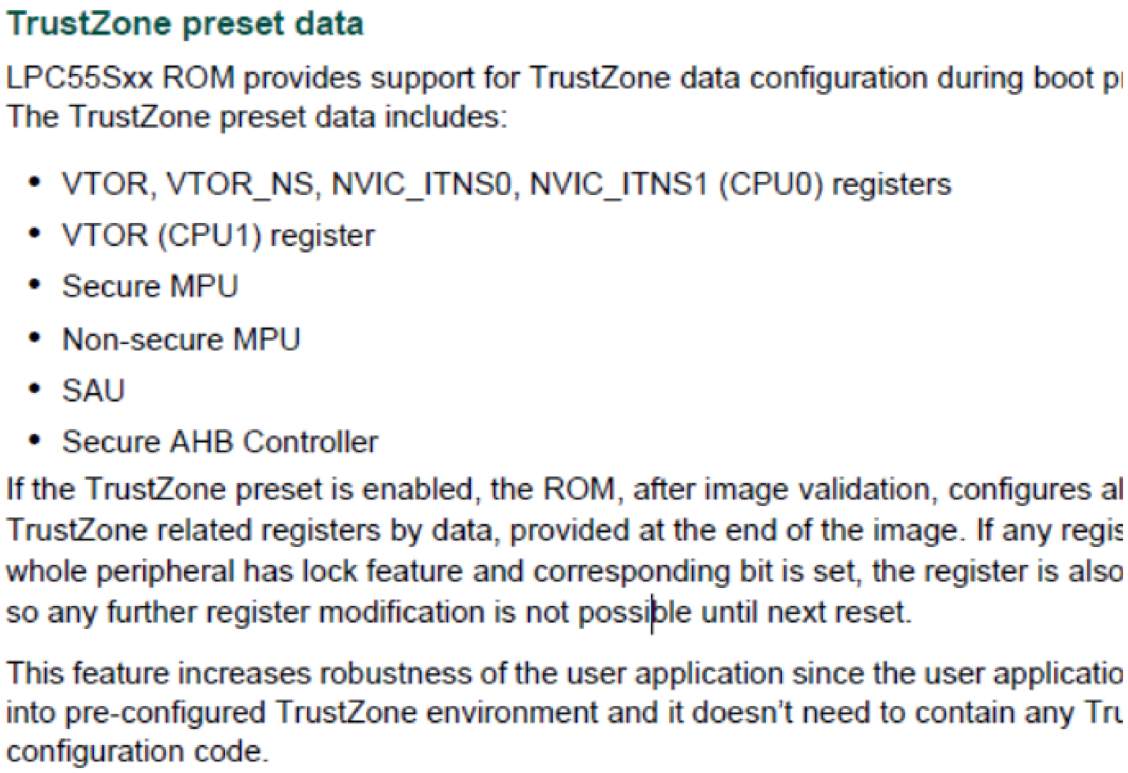


0xFFFF_FFFF
Secure 256MB
0xF000_0000
Non Secure 256MB
0xE000_0000
PPB
Secure 256MB
0xD000_0000
Non Secure 256MB
0xC000_0000
Secure 256MB
0xB000_0000
Non Secure 256MB
0xA000_0000
Secure 256MB
0x9000_0000
Non Secure 256MB
0x8000_0000
Secure 256MB
0x7000_0000
Non Secure 256MB
0x6000_0000
Ext Memory (unused)
Secure 256MB
0x5000_0000
Non Secure 256MB
0x4000_0000
Peripherals
Secure 256MB
0x3000_0000
Non Secure 256MB
0x2000_0000
Data
Non Secure 256MB 256MB
0x1000_0000
0x0000_0000
Program

# Virtualization/Hardware Firewalls

## ROM Configuration of Trustzone

- During boot process Trustzone preset data can be provisioned by the zero stage boot (ROM)
  - This ensures that before any software runs on the device, TrustZone settings are pre-loaded
  - This extends the TrustZone protections from the very start



### TrustZone preset data

LPC55Sxx ROM provides support for TrustZone data configuration during boot p
The TrustZone preset data includes:

- VTOR, VTOR_NS, NVIC_ITNS0, NVIC_ITNS1 (CPU0) registers
- VTOR (CPU1) register
- Secure MPU
- Non-secure MPU
- SAU
- Secure AHB Controller

If the TrustZone preset is enabled, the ROM, after image validation, configures al TrustZone related registers by data, provided at the end of the image. If any regis whole peripheral has lock feature and corresponding bit is set, the register is also so any further register modification is not possible until next reset.

This feature increases robustness of the user application since the user applicatio into pre-configured TrustZone environment and it doesn't need to contain any Tru configuration code.

# Virtualization/Hardware Firewalls

## Secure GPIO

- GPIO Read path is always available on a standard microcontroller
  - Secret data could be accessible from this read path

- With Secure GPIO peripheral, when SEC_GPIO_MASK is cleared, the read path from pins is blocked

