NXP | FTF 2016 TECHNOLOGY FORUM

SECURING THE IoT ONE EDGE NODE AT A TIME WITH KINETIS

FTF-DES-N1952

MELISSA HUNTER, KINETIS SYSTEMS ENG
DONNIE GARCIA, SYSTEMS & APPLICATIONS ENG
FTF-DES-N1952
MAY 18, 2016

PUBLIC USE

AGENDA

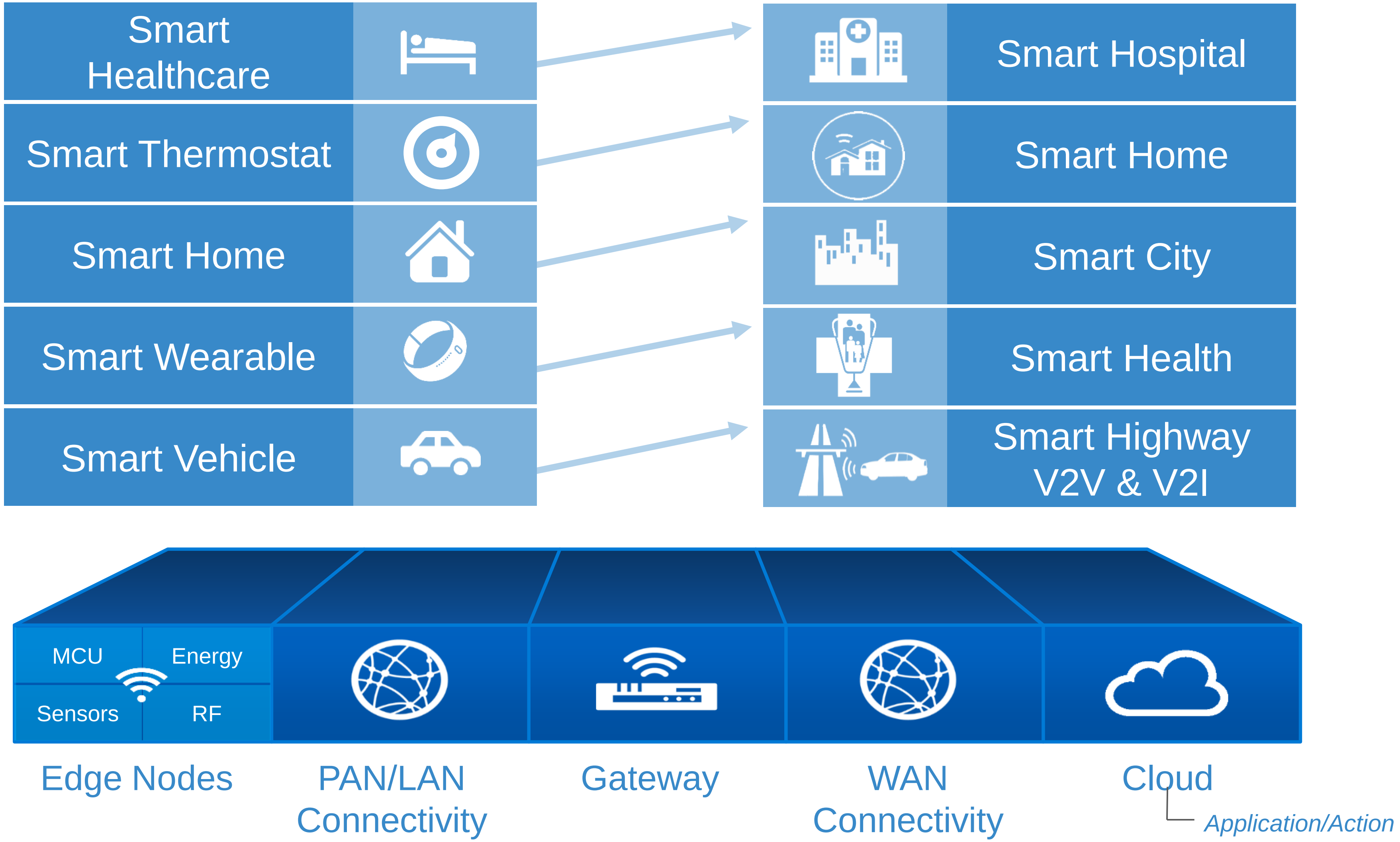
- IOT Security Needs
- Crypto
  - Kinetis HW Crypto Acceleration Features
  - Lab: Crypto Throughput
- Anti-Tamper
- Trust
  - Kinetis Trust Features
  - Serial NOR Flash On-the-Fly Decryption
    - Lab: QSPI OTFAD
- Enablement
- Summary

IOT SECURITY NEEDS

Comparing Security & Taxes

- Complicated codes rules and regulations
- Penalties for non-compliance
- Differences across world regions
- The more you make the more you have to pay
- Social benefits for paying
- Dependence on self accountability
- Easiest when done with a guide

Internet of Tomorrow → Smart, Connected and Secure



Smart Healthcare

Smart Thermostat

Smart Home

Smart Wearable

Smart Vehicle

Smart Hospital

Smart Home

Smart City

Smart Health

Smart Highway V2V & V2I

Edge Nodes

PAN/LAN Connectivity

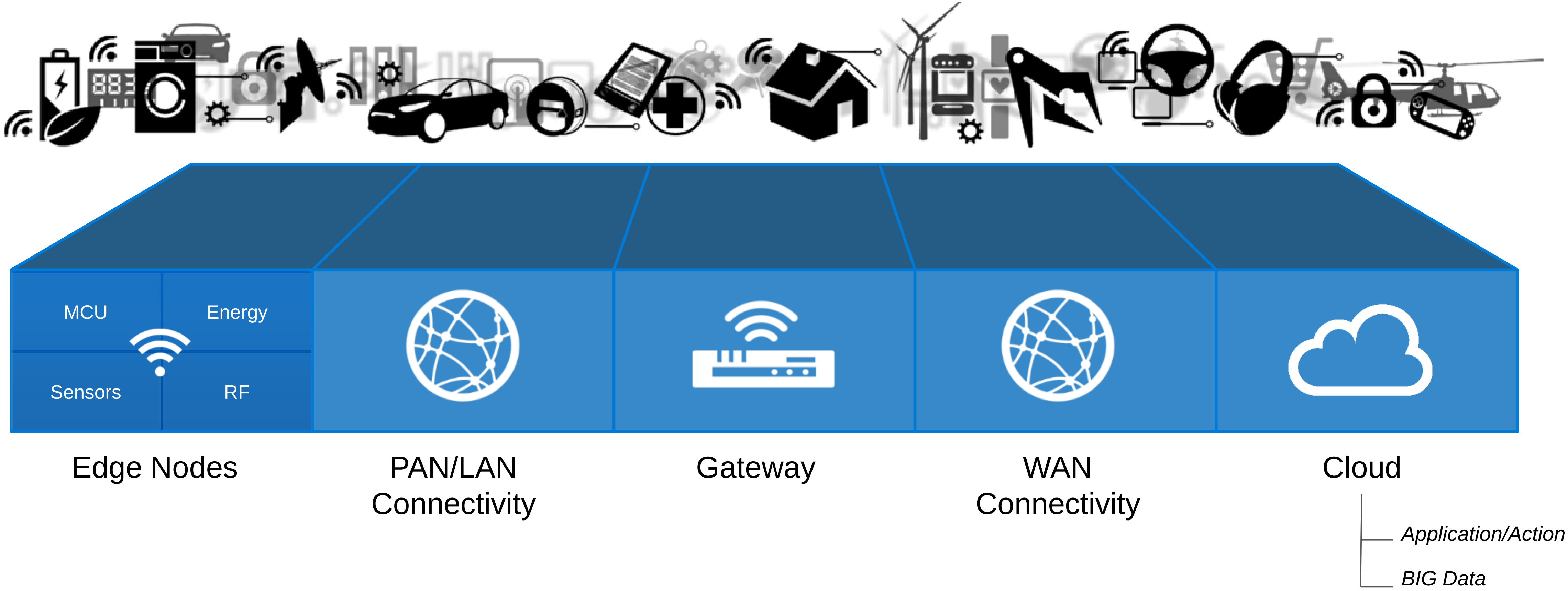
Gateway

WAN Connectivity

Cloud

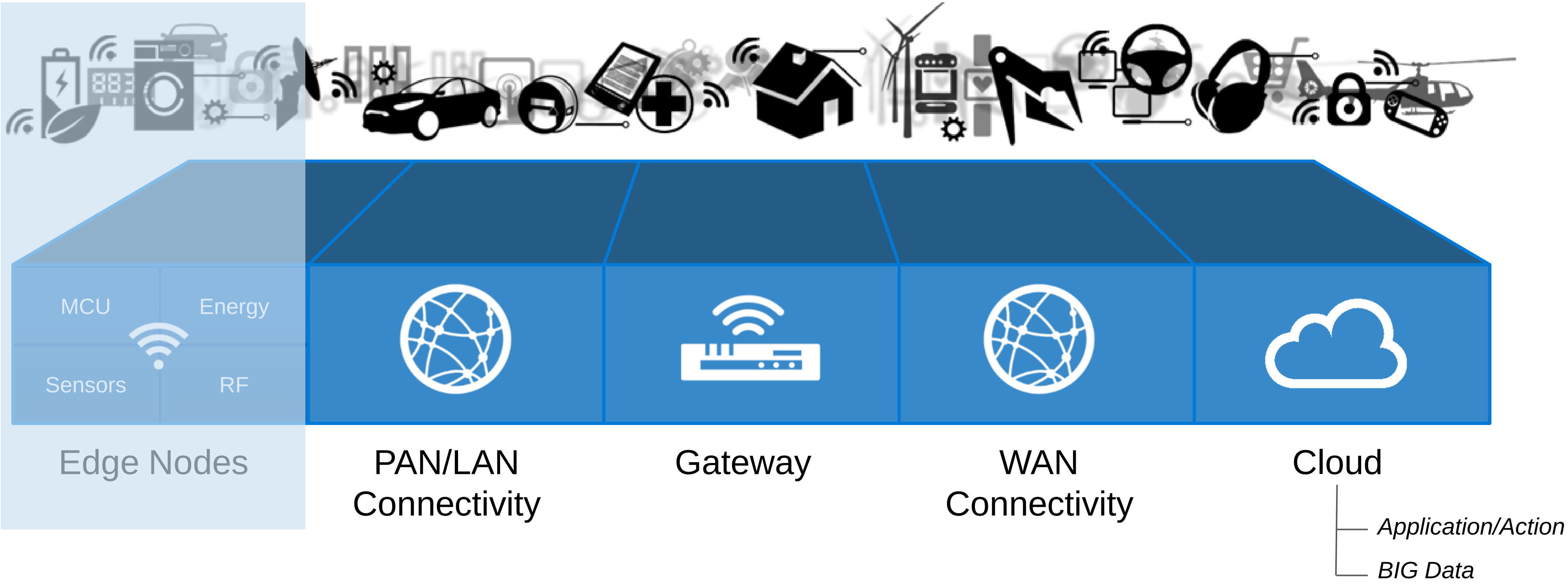
Application/Action

Connecting ‘Things at the Edge’ to the ‘Cloud’



Definition

- Protecting information that would
- be harmful if modified or released.