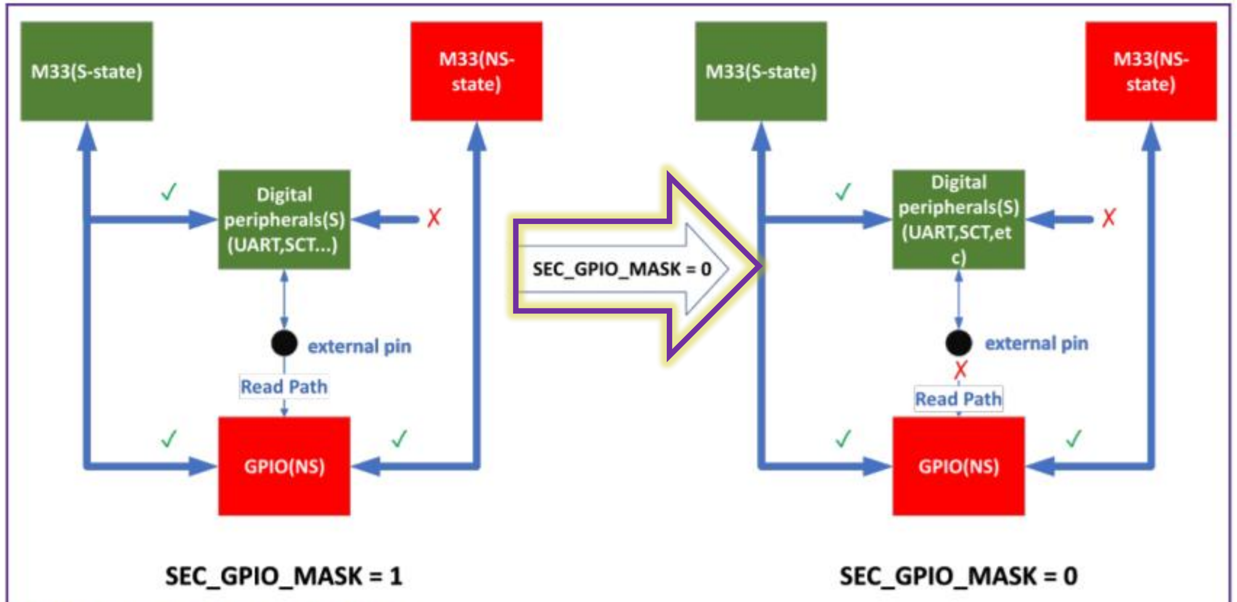


Figure 5. Usage of SEC_GPIO_MASK

# Secure debug

## Debug protection mechanism

### Challenges

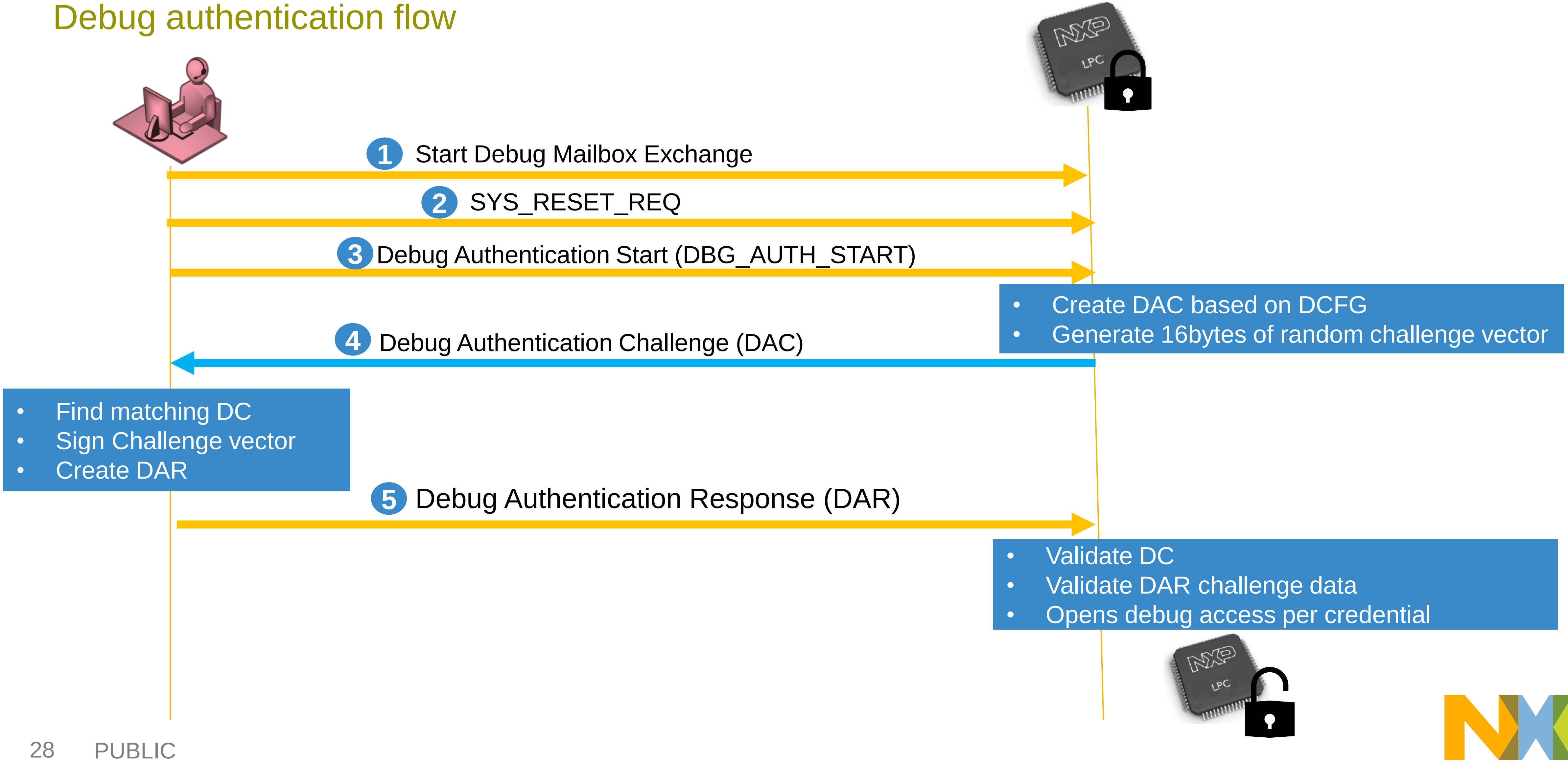
- Only authorized external entity allowed to debug
- Permit access only to allowed assets
- Support Return Material Analysis (RMA) flow without compromising security

### LPC55S69 Solution

- Supports RSA-2048/RSA-4096 signed certificate based challenge response authentication to open debug access
- Provides individual debug access control over partitioned assets
- Provides flexible security policing
  - Enforce UUID check
  - Certificate revocations
  - OEM customizable attribution check (model number, department ID etc)
- Security policy fixed at manufacturing