Challenges for the Smart, Secure, Connected World of Tomorrow

- **Connected accessibility:**
  Connectivity opens new doors for attacks
- **Physical accessibility:**
  Unlimited attacks of remote nodes
- **Data dependence:**
  Harmful repercussions for missing data
- Number: Network overload threats
- **Data misuse:**
  Data used beyond intended methods
- **Mitigation of weaknesses as they arise:**
  Need for secure firmware updates.
- **Social responsibility:**
  The public is no longer forgiving of security breaches

Security Risk Multipliers

- Accessibility
- Firmware updates
- Resource constraints
- Increasing value
- Increased Attacker Capability

Attacks and Attackers

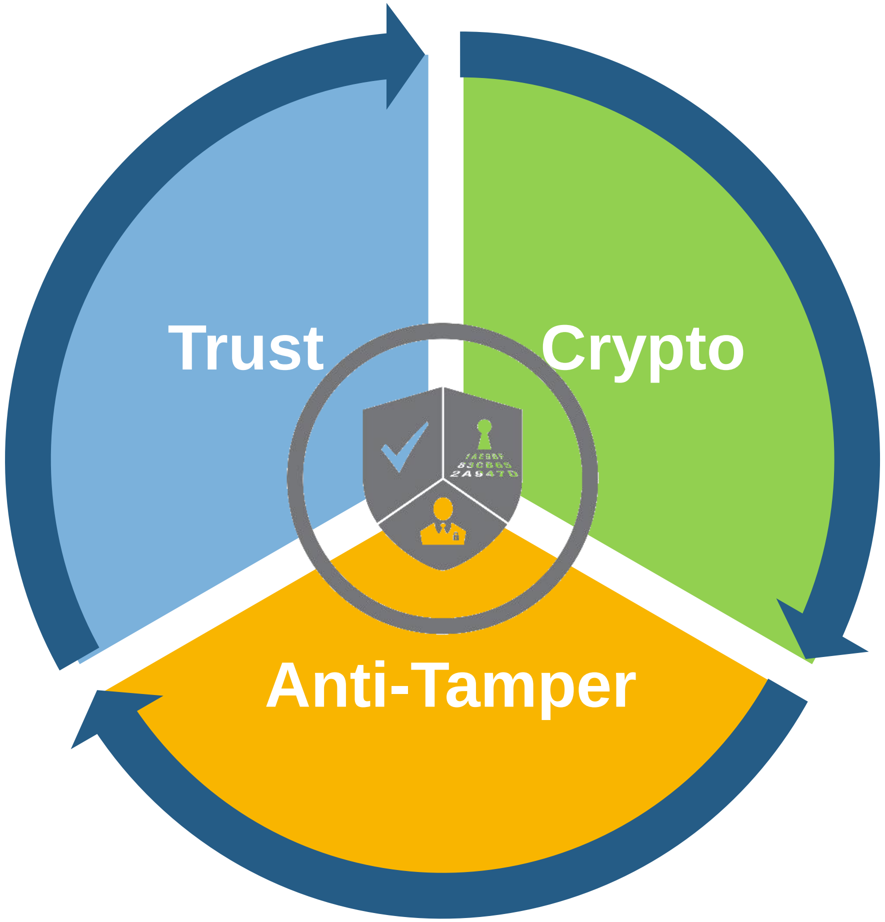
Attacks

- Insider Attacks
  - Financial gain / fraud
  - Revenge / payback
  - Blackmail
- Midnight Attacks
  - Take place during a small window of time
- Focused Attacks
  - Time, money, and resources are not factors

Attackers

- Outsiders (Curious Hackers)
  - Intelligent, but limited knowledge of the system
  - Attempt to use existing security weaknesses
- Insiders (Professionals / Academics)
  - Have significant specialized technical experience
  - Access to sophisticated tools and instrumentation
- Organizations (Crime Syndicates / Governments)
  - Specialists with significant funding resources
  - Advanced analysis tools and in-depth analysis and attacks

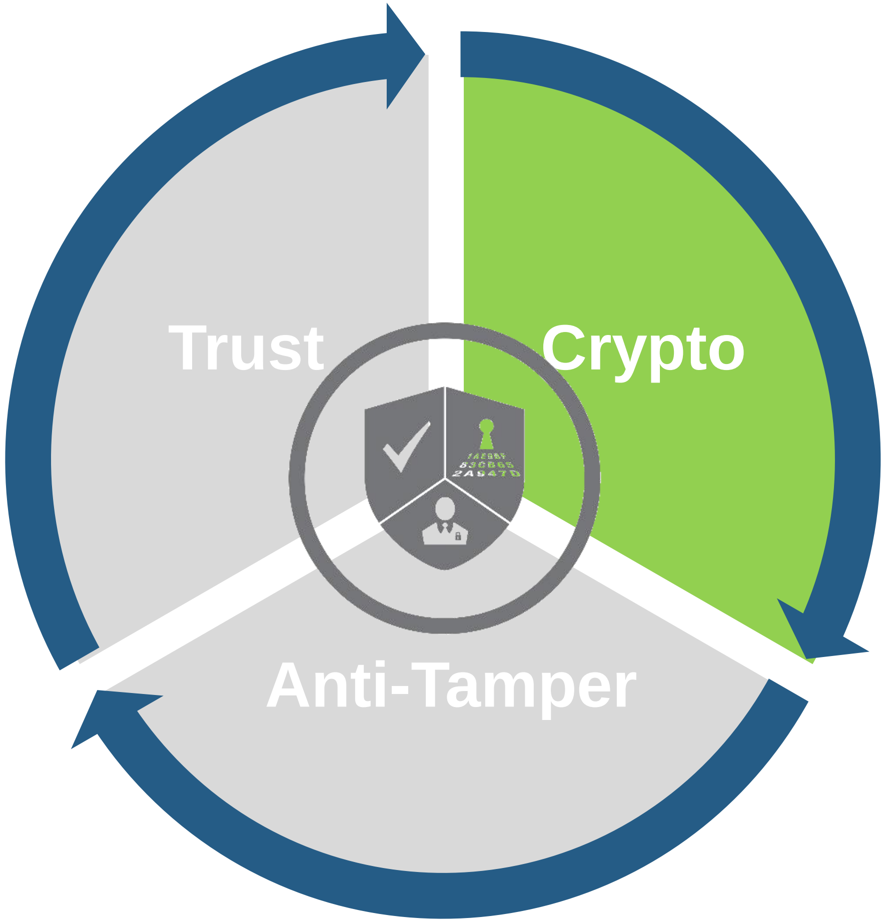
NXP’s Security Technology for Kinetis MCUs



Cryptography

The science of protecting data through encoding and decoding

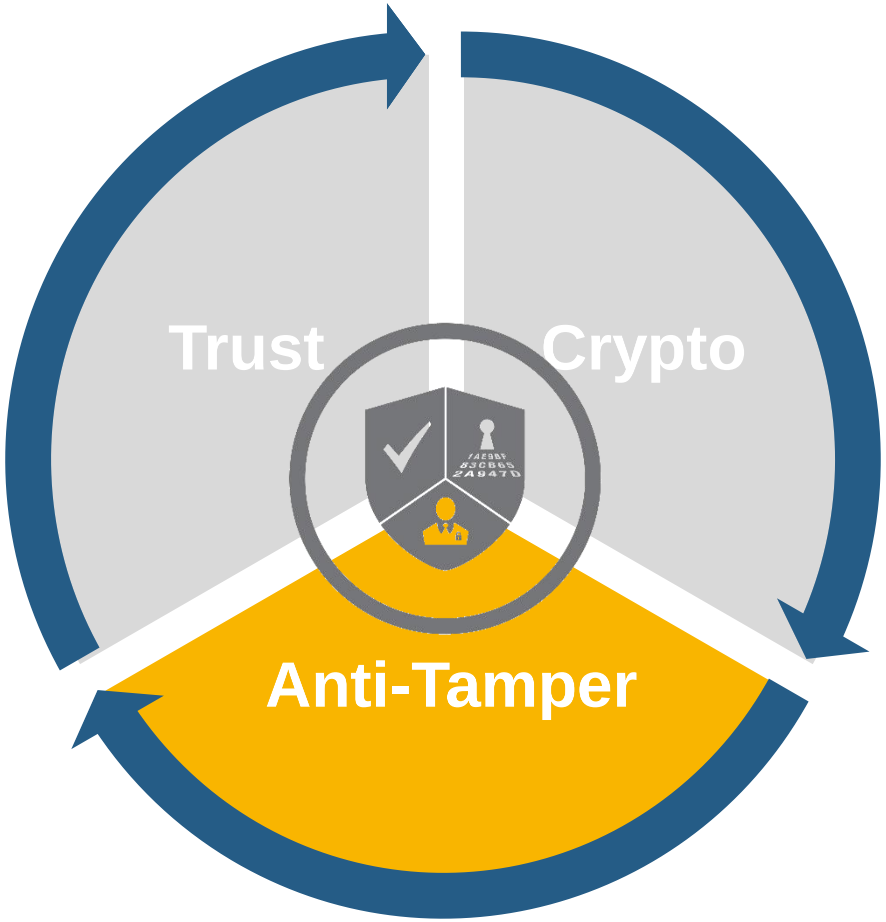
- Symmetric Encryption
  - DES/DES3, AES
- Asymmetric Encryption
  - RSA, ECC
- Hashing
  - CRC, MD5, SHA
- True Random Number Generation
- Security Protocols
  - SSL, HomeKit, Thread



Anti-Tamper

Proactive monitoring of physical and environmental system attacks

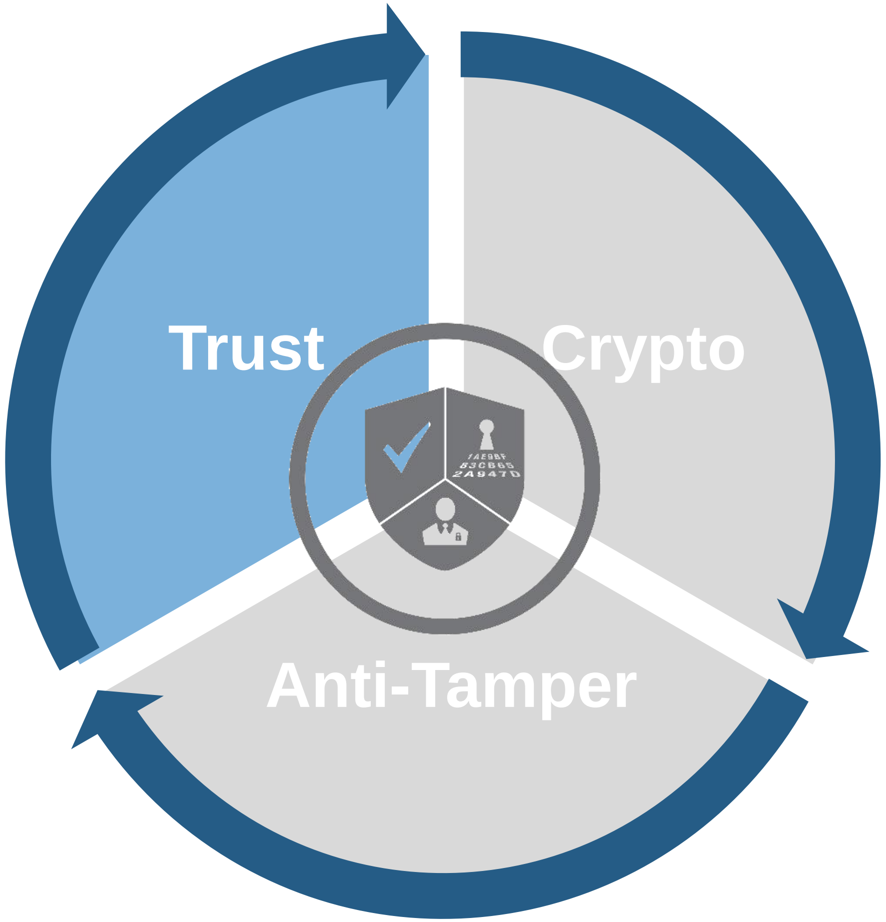
- Tamper Detection
  - Physical
    - Enclosure intrusion
    - Drilling and probing
  - Environmental
    - Voltage
    - Temperature
    - Frequency
- Secure Storage



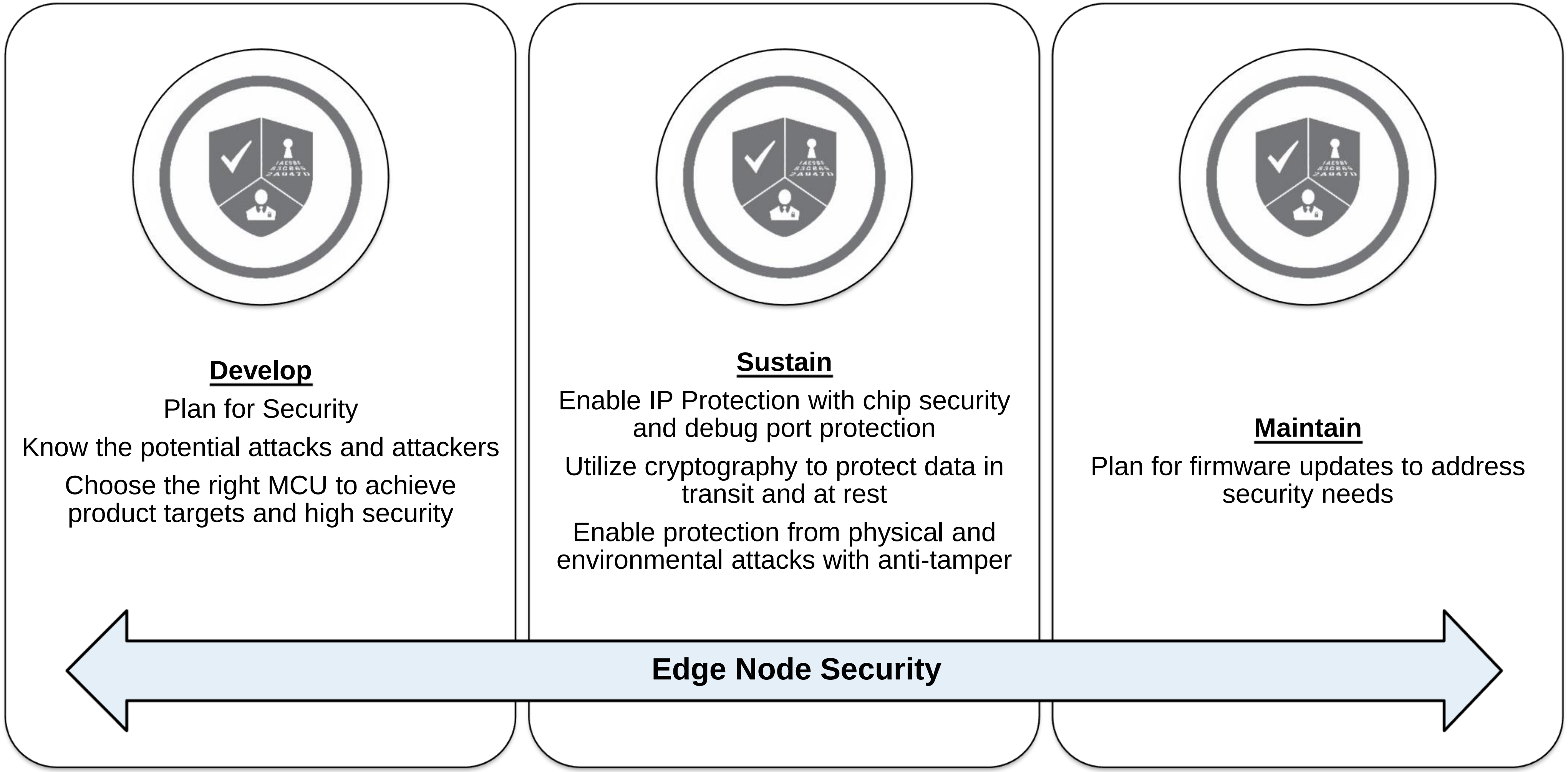
Trust

The assurance that only access from a reliable source will occur

- Code I/P Protection
  - Internal memory protection
  - External memory protection
- Debug Port Protection
- Authentication
  - Software updates
  - Device verification
- Secure Boot



Implementation Guidelines for Embedded Security



Develop

Plan for Security

Know the potential attacks and attackers

Choose the right MCU to achieve product targets and high security

Sustain

Enable IP Protection with chip security and debug port protection

Utilize cryptography to protect data in transit and at rest

Enable protection from physical and environmental attacks with anti-tamper

Maintain

Plan for firmware updates to address security needs

Edge Node Security

Kinetis K8x MCU Family

The industry's most secure MCU platform based on ARM® Cortex®-M core technology

Design with Advanced Security

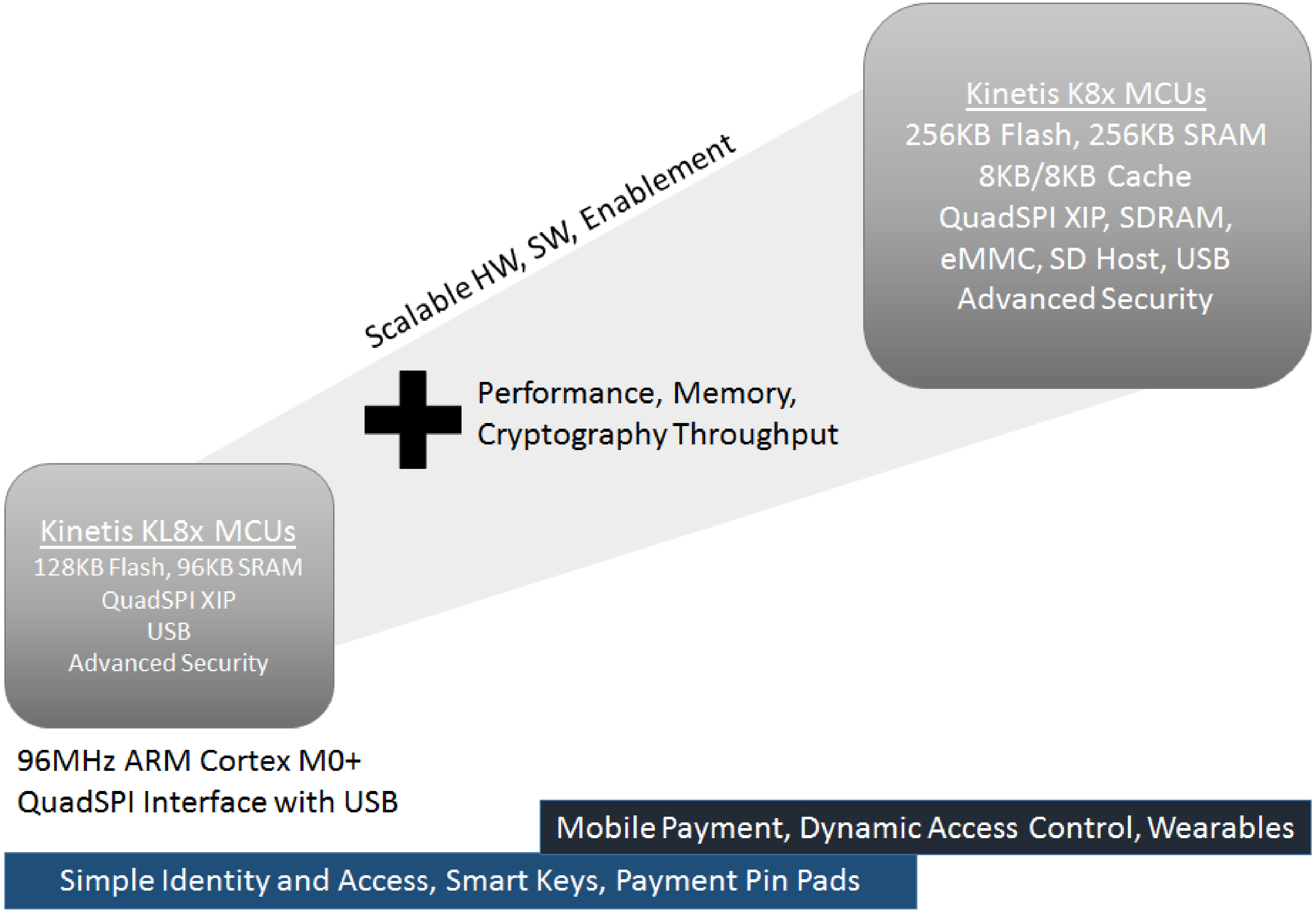
- First multi-purpose MCU with hardware asymmetric cryptography
- Physical anti-tamper capability
- First MCU that supports on-the-fly decryption from external flash

Design with Scalability

- Easily expand with XIP from QuadSPI and external boot option
-

Design with Flexibility

- Explore new functionalities with the FlexIO peripheral



Kinetis KL8x MCUs

128KB Flash, 96KB SRAM
QuadSPI XIP
USB
Advanced Security

96MHz ARM Cortex M0+
QuadSPI Interface with USB

Scalable HW, SW, Enablement

+ Performance, Memory, Cryptography Throughput

Kinetis K8x MCUs

256KB Flash, 256KB SRAM
8KB/8KB Cache
QuadSPI XIP, SDRAM,
eMMC, SD Host, USB
Advanced Security

Mobile Payment, Dynamic Access Control, Wearables

Simple Identity and Access, Smart Keys, Payment Pin Pads

FRDM-K82F: A Platform for Embedded Innovations

FRDM-K82F

Manage Power

- K82 MCU with low static and dynamic IDDs
- 1.8V power rail
- Optional coin cell socket

Connect

- FS USB host and device
- Expansion through shield boards
- Standard BT header
- Standard RF header

Sense & Store

- On chip ADC and comparators
- 6 Axis sensor (FXOS8700)
- 32MB Serial NOR
- On chip temperature sensor

Compute

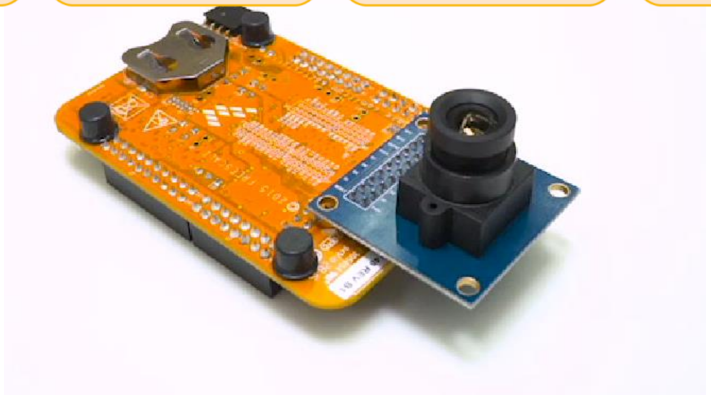
- 150MHz Cortex M4F with code and system cache
- Crypto. engines
- Floating point

Interface

- Capacitive touch pads
- Push buttons

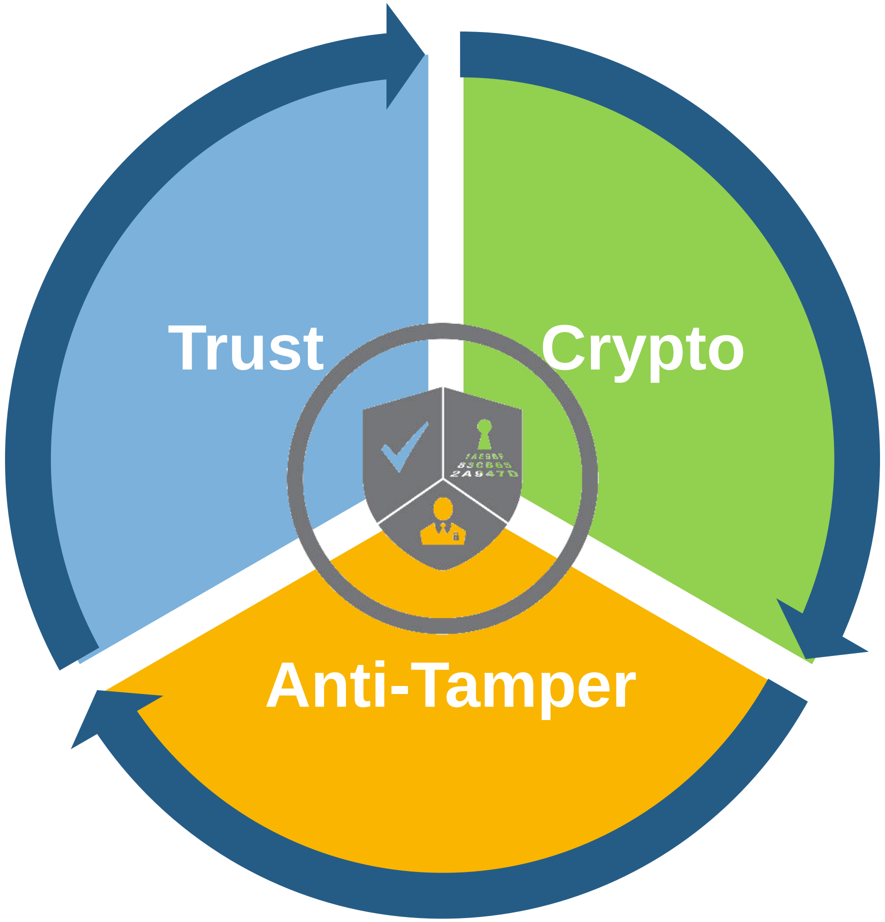
Display & Camera

- FlexIO header for camera or display interface
- Expansion through shield boards



CRYPTO

Crypto – Protecting Data with Encoding



Crypto: A Handful of Kinetis Support

Feature	Benefit	Feature Details	Enablement
Symmetric Cipher Hardware Acceleration	Reduce the time and/or software overhead for crypto	DES/3DES and AES acceleration using mmCAU or LTC	KSDK, mmCAU library and API user’s guide, AN4307
Hashing Function Hardware Acceleration	Reduce the time and/or software overhead for crypto	MD-5, SHA-1, and SHA-256 acceleration using mmCAU or LTC	KSDK, mmCAU library and API user’s guide, AN4307
Asymmetric Cipher Hardware Acceleration	Reduce the time and/or software overhead for crypto	RSA or ECC acceleration using LTC	KSDK and third party stacks

Symmetric Encryption

- **Definition**
  - Symmetric-key encryption is the set of cryptographic algorithms that use the *same* cryptographic key for both the encryption of data into plaintext and decryption of the ciphertext back into data.
- **Purpose**
  - Provide a method for two parties to exchange messages on a public communication channel and keep the information secret (privacy).
- **Algorithms**
  - Data Encryption Standard (DES) – 56-bit keys
  - Triple Data Encryption Algorithm (3DES) – 168-bit keys
  - Advanced Encryption Standard (AES) – 128-, 192-, and 256-bit keys
- **Block Cipher Modes**
  - Electronic Code Book (ECB), Counter (CTR), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), etc.