# Secure debug

## Debug authentication flow



1 Start Debug Mailbox Exchange

2 SYS_RESET_REQ

3 Debug Authentication Start (DBG_AUTH_START)

- Create DAC based on DCFG
- Generate 16bytes of random challenge vector

4 Debug Authentication Challenge (DAC)

- Find matching DC
- Sign Challenge vector
- Create DAR

5 Debug Authentication Response (DAR)

- Validate DC
- Validate DAR challenge data
- Opens debug access per credential

# Secure debug

## Debug protection mechanism

### Debug Credential (DC) Certificate

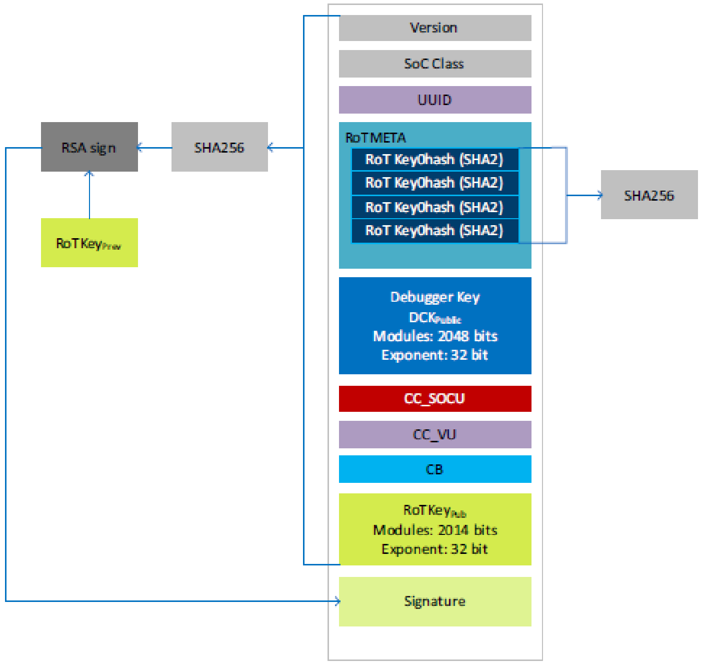

Version
SoC Class
UUID
RoTMETA
RoT Key0hash (SHA2)
RoT Key0hash (SHA2)
RoT Key0hash (SHA2)
RoT Key0hash (SHA2)
Debugger Key
$DCK_{Public}$
Modules: 2048 bits
Exponent: 32 bit
CC_SOCU
CC_VU
CB
$RoTKey_{Pub}$
Modules: 2014 bits
Exponent: 32 bit
Signature
RSA sign
SHA256
$RoTKey_{Priv}$
SHA256

**Fig 190. Debug Credential certificate fields**

### PKI for Secure boot and Debug

- Same Root of Trust Private keys are used to create the DC signature
- Options for HW and SW constraints
  - Device Unique ID bound
  - Level of Debug access
  - Mass erase enable

# Secure Debug

## LPC55Sxx Debug Domains – SoC Credential Constraints

### DC HW Credential Constraints

NIDEN - Non-secure non-invasive debug.

DBGEN - Non-secure invasive debug

SPNIDEN - Secure non-invasive debug

SPIDEN - Secure invasive debug

TAPEN - TAP (Test Access Point) controller

uDBGEN - Micro-CM33 invasive debug

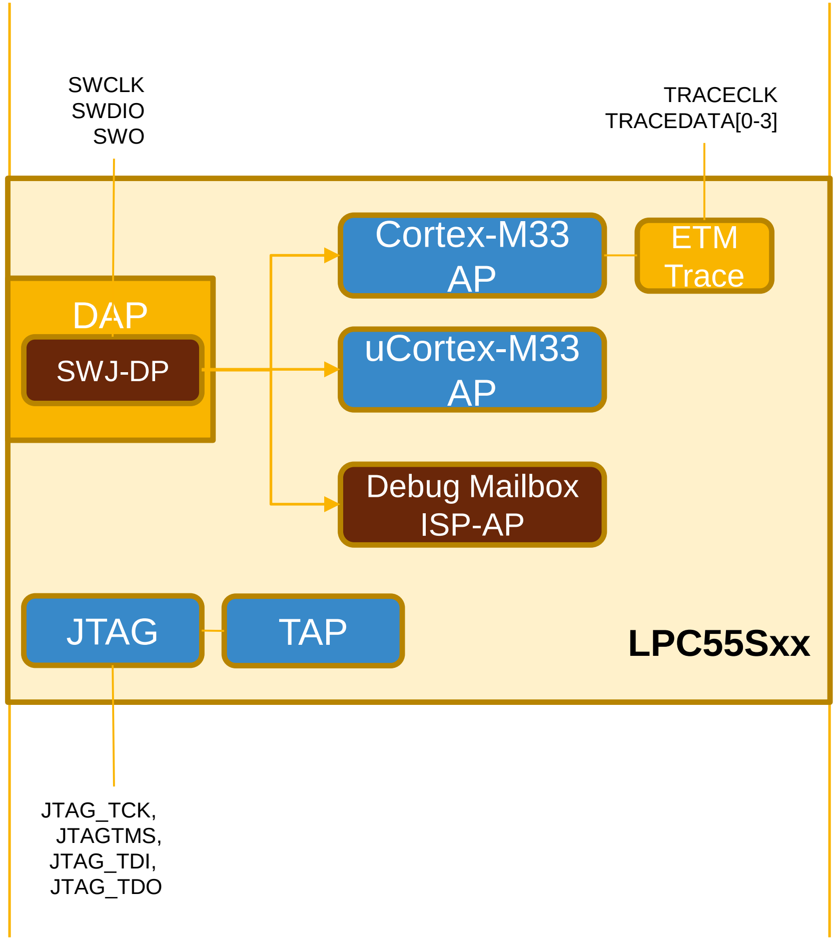
uNIDEN - Micro-CM33 non-invasive debug

### DC SW Credential Constraints

ISPEN - ISP boot command

FAEN - Field Return Analysis mode command

MEEN- Flash mass erase command



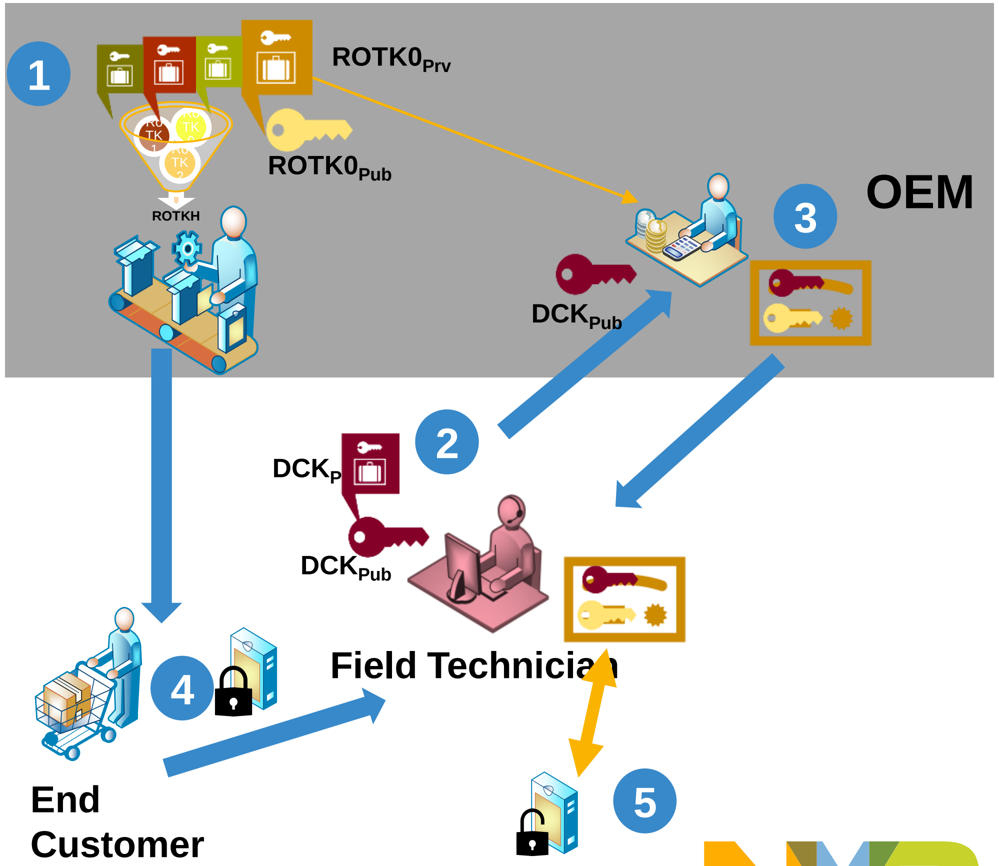
### Configuration Control

- Fields in Customer Programed Protect Flash Region provide control of the sub-domains
  - Disabled permanently
  - Enabled after debug authentication
  - Enabled permanently
- Other controls
  - Enforce UUID checking
  - Revoke debug keys