Asymmetric Encryption

- **Definition**
  - Asymmetric-key encryption is cryptographic algorithms that use *different* cryptographic keys (public and private) for the encryption of plaintext and decryption of the ciphertext.
- **Purpose**
  - Provide a method for two parties to exchange messages on a public communication channel and keep the information secret (privacy, authentication, and non-repudiation)
- **Algorithms**
  - Rivest-Shamir-Adleman (RSA) – 1024-, 2048-, 3072-, 4096-bit keys
  - Elliptical Curve Cryptography (ECC) – 160-, 224-, 256-, 384-bit keys

Hashing

- **Definition**
  - Hashing is an algorithms that can be used to map data of arbitrary size to data of fixed size.
- **Purpose**
  - Provides the ability to check the integrity of the data that has been received to ensure that it has not been modified (integrity)
- **Algorithms**
  - Cyclic Redundancy Check (CRC) – 16-, 32-bit hash
  - Message Digest Algorithm (MD5) – 128-bit hash
  - Secure Hash Algorithm 1 (SHA-1) – 160-bit hash
  - Secure Hash Algorithm 2 (SHA-2) – 224-, 256-, 384-, 512-bit hash

True Random Number Generator

- **Definition**
  - An apparatus that generates random numbers from a physical process with a significant amount of entropy, rather than a computer program.
- **Purpose**
  - Provide a method for generating initial key values for encryption algorithms that can not be guessed or duplicated.
- **Methods**
  - Thermal noise
  - Voltage fluctuations
  - Photoelectric effect
  - Quantum phenomena

Crypto Algorithms/Protocols

	Required	Used
OTA secure firmware update Secure boot	RSA-2048 verify SHA-256 over firmware image	At each update At each boot
HomeKit	SRP-3072 Ed22519 sign/verify Curve-25519 SHA-512 based KDF ChaCha20 cipher Poly-1305 MAC SHA-256	At first device pairing At first device pairing At each connection with accessory At each connection with accessory
Thread	EC-JPAKE (NIST-P256) AES-128 CCM (TLS) HMAC-SHA256 based KGF SHA-256	At first device pairing
AllJoyn	ECDHE-PSK ECDHE-ECDSA ECDHE-NULL NIST-P256 X509 Certificates SHA-256	At each connection

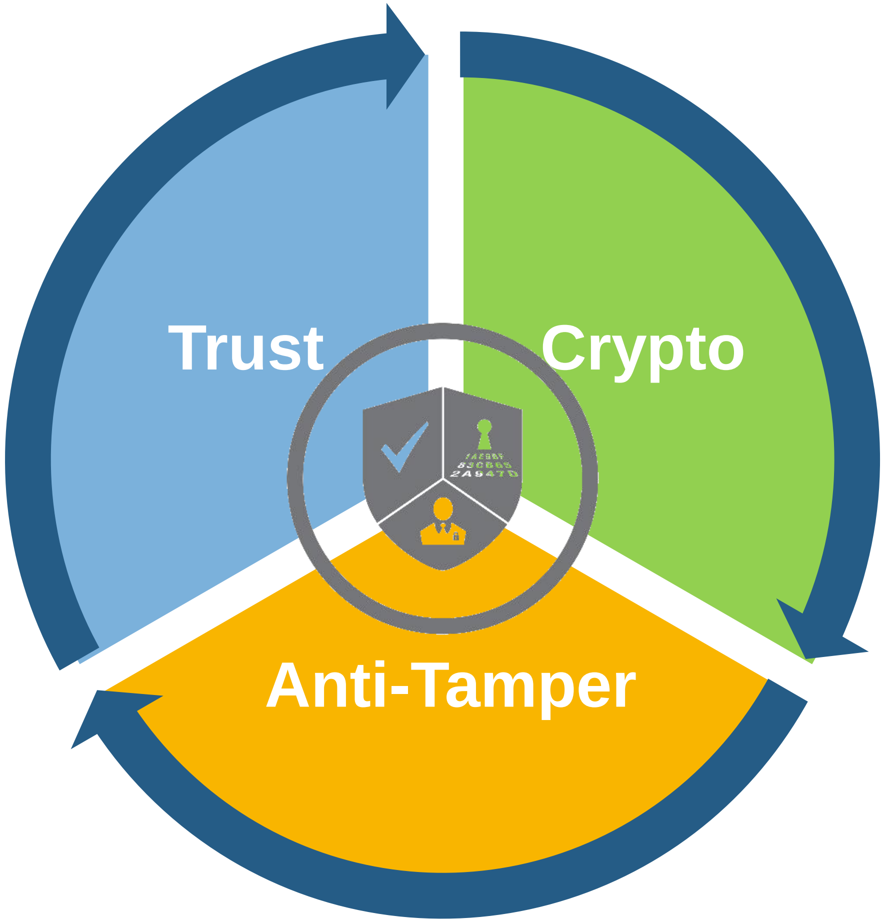
CRYPTO THROUGHPUT LAB

Crypto HW Acceleration Lab

We’ll run a crypto benchmarking algorithm using LTC acceleration, mmCAU acceleration, and pure software so that you can see the difference in performance first hand.

ANTI-TAMPER

Tamper Resistance – Detect and React to Attacks



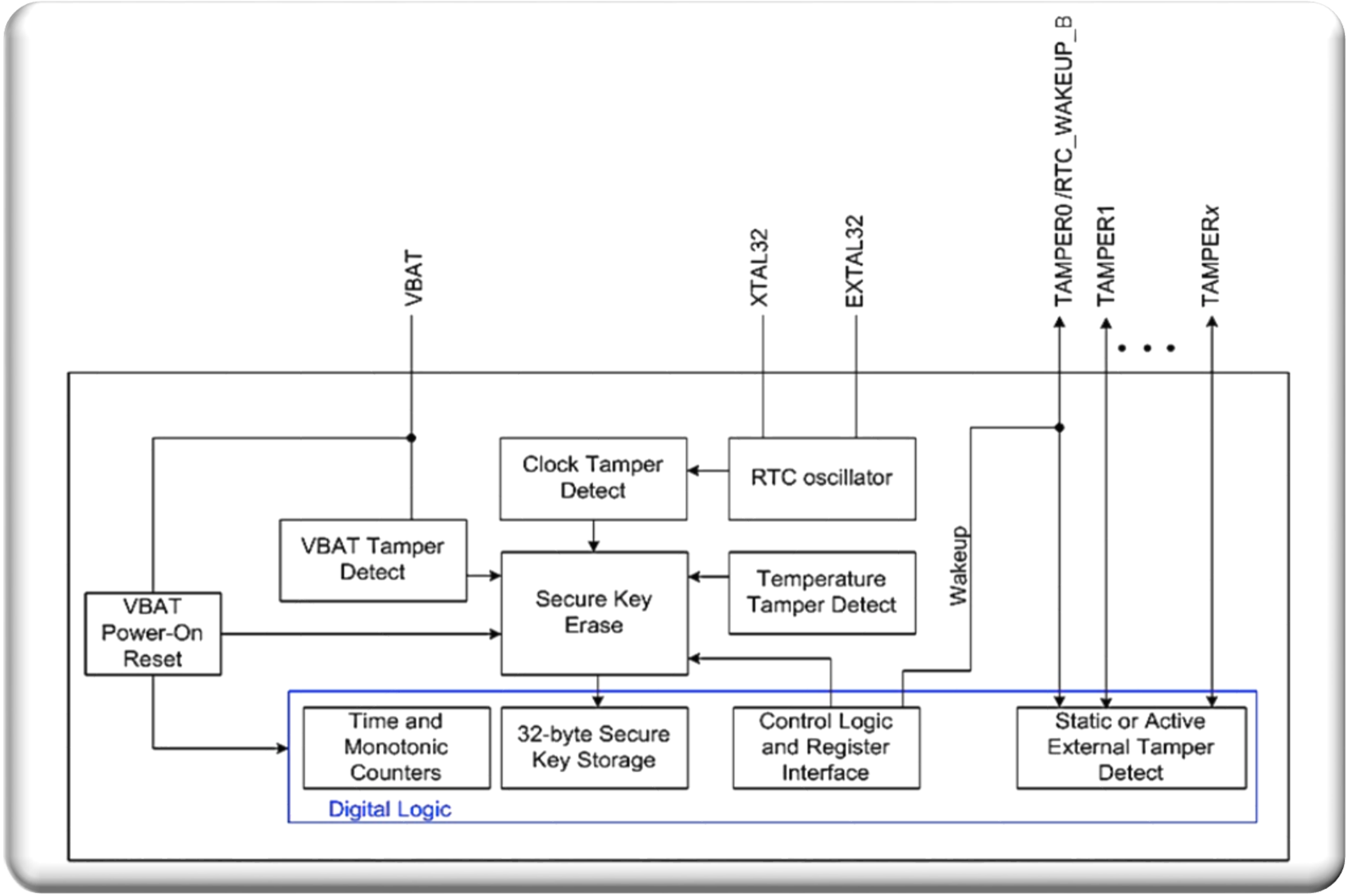
Tamper Resistance: A Handful of Kinetis Support

Feature	Benefit	Feature Details	Enablement
Tamper detect module with up to 8 tamper pins	Reduce external circuits needed to support Tamper Resistance mechanisms	Tamper detection for pin, temperature, voltage and clock. As well as active tamper.	Application note available for NDA customers
Secure storage	Key storage that is automatically erased on tamper (no software intervention required)	Secure key storage space with asynchronous erasure when external tamper events occur.	Application note available for NDA customers

DryIce (Tamper Detection Module) Features

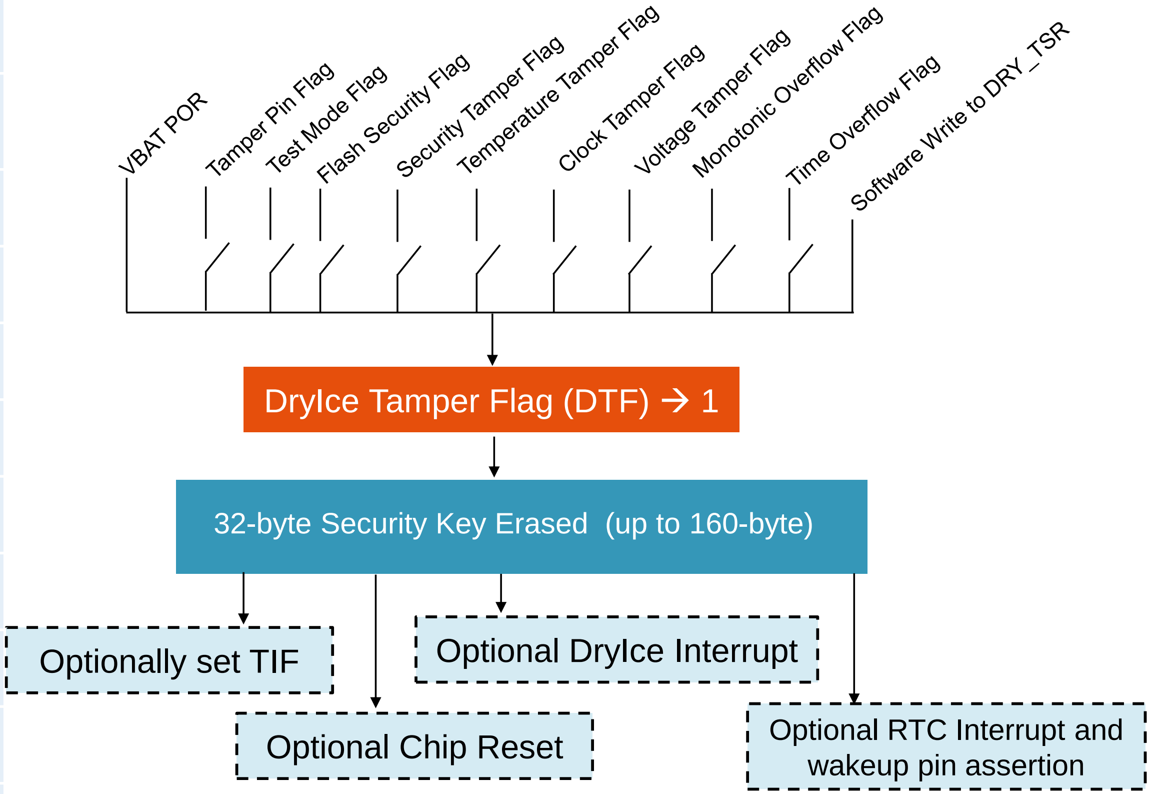
- Independent power supply (VBAT), power-on reset (POR) and 32 kHz crystal oscillator (RTCOSC)
- 32-bytes (8 x 32-bit words) of secure storage that are erased upon a tamper event
- Tamper time register that records time of tamper event
- Register lock protection to prevent read and write access
- Up to 10 internal tamper sources plus software-initiated tamper capable of generating interrupt or tamper event
- Up to 8 external tamper pins capable of generating an interrupt or tamper event
- Two active tamper shift registers each with configurable polynomial