# Secure debug

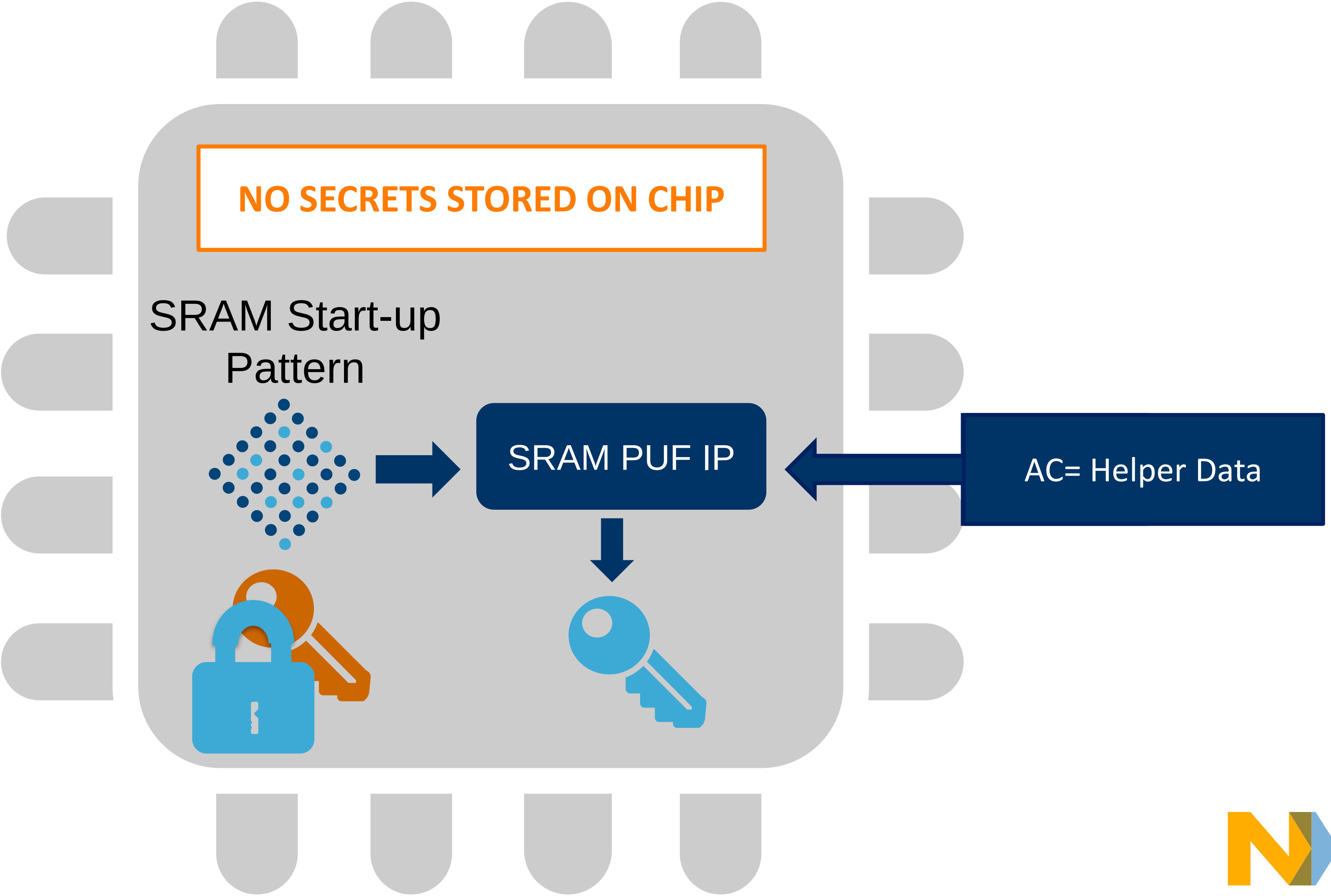
## Debug authentication for RMA use case

1. OEM generates RoT key pairs and programs the device before shipping.
   - SHA256 hash of RoT public key hashes
2. Field Technician generates his own key pair and provides public key to OEM for authorization.
3. OEM attests the Field Technician’s public key. In the debug credential certificate he assigns the access rights.
4. End customer having issues with a locked product takes it to Field technician.
5. Field technician uses his credentials to authenticate with device and un-locks the product for debugging.

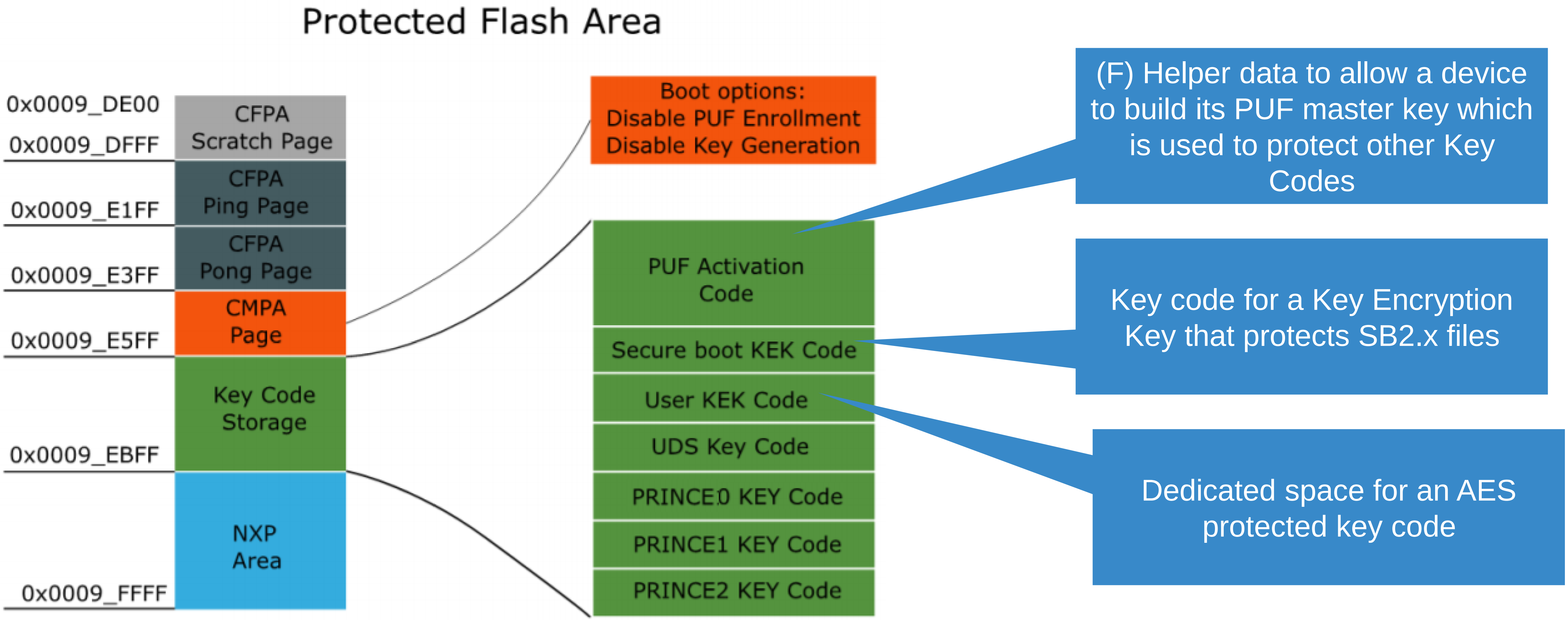


# PUF BASED KEY MANAGEMENT

# Using PUF Technology



# PUF based Key Management on LPC5500 Series



Protected Flash Area

0x0009_DE00
0x0009_DFFF
CFPA Scratch Page

0x0009_E1FF
CFPA Ping Page

0x0009_E3FF
CFPA Pong Page

0x0009_E5FF
CMPA Page

0x0009_EBFF
Key Code Storage

0x0009_FFFF
NXP Area

Boot options:
Disable PUF Enrollment
Disable Key Generation

PUF Activation Code

Secure boot KEK Code

User KEK Code

UDS Key Code

PRINCE0 KEY Code

PRINCE1 KEY Code

PRINCE2 KEY Code

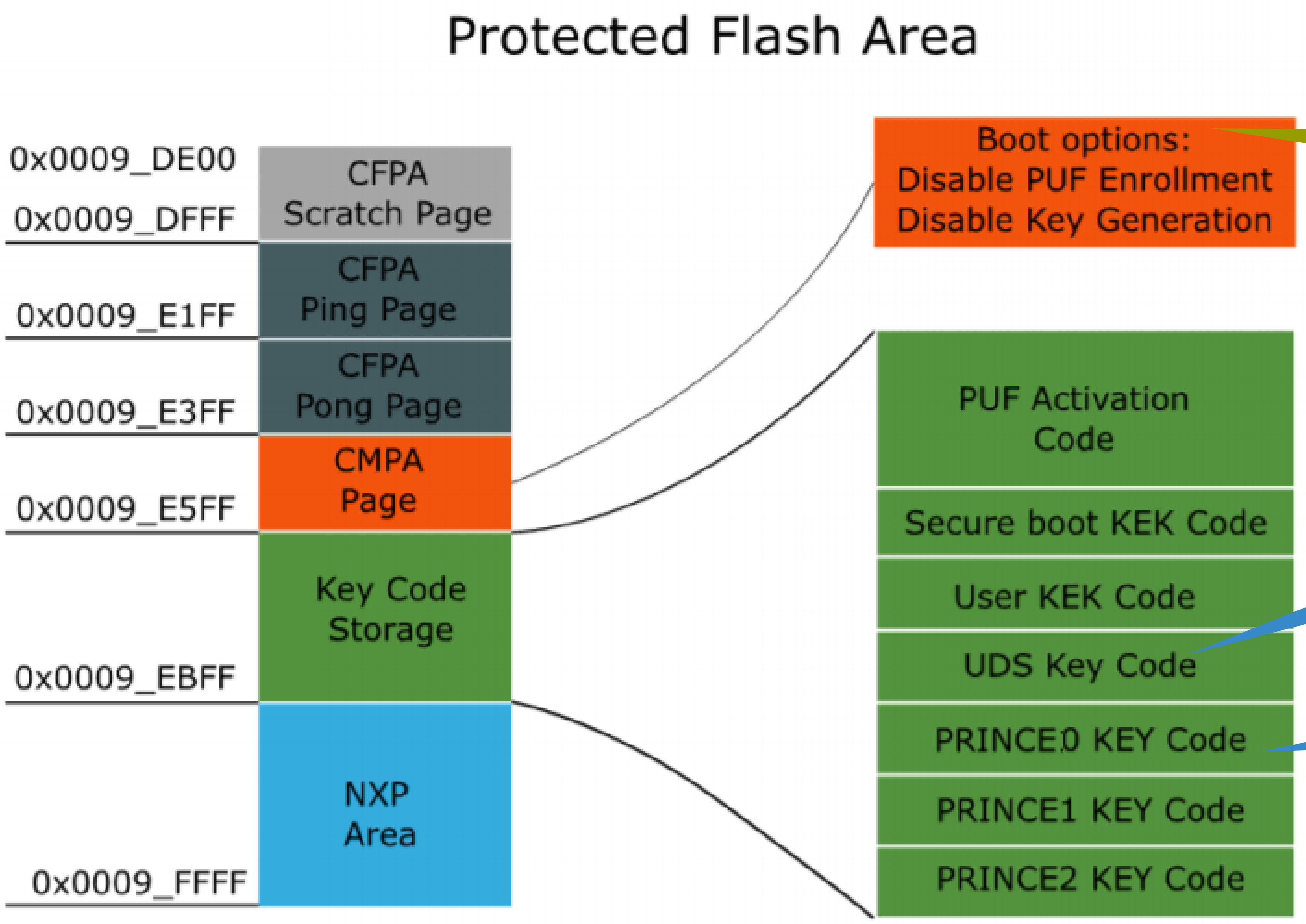
(F) Helper data to allow a device to build its PUF master key which is used to protect other Key Codes

Key code for a Key Encryption Key that protects SB2.x files

Dedicated space for an AES protected key code

CFPA Customer Field Programmable Area
CMPA Customer Manufacturing floor Programmable Area