DryIce/Secure RTC Block Diagram



DryIce Tamper Event

Events That Can Set DryIce Tamper Flag (DRY_SR[DTF])
External
VBAT Power-On Reset (POR)
Assertion of DryIce Tamper Pin x Flag (TPFx) [Up to 8 pins]
Internal
Assertion of DryIce Test Mode Flag (TMF)
Assertion of DryIce Flash Security Flag (FSF)
Assertion of DryIce Security Tamper Flag (STF)*
Assertion of DryIce Temperature Tamper Flag (TTF)
Assertion of DryIce Clock Tamper Flag (CTF)
Assertion of DryIce Voltage Tamper Flag (VTF)
Assertion of DryIce Monotonic Overflow Flag (MOF)
Assertion of DryIce Time Overflow Flag (TOF)
Software-Initiated Write to DRY_TSR (no flag)



VBAT POR

Tamper Pin Flag

Test Mode Flag

Flash Security Flag

Security Tamper Flag

Temperature Tamper Flag

Clock Tamper Flag

Voltage Tamper Flag

Monotonic Overflow Flag

Time Overflow Flag

Software Write to DRY_TSR

DryIce Tamper Flag (DTF) → 1

32-byte Security Key Erased (up to 160-byte)

Optionally set TIF

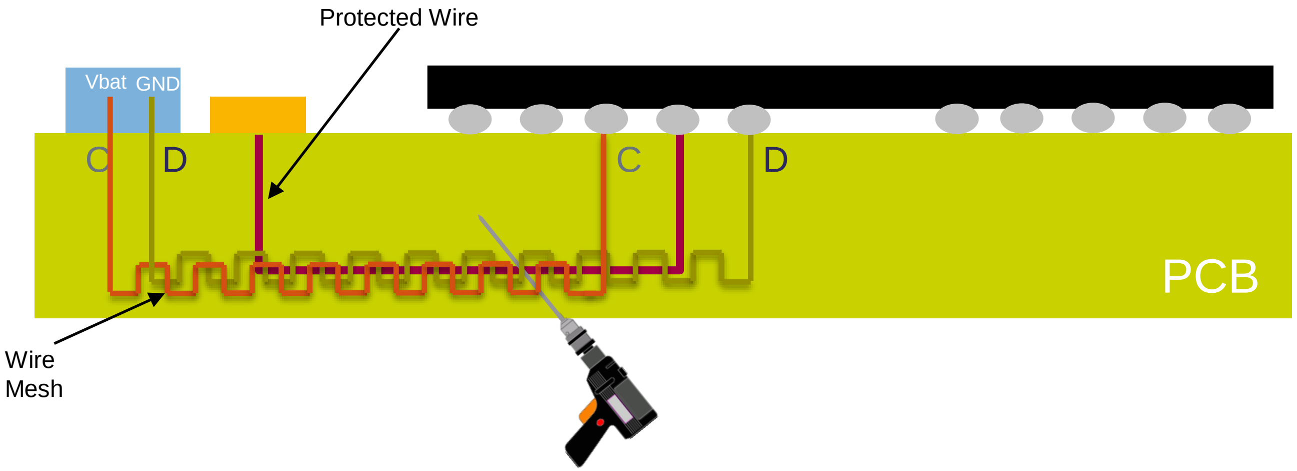
Optional DryIce Interrupt

Optional Chip Reset

Optional RTC Interrupt and wakeup pin assertion

External Physical Tamper Detection

DryIce supports static and active tamper configurations



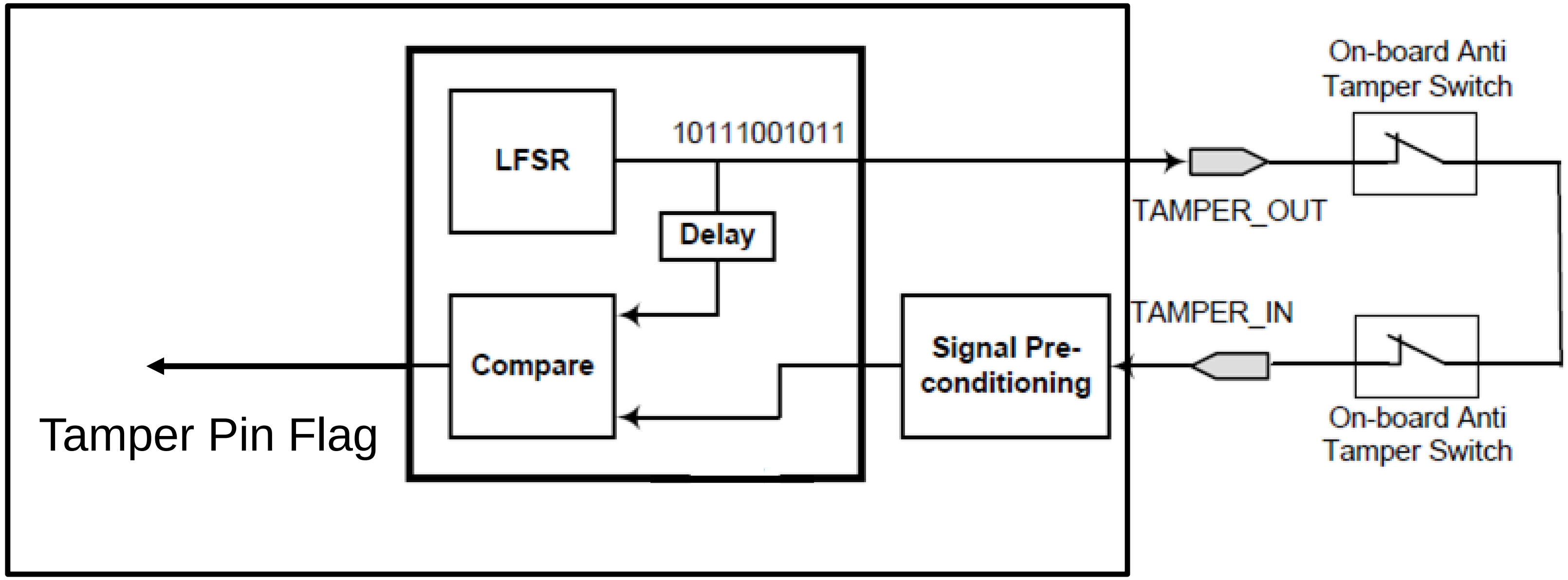
DRYICE module will detect Tamper and initiate key erasure when:

- C is disconnected (floating)
- D is disconnected (floating)
- C and D are short-circuited

Active Tamper Detection

How it works:

- DryIce can output up to two unique active tamper output signals. Each of the active tamper output pins outputs a random value that can change every second (1 Hz).
- For every active tamper output pin, at least one associated active tamper input pin should be designated. The input pin expects to see the associated active tamper output signal (with some propagation delay allowed as configured using a glitch filter).



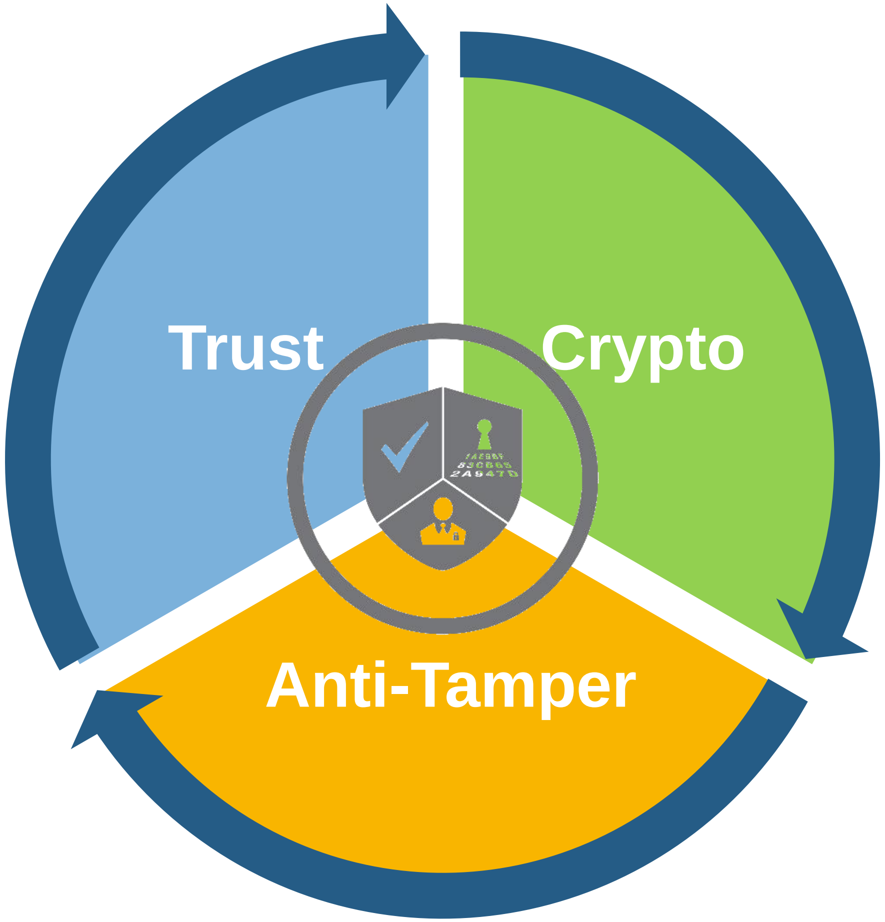
Secure Storage

- Amount of secure storage associated with the DryIce module can vary from device to device

- Latest K8x parts support:
  - 32-bytes of secure key storage in the DryIce block which is erased on tamper (battery backed)
  - 128-byte VBAT register file (battery backed memory that is optionally erased on tamper as determined by the DRY_CR[SRF] setting)
  - 2KB secure session RAM that is erased on a tamper and system reset

TRUST

Trust – Ensuring Operation from Reliable Sources



Trust: All Kinetis Devices Support

FEATURE	BENEFIT	DETAILS	ENABLEMENT
On chip Flash security and protection mechanisms	Protection from firmware theft and application cloning	Ability to prevent debug access to the processor. Ability to set a 64-bit key to regain debug access.	**AN4507:** “*Using the Kinetis Security and Flash Protection Features*”
Debug port configuration	Block external access to debug or flash re-programming	Flash security disables debug port. JTAG pins can also be disabled via software	**AN4507:** “*Using the Kinetis Security and Flash Protection Features*”
Unique ID	Software can be used to uniquely identify the MCU as a trusted device	**L-Series:** 80-bit unique ID **All Others:**128-bit unique ID	
Boot from internal memory only	Controlled boot conditions to avoid attacks that use external memories	Boot from on-chip flash or ROM only	