# PUF based Key Management on LPC5500 Series



Protected Flash Area

0x0009_DE00
0x0009_DFFF
CFPA Scratch Page

0x0009_E1FF
CFPA Ping Page

0x0009_E3FF
CFPA Pong Page

0x0009_E5FF
CMPA Page

0x0009_EBFF
Key Code Storage

0x0009_FFFF
NXP Area

Boot options:
Disable PUF Enrollment
Disable Key Generation

PUF Activation Code

Secure boot KEK Code

User KEK Code

UDS Key Code

PRINCE0 KEY Code

PRINCE1 KEY Code

PRINCE2 KEY Code

(L) Lifecycle management of PUF, block future cryptographic context generation for the device

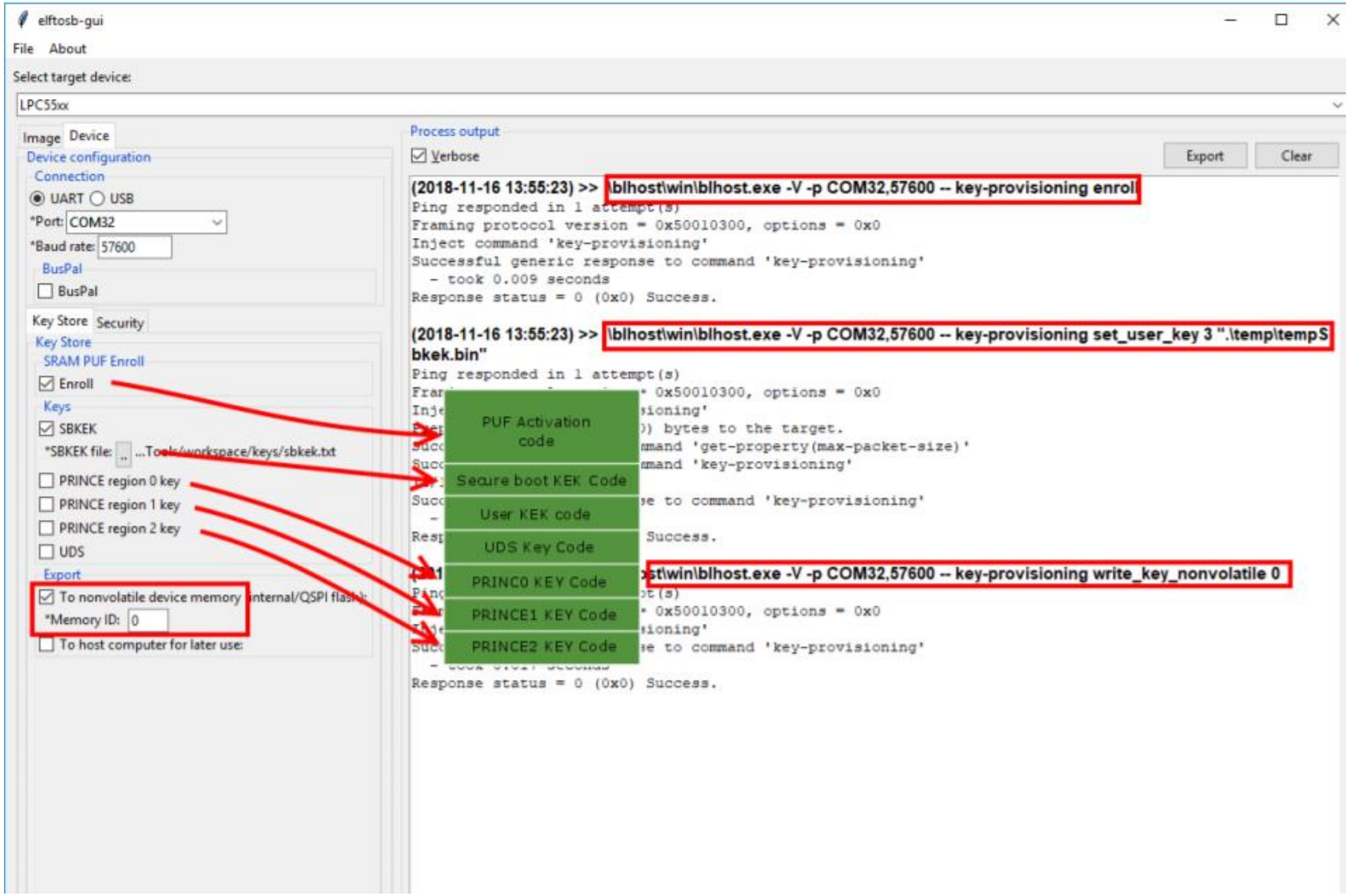
DICE related key code: Trusted Computing Group documentation

Dedicated space for PRINCE region key codes (128bit)

CFPA Customer Field Programmable Area

CMPA Customer Manufacturing floor Programmable Area

# Command line or GUI options for PUF provisioning



elftosb-gui

File  About

Select target device: LPC55xx

Image | Device

Device configuration

Connection: ● UART ○ USB

*Port: COM32

*Baud rate: 57600

BusPal: □ BusPal

Key Store | Security

Key Store

SRAM PUF Enroll: ☑ Enroll

Keys

☑ SBKEK

*SBKEK file: ... ...Tools/workspace/keys/sbkek.txt

□ PRINCE region 0 key

□ PRINCE region 1 key

□ PRINCE region 2 key

□ UDS

Export

☑ To nonvolatile device memory (internal/QSPI flash):

*Memory ID: 0

□ To host computer for later use:

Process output

☑ Verbose

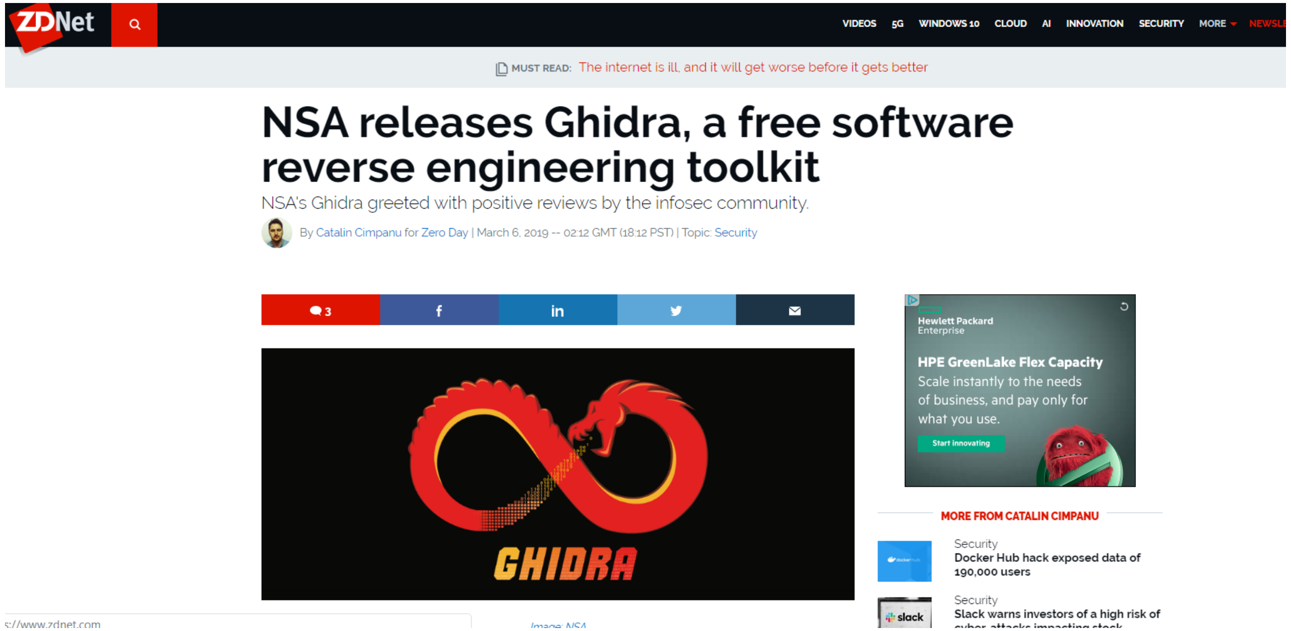
```
(2018-11-16 13:55:23) >> \blhost\win\blhost.exe -V -p COM32,57600 -- key-provisioning enroll
Ping responded in 1 attempt(s)
Framing protocol version = 0x50010300, options = 0x0
Inject command 'key-provisioning'
Successful generic response to command 'key-provisioning'
  - took 0.009 seconds
Response status = 0 (0x0) Success.

(2018-11-16 13:55:23) >> \blhost\win\blhost.exe -V -p COM32,57600 -- key-provisioning set_user_key 3 ".\temp\tempS
bkek.bin"
Ping responded in 1 attempt(s)
Fra                   = 0x50010300, options = 0x0
Inje                  sioning'
                      )) bytes to the target.
Succ                  mmand 'get-property(max-packet-size)'
Succ                  mmand 'key-provisioning'

Succ                  se to command 'key-provisioning'
  -
Resp                  Success.

(201                  st\win\blhost.exe -V -p COM32,57600 -- key-provisioning write_key_nonvolatile 0
Ping                  pt(s)
                      = 0x50010300, options = 0x0
                      sioning'
Succ                  se to command 'key-provisioning'

Response status = 0 (0x0) Success.
```

Scalable methods for instantiating device unique keys which are protected by PUF technology

"*Using A1 silicon we are working on enabling support for Untrusted-CM manufacturing. When that happens device unique key store( PUF activation code, Prince keys and device key with NXP certificate) is pre-programmed in PFR.* "

# Challenge: Asset Protection

- On-chip non-volatile storage is used for storing important assets
  - Secret keys
  - Proprietary SW from OEM and Silicon Manufacturer
  - Application code
  - Other sensitive information
- Prone to attacks with malicious intent
  - Reading the code for cloning
  - Tampering for
    - Illegally gaining trust
    - Changing execution sequence
    - Changing programming value
  - Stealing keys
- Solution:
  - Encrypt the code stored in Flash
    - System performance cannot be compromised



MUST READ: The internet is ill, and it will get worse before it gets better

NSA releases Ghidra, a free software reverse engineering toolkit

NSA's Ghidra greeted with positive reviews by the infosec community.

By Catalin Cimpanu for Zero Day | March 6, 2019 -- 02:12 GMT (18:12 PST) | Topic: Security