Trust: A Handful of Kinetis Devices Support

FEATURE	BENEFIT	DETAILS	ENABLEMENT
Execute and supervisor only access	Protection of software IP	Non-Volatile control registers to set access privileges of on chip flash resources. Supervisor or execute only access can be set for up to 64 different segments	**AN5112:** "*Using the Kinetis Flash Execute-only Access Control Feature*"
Encrypted firmware updates	Protection from firmware theft and application cloning	ROM enables encrypted image downloads for internal flash or external serial NOR. Code is stored in the clear on the device.	Kinetis Bootloader K80 Tools & KBOOT
On-the-Fly AES Decryption (OTFAD)	Protection from firmware theft and application cloning	Code in external serial NOR flash can be stored encrypted in the external memory and is only decrypted as it is read by the processor	Kinetis Bootloader K80 Tools & KBOOT

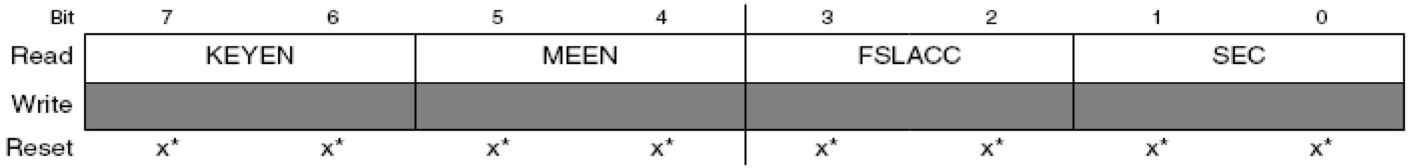
Flash Security and Protection Features

- Flash security and protection features are found on all Kinetis devices

- **Security Features**
  - Kinetis offers several levels of flash security
  - Flash security is a system-level feature
    - The flash is fully functional when secured (firmware updates are still possible if resident firmware is setup to program the flash)
    - Security effects are really a system level concern. The security setting determines what the SoC will allow.
  - **Software IP is a large investment. Enabling security helps to protect that IP investment.**

- **Protection Features**
  - Flash protection can be used to prevent accidental erase or programming
  - Initial protection values are loaded from the flash configuration field at reset

Flash Security Register (FSEC)



Bit	7	6	5	4	3	2	1	0
Read	KEYEN		MEEN		FSLACC		SEC	
Write	▨		▨		▨		▨	
Reset	x*	x*	x*	x*	x*	x*	x*	x*

KEYEN	Backdoor Key Access
00	Disabled
01	Disabled (preferred setting)
10	Enabled
11	Disabled

FSLACC	Freescale Factory Access
00	Granted
01	Denied
10	Denied
11	Granted

MEEN	Mass Erase
00	Enabled
01	Enabled
10	Disabled
11	Disabled

SEC	Security
00	Secure
01	Secure
10	Unsecure (shipping state)
11	Secure

Security settings are loaded from the flash configuration field at reset.

Execute-Only Application Code

The concept of execute-only application code is:

- A primary developer (for example, NXP or a 3rd-party software provider) creates useful, value added application(s) for a given MCU or MPU. Entry point and use information (for example, ABI details) is supplied, but source code is not. This code is assumed to be “trusted”.
- A method exists to load this software into a part without exposing it during the loading process and mark it “execute-only”.
- Other developers and/or users add more software to these parts and use these application(s).
- The user of the application(s) will not be able to see or expose any of the protected execute-only code.
- There may be multiple layers of “execute-only” application code added with all previously added layers protected.
- This feature is available on newer Kinetis devices and are controlled from the Flash FTF register set.

New Features in the ROM Bootloader for K8x Family

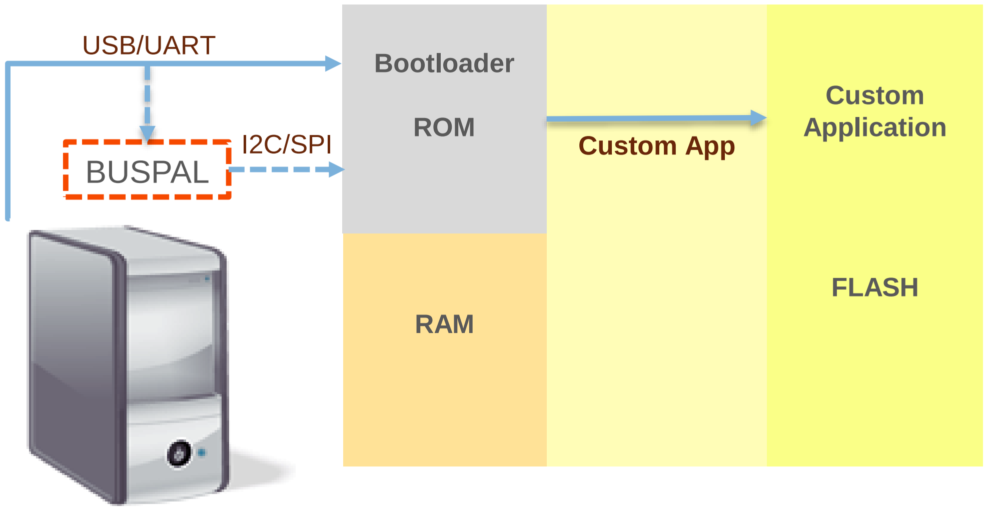
- Support for downloading encrypted binary files (sb file format) to internal flash or external serial NOR flash (available on all K8x MCUs)

- Configuration of on-the-fly decryption (OTFAD) module for executing encrypted image stored in external serial NOR flash with no added latency (feature only available on K81/K82 MCUs)

ROM Bootloader

- Available on K80 and other Kinetis devices featuring a boot ROM.
- Pre-programmed in device ROM; failsafe boot mechanism for factory and field programming
- Configured specifically for each MCU family
  - Peripheral interfaces
  - Command set
- User configurable via parameters stored in user flash (bootloader configuration area, BCA)
- Can run at system start-up and is callable from user application at runtime

Kinetis MCU w/ ROM



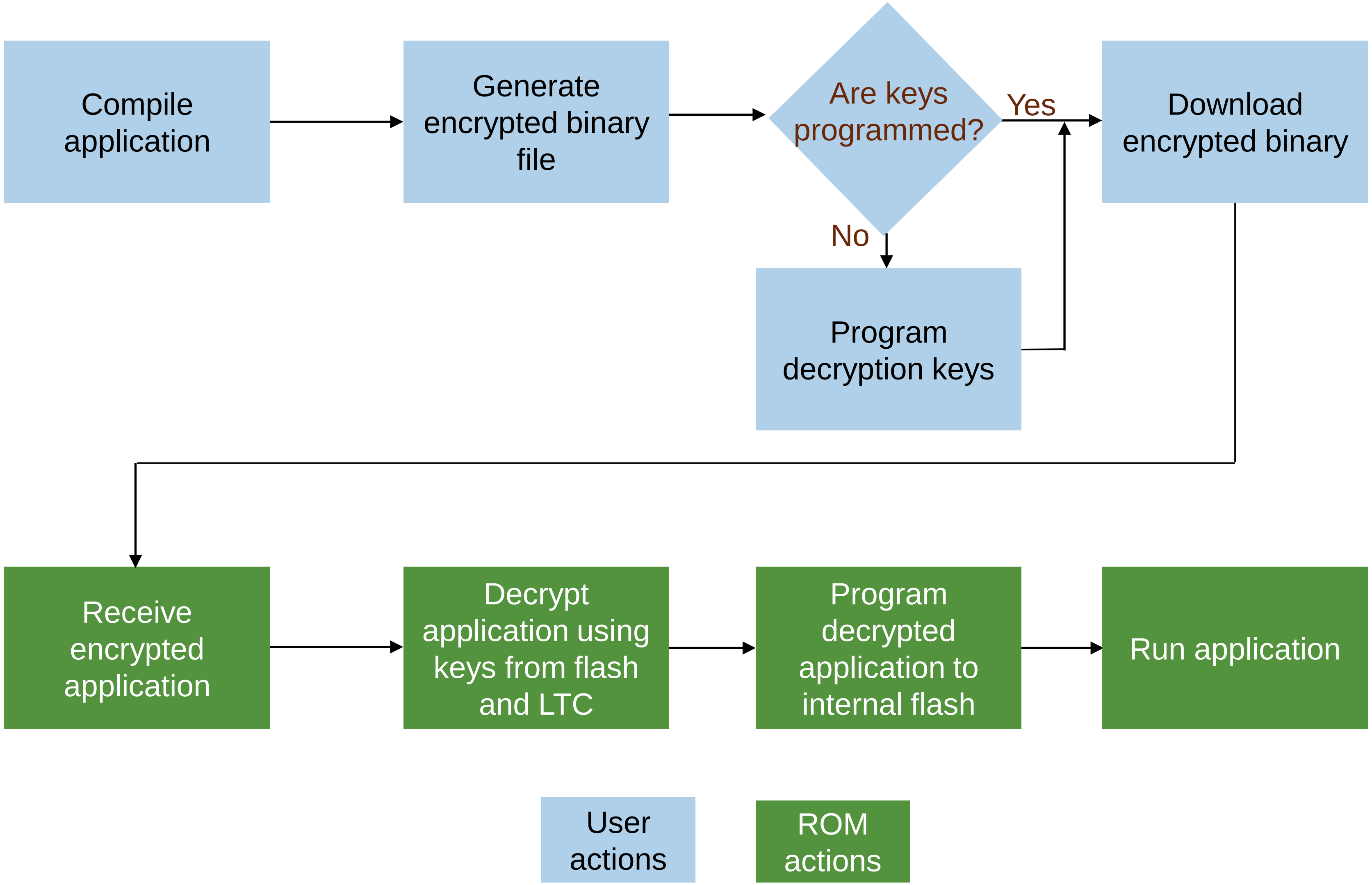
ROM Bootloader
(MCU specific)

Encrypted Binary File Downloads (K8x Family)

- The ROM supports downloading encrypted binary files in an Secure Binary (SB) file format (AES-128 CBC-MAC encryption is used)
- Application is download encrypted, but stored to internal flash decrypted (there is also an option to download an encrypted file to external serial NOR flash too)
- A key value pre-programmed into the flash “eraseable program once” space is used for decryption
- Encrypted SB files can be downloaded when device is secure or unsecure, but if the device is secure, then the ROM disables all memory reads and writes with the exception of encrypted SB files.
- So if you enable security, encrypted binaries are the only way the ROM can be used to update the firmware
- Tools that generate encrypted binaries are available with the KBOOT software

- On K81/K82 devices the LP Trusted Cryptography (LTC) module is used to decrypt the binary
- On K80 devices the Memory Mapped Cryptographic Acceleration Unit (mmCAU) is used to decrypt the binary

Encrypted File Download Process (K81/K82 MCUs)



Compile application

Generate encrypted binary file

Are keys programmed?