MORE FROM CATALIN CIMPANU

Security
Docker Hub hack exposed data of 190,000 users

Security
Slack warns investors of a high risk of cyber attacks impacting stock

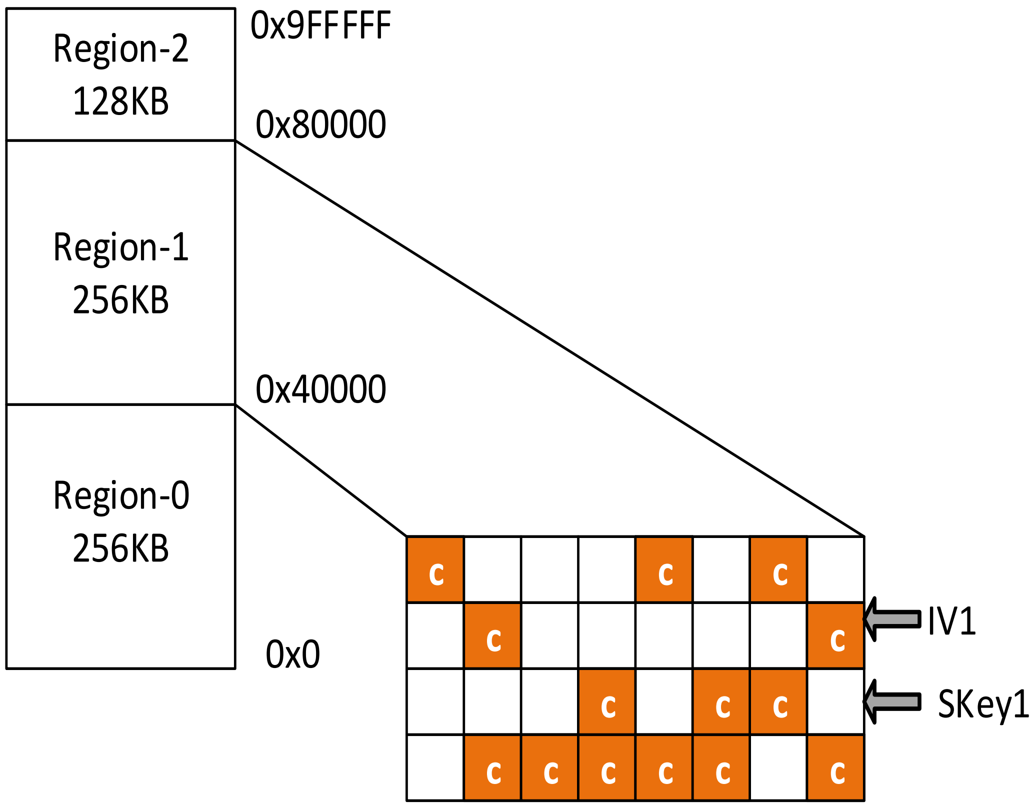
s://www.zdnet.com

Image: NSA

# PRINCE for encrypted execution

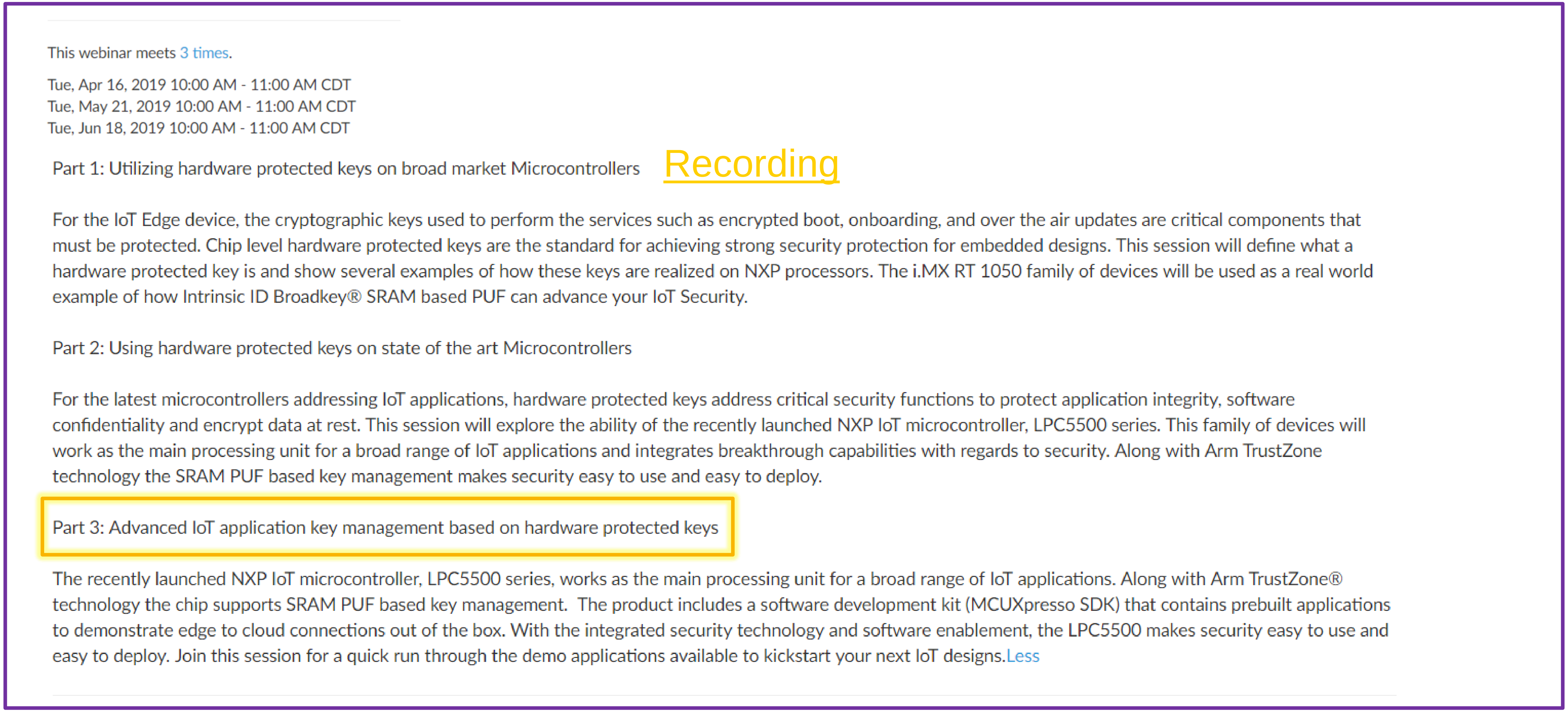
- Is a cryptographic algorithm developed by NXP + 2 Universities
  - https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2012/529&version=20140612:115014&file=529.pdf
- A light-weight symmetric block cryptography algorithm
  - 64b block cipher, with 128b crypto key
  - Same HW block supports encrypt and decrypt
- Real-time
  - Low latency decryption, no additional cycles added to read path (compared to 10-14 cycles in AES)
  - No initialization time
  - Combinatorial logic
- Efficient
  - Low cost (Si area)
  - Power efficient
  - No RAM buffers needed

# LPC55Sxx Encrypted Flash Regions



- Data stored in Flash is encrypted version
- Supports 3 regions in 640KB Flash
  - Each region is be at 256KB Address boundary
  - Allows multiple code images from independent source to co-exist
  - Secret-Key and IV Pair per region
- Register programmable crypto-enable bit per sub-region
  - One register per region
  - Each sub-region has 8kB granularity
  - Settings can be stored in PFR and be applied by ROM
- Cached data in FMC (cache) is obscured further using XOR mask with random number

# Hardware Protected Keys Webinar Series



This webinar meets 3 times.

Tue, Apr 16, 2019 10:00 AM - 11:00 AM CDT
Tue, May 21, 2019 10:00 AM - 11:00 AM CDT
Tue, Jun 18, 2019 10:00 AM - 11:00 AM CDT

Part 1: Utilizing hardware protected keys on broad market Microcontrollers Recording

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. 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.

Part 2: Using hardware protected keys on state of the art Microcontrollers

For the latest microcontrollers addressing IoT applications, hardware protected keys address critical security functions to protect application integrity, software confidentiality and encrypt data at rest. This session will explore the ability of the recently launched NXP IoT microcontroller, LPC5500 series. This family of devices will work as the main processing unit for a broad range of IoT applications and integrates breakthrough capabilities with regards to security. Along with Arm TrustZone technology the SRAM PUF based key management makes security easy to use and easy to deploy.

Part 3: Advanced IoT application key management based on hardware protected keys

The recently launched NXP IoT microcontroller, LPC5500 series, works as the main processing unit for a broad range of IoT applications. Along with Arm TrustZone® technology the chip supports SRAM PUF based key management. The product includes a software development kit (MCUXpresso SDK) that contains prebuilt applications to demonstrate edge to cloud connections out of the box. With the integrated security technology and software enablement, the LPC5500 makes security easy to use and easy to deploy. Join this session for a quick run through the demo applications available to kickstart your next IoT designs.Less

# CONCLUSION

# Summary

- LPC55S69 provides rich peripheral interfaces and security features needed for todays IoT applications.
  - PUF based key protection
  - ROM enabled key management
- Arm Trustzone for cortex-M enhances protection from scalable remote attacks
  - Enforced for the CPU and the SoC microarchitecture (i.e. Secure GPIO)
- Secure Debug capabilities address the usability of security enabled systems
  - Enabled by ROM
  - Using the same PKI as Secure Boot

**Thanks!**

# Summary

- LPC55S69 provides rich peripheral interfaces and security features needed for todays IoT applications.
  - PUF based key protection
  - ROM enabled key management
- Arm Trustzone for cortex-M enhances protection from scalable remote attacks
  - Enforced for the CPU and the SoC microarchitecture (i.e. Secure GPIO)
- Secure Debug capabilities address the usability of security enabled systems
  - Enabled by ROM
  - Using the same PKI as Secure Boot

**Questions & Answers Session**

SECURE CONNECTIONS
FOR A SMARTER WORLD