Yes

No

Program decryption keys

Download encrypted binary

Receive encrypted application

Decrypt application using keys from flash and LTC

Program decrypted application to internal flash

Run application

User actions

ROM actions

Bootloader Configuration Area (BCA)

- The Bootloader configuration area (BCA) holds optional configuration parameters
- BCA for K8x ROM is in internal flash at address 0x3C0
- For the K81/K82 none of the BCA values are required to enable encrypted binary downloads, but on the K80 an mmCAU configuration pointer is needed.
- mmCAU configuration pointer is at offset 0x20 (absolute address 0x3E0)
- Can be modified by the *write memory* command or can be set by the application image (similar to flash protection and security area)
- BCA is loaded at reset, so if the BCA is changed the new value doesn’t take effect until a reset
- Includes options such as enabled peripherals, peripheral-specific settings, and bootloader timeout

Initial Application Load for K8x MCUs

- On the K80 the BCA, mmCAU struct, and AES functions must be loaded into the processor in order to perform an encrypted binary download

- For all K8x processors you need to think about where/when the key loading will take place

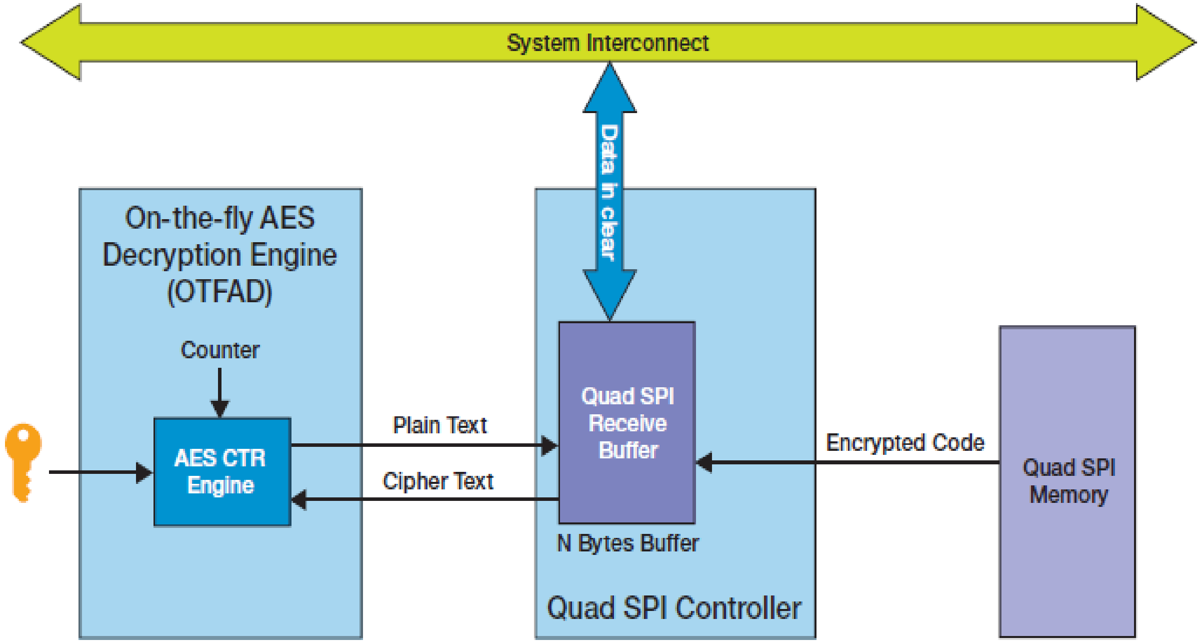
- If the first application load is done in a trusted environment…
  - Initial programming of a device can use an unencrypted binary of the application
  - As usual the application should contain a valid BCA
  - mmCAU struct and AES functions can be programmed to a fixed location or included as part of the application (for the K80)
  - Keys can be programmed the first time the application runs or as a step in the initial load
  - After the initial load is done the part has everything configured so that subsequent loads can use an encrypted binary file with the regular flow

- If the first application load is not done in a trusted environment…
  - The keys should be pre-loaded in a trusted environment
  - Valid BCA, mmCAU struct, and AES functions must be loaded before the encrypted application can be downloaded. This could be loaded at the same time as keys or a binary could be provided to the production environment. If the mmCAU struct and AES functions are part of the application normally, then the versions loaded to allow the first app download to work can potentially reside in a different location than the final versions that will be part of the application.
  - After the initial load is done the part has everything configured so that subsequent loads can use an encrypted binary file with the regular flow

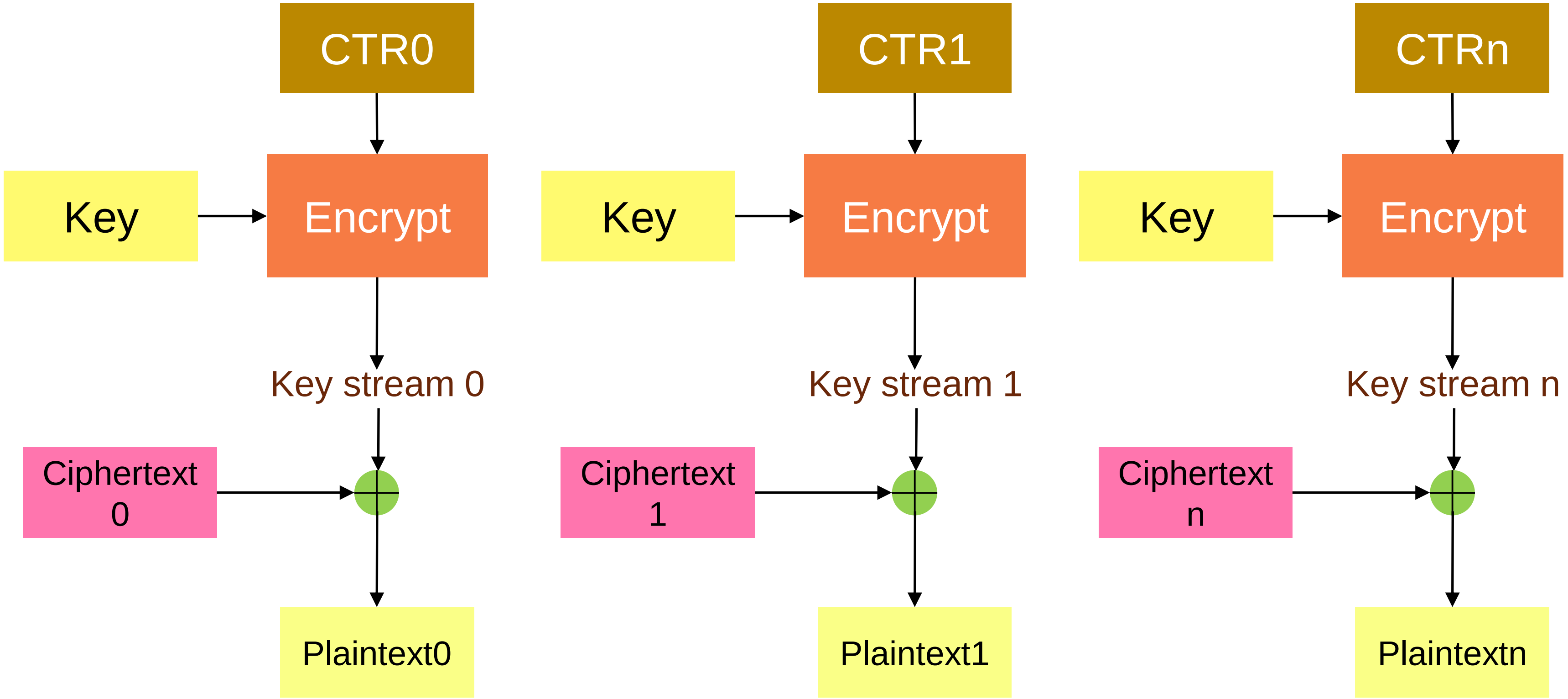
OTFAD LAB

On-The-Fly AES Decryption (OTFAD) Features (K8x Family)

- Allows on-the-fly decryption of the encrypted code in Quad SPI
- Allows to Execute-in-Place encrypted code from Quad SPI
- Based on AES128-CTR Symmetric Algorithm
- OTFAD engine post decryption, transfers the data in clear back to QuadSPI Rx buffer that is then available for the system.
- Provides *anti-cloning* and *IP protection* capabilities by securing customer end product code and data
- Hardware support for 4 independent decryption segments, known as memory context
- Each context has a unique 128-bit key, 64-bit counter and 64-bit memory region descriptor



OTFAD Decryption Process



- Counter and key are what actually get encrypted.
- Key stream can be pregenerated
- If key stream value is available already, then a simple XOR is all that is needed to get plaintext.
- This is how the OTFAD works without adding latency.

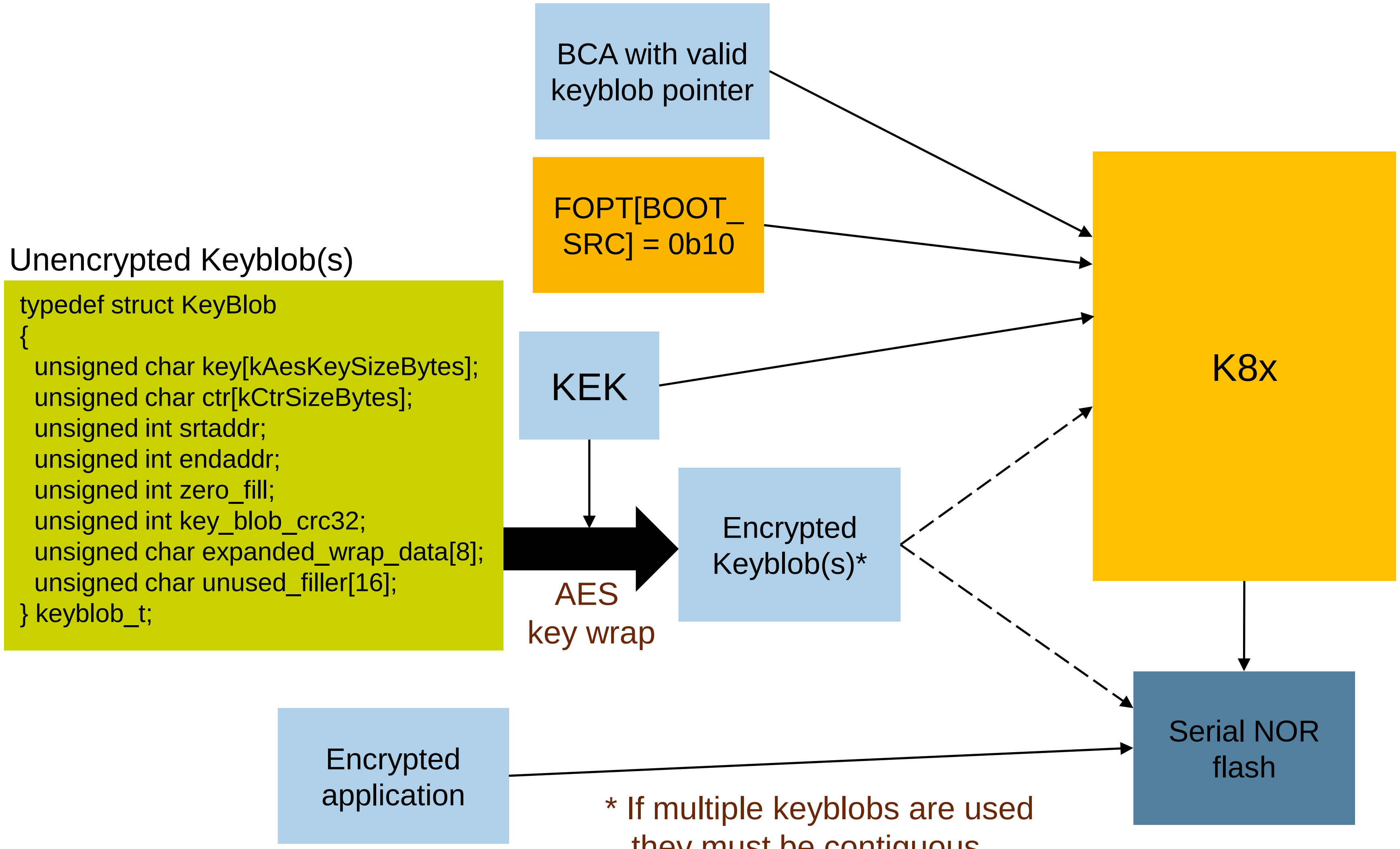
OTFAD Advantages

- Better security than Electronic Code Book (ECB) mode cryptographically. In Counter (CTR) mode, same plaintext input results in the different ciphertext output, so patterns in the input cannot be easily eavesdropped like in ECB mode.
- Unique counter (CTRn) for every 128-bit block ensures unique value of the key stream per 128-bit block.
- Unlike the traditional approach of moving the decrypted code in on-chip SRAM, technique relies to directly execute from external Flash making it more secure and immune against SRAM attacks to access the code.
- Generation of Key stream is dependent on “System bus address” [to generate 4-byte counter value] and not on the input data stream, generation of key stream per 128-bit block can be done in parallel with QuadSPI fetch from external memory, effectively resulting in no additional overhead to random or sequential reads from QuadSPI.
- To avoid initial latency, generation of key stream is pipelined and stores two encrypted counters

OTFAD Key Management

- OTFAD needs more information than just a key, so instead of storing a single key value there is a structure called a keyblob
- The keyblob contains:
  - Key
  - Initial counter value
  - Start address for memory context
  - End address for memory context
  - CRC32 over first 32-bytes of the keyblob (used for verification of the keyblob)
  - Several sections of padding (0s)
- The keyblob goes through an AES-128 key wrap where it is encrypted using a key encryption key (KEK)
- The wrapped/encrypted keyblob is what is programmed into the processor (pointed to by the keyblob pointer in the BCA)
- The KEK is written to the internal flash “eraseable program once” space
- Unlike the location for the mmCAU/LTC key, the KEK location is not readable using the flash read once command
- If the part is configured for QSPI boot, the ROM will look for a valid KEK and one of more valid keyblobs (decrypted CRC32 is used to validate unwrapped keyblob). If valid keyblobs are successfully unwrapped the ROM will configure the OTFAD module for decryption.

Components to Program When Using OTFAD



BCA with valid keyblob pointer

FOPT[BOOT_SRC] = 0b10

KEK

K8x

Unencrypted Keyblob(s)

```
typedef struct KeyBlob
{
  unsigned char key[kAesKeySizeBytes];
  unsigned char ctr[kCtrSizeBytes];
  unsigned int srtaddr;
  unsigned int endaddr;
  unsigned int zero_fill;
  unsigned int key_blob_crc32;
  unsigned char expanded_wrap_data[8];
  unsigned char unused_filler[16];
} keyblob_t;
```

AES key wrap

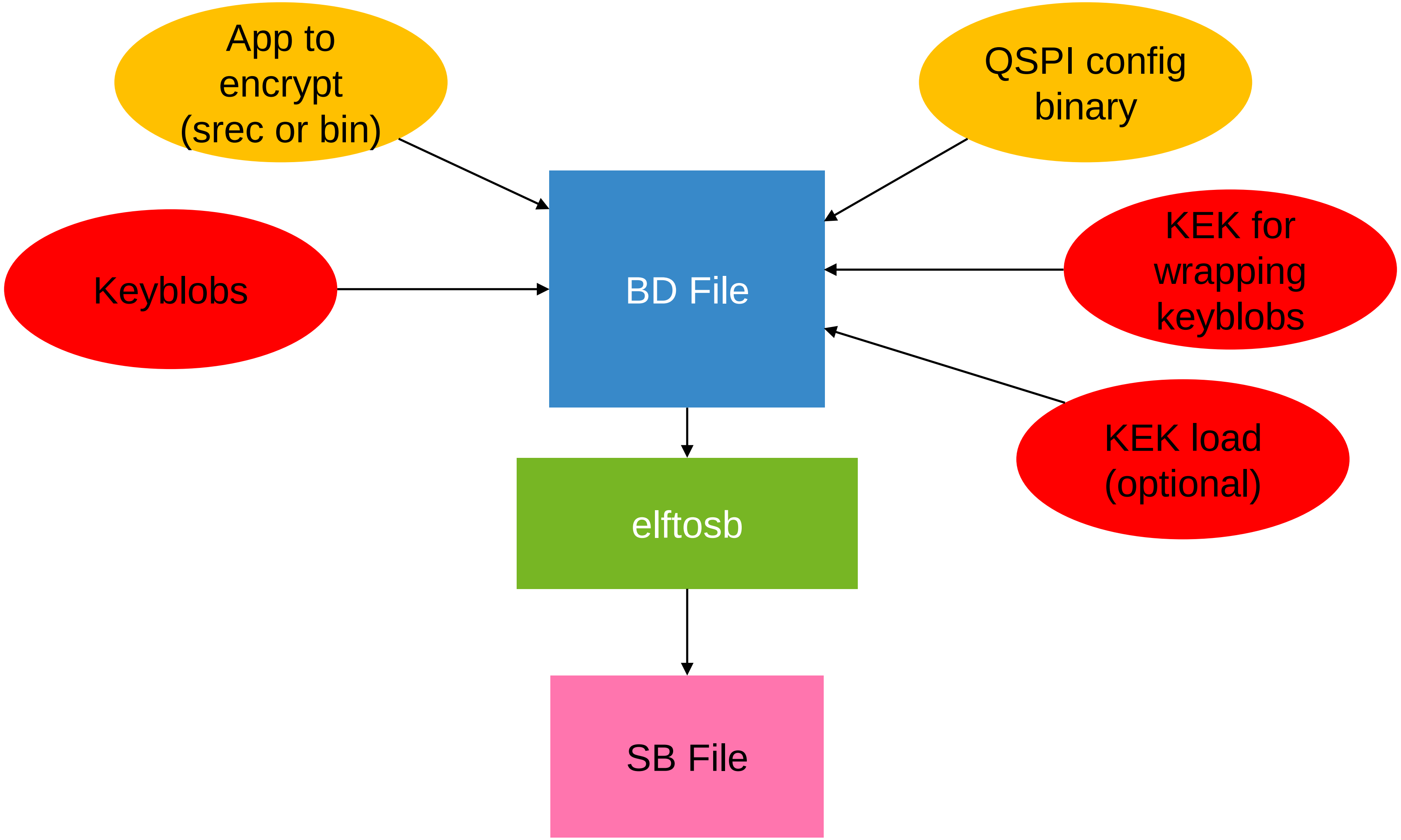
Encrypted Keyblob(s)*

Serial NOR flash

Encrypted application

* If multiple keyblobs are used they must be contiguous

Using the ROM to Load and Configure OTFAD



App to encrypt (srec or bin)

QSPI config binary

Keyblobs

BD File

KEK for wrapping keyblobs

KEK load (optional)

elftosb

SB File

Hands-On

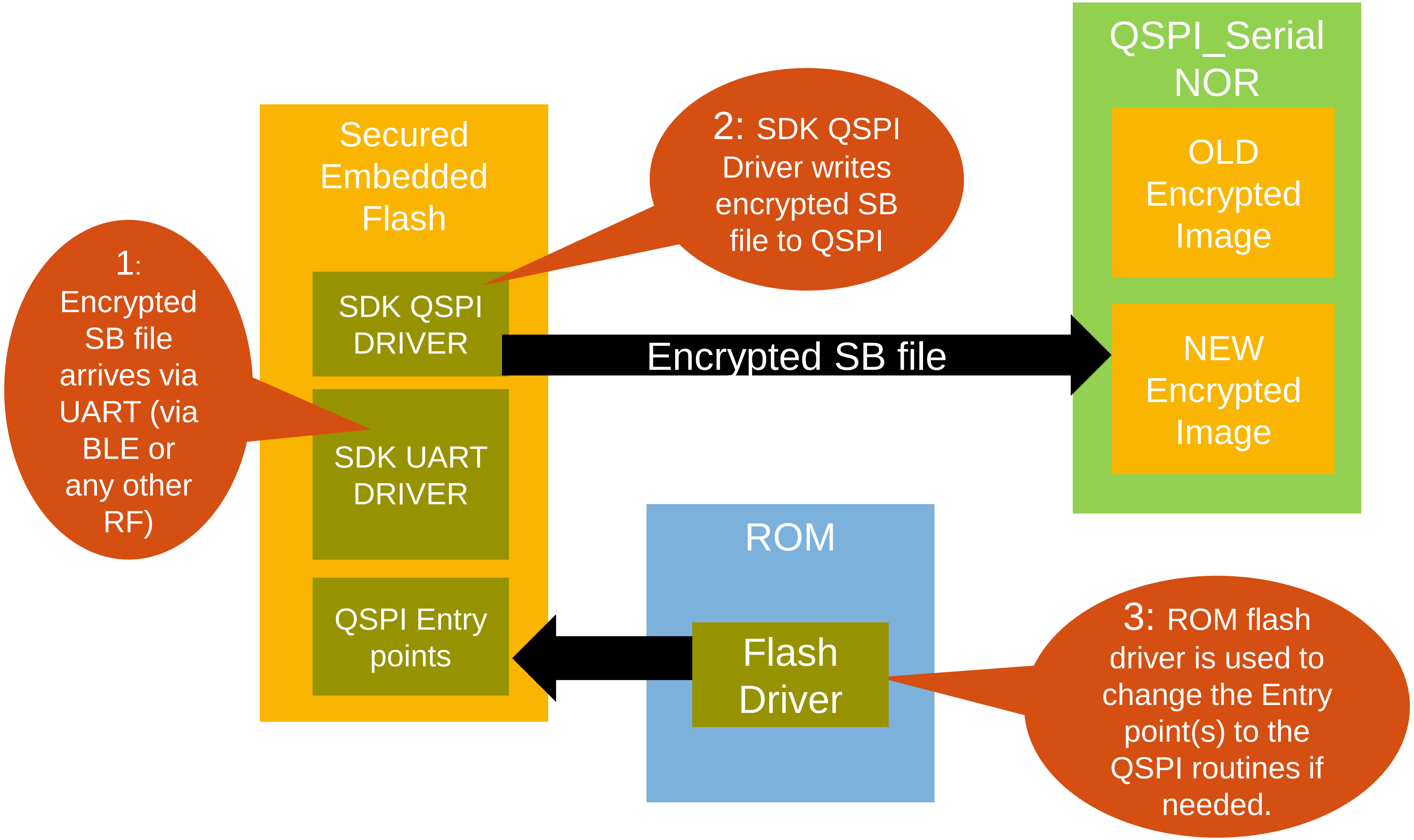
- Now it’s your turn!
- Follow the directions in the lab guide to load an encrypted application to the external serial NOR flash.

CUSTOMIZING YOUR FIRMWARE UPDATE FLOW

Other Options for Firmware Updates

- Today we’ve looked at the features built into the K8x ROM and how they can enable secure firmware updates
- Going through the ROM is not the only way to accomplish this
- If you code your own firmware update process into your application, you can open up additional options for:
  - Interface used to download new image
  - Changes to encryption algorithm used
  - Key management/storage options
- Because Kinetis bootloader and the ROM are built from the same code base, when the K8x support is added to the Kinetis bootloader (release planned by launch) you can use the bootloader code as a starting point and modify as needed.

Over-the-Air Update of Encrypted QSPI Application



1: Encrypted SB file arrives via UART (via BLE or any other RF)

Secured Embedded Flash

SDK QSPI DRIVER

SDK UART DRIVER

QSPI Entry points

2: SDK QSPI Driver writes encrypted SB file to QSPI

Encrypted SB file

QSPI_Serial NOR

OLD Encrypted Image

NEW Encrypted Image

ROM

Flash Driver

3: ROM flash driver is used to change the Entry point(s) to the QSPI routines if needed.

ENABLING SECURE DESIGNS

Resources

- **AN4507**: "*Using the Kinetis Security and Flash Protection Features*"
- **AN5112**: "*Using the Kinetis Flash Execute-only Access Control Feature*"
- **AN4307**: "*Using the mmCAU in Kinetis*"
  - **AN4307SW**: Example software for AN4307
- **MMCAU_SW_LIB**: CAU and mmCAU software library
- **CAUAPIUG**: CAU and mmCAU API User Guide
- **AN4733**: "*Using the DryIce Tamper Detection Unit on Kinetis Microcontrollers*" (available under NDA only)

K8x Hardware Options

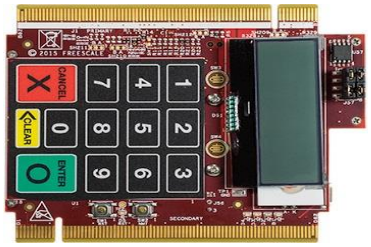
TWR-K80F150M Tower System Development Platform

The TWR-K80F150M development board is designed to work in standalone mode or as part of the NXP Tower System, a modular development board platform that enables rapid prototyping and tool re-use through reconfigurable hardware. Begin construction your Tower System evaluation board platform today and find additional Tower System boards and compatible peripherals at **nxp.com/Tower**.

FRDM-K82F NXP Freedom Development Platform

The FRDM-K82F NXP Freedom development boards are small, low-power, cost-effective evaluation and development platforms perfect for quick application prototyping and demonstration of Kinetis MCU families and NXP sensors. These evaluation boards offer and easy-to-use mass-storage device mode flash programmer, a virtual serial port and classic programming and run-control capabilities.

TWR-PoS-K81 Tower System Development Platform



The TWR-PoS-K81 development platform is a reference platform for a payment pin entry device. This board includes Cirque SecureSense AFE for secure pin entry. The design files and associated software show an example pin pad application that has been submitted for Payment Card Industry certification. The board is designed to work standalone or as part of the NXP Tower System.

SUMMARY

Summary

- In today’s connected world, security is important for protecting you and your customers.
- Firmware updates are incredibly likely and need to be planned for, and securing the firmware updates is important to protect IP, prevent product misuse, and protect customer data.
- K8x family includes new features to make security easier to use and faster, building on the security features that have already been offered on Kinetis devices.

Things You’ve Learned Today

- Need for security features for IOT applications and how Kinetis meets the challenge
- Seen how hardware acceleration affects crypto performance
- How Kinetis Anti-Tamper features can be used to protect an application
- How to use ROM features in K8x family for encrypted firmware downloads to external serial NOR flash
- How to use elftosb and blhost tools with the ROM
- Kinetis bootloader can be a software building block for implementing other secure firmware update methods

NXP

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT