Understanding the Principles of a Secure Development Workflow

Chris Jones
Senior FAE, Secure Thingz

Key Points

- The Chain of Trust is anchored directly in hardware
- Software can be protected using development tools that leverage the Chain of Trust

Key Elements of Security

- Development process flow
- Trust anchors
- Root of Trust
- Identity
- MCU requirements
- Immutable bootloader
- Secure Boot Manager
- Secure key storage
- Chip vendor security support
- Establishing the Chain of Trust
- Secure manufacturing
- Provisioning
- Secure provisioning
- Mastering
- Image signing
- Image encryption
- Image deployment
- Image final delivery
- Updates
- “Zero Trust” philosophy

Development Process Flow

- When developing a new IoT product, it's important to identify:
  - What trust anchors are available during software development by …
    - … supporting the formation of the product identity
    - … creating a robust Root of Trust
    - … establishing a Public Key Infrastructure (PKI) and issuing the required certificates
  - How a secure manufacturing / programming process will be realised by …
    - … secure provisioning during manufacturing
    - … avoiding key leakage
    - … protecting product identities against overproduction
    - … preventing the theft of software IP
  - How security will continue to be enforced after the product has been manufactured by …
    - … enabling secure software updates
    - … ensuring the system integrity
    - … protecting customer data

Trust Anchors

- Defining the meaning of **Trust Anchor** in the context of IoT product development:
  - In terms of certificates and PKI, a public key and associated data used by a relying party to validate a signature on a signed object. It is commonly referred to as a Root Certificate but has only local significance.
  - In terms of hardware, a combination of technologies, both hardware and software that form an implicitly trusted platform. This platform is commonly referred to as the Root of Trust (RoT).

Root of Trust

- Requirements that must be addressed in order to establish a **Root of Trust** in an IoT product:
  - **Unique Product Keys**:
    - Must be setup / provisioned, immutable and protected.

  - **Unique Product Identity**:
    - Can be verified using cryptographic means

  - **Authentication**:
    - Can be done by immutable cryptographic method

  - **Platform Integrity**:
    - Immutable boot path to a RoT Boot Manager that verifies software

Identity

- A unique, strong device **identity** is crucial, it addresses important requirements in a secure IoT ecosystem:

– **Trust**: A device must be able to authenticate itself and securely communicate encrypted data and information.

– **Privacy**: Personal, sensitive and financial information must be kept secure. A strong identity can ensure communications are encrypted so that data exchanged remains private.

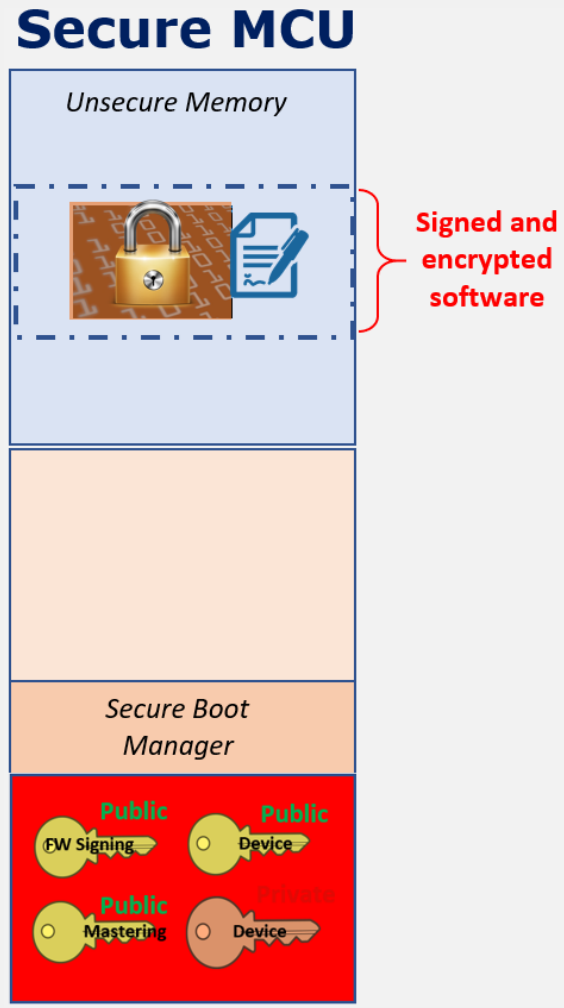
– **Safety**: Malicious attacks in industrial and medical environments can put the consumer/user at risk.

– **Integrity**: A device must prove that it is what it says it is, to ensure that device software and firmware are legitimate. This applies to both the device itself and the data it transmits.

Microcontroller (MCU) Requirements

- Key **MCU requirements** to establish a secure RoT:

1. Secure Chain of Trust (CoT) to a boot manager. An *immutable boot path* that cannot be interrupted by the debug / JTAG interface.
2. Ability to program and lock a section of boot flash memory so that it is immutable.
3. A small isolated memory area (flash or fuse) for secure storage of Security Content (keys) is needed. This memory area must *ONLY* be accessible by a Secure Boot Manager (SBM).



Immutable Bootloader

- Essential in forming the Chain of Trust
  - From the reset vector of the MCU to the bootloader, nothing can be altered or changed
  - Each loader verifies the next stage code image before execution.

- Some MCUs contain a built-in 1st stage secure bootloader
  - Used to decrypt and/or check the integrity of the boot image before the MCU boots.
  - Can be used the verify and execute a 2nd stage loader in order to improve the boot process.
  - May also enable a secure SW programming process. Pre-provisioned secure keys (in the MCU key store area) are used in order to facilitate a secure SW loading process.

Secure Boot Manager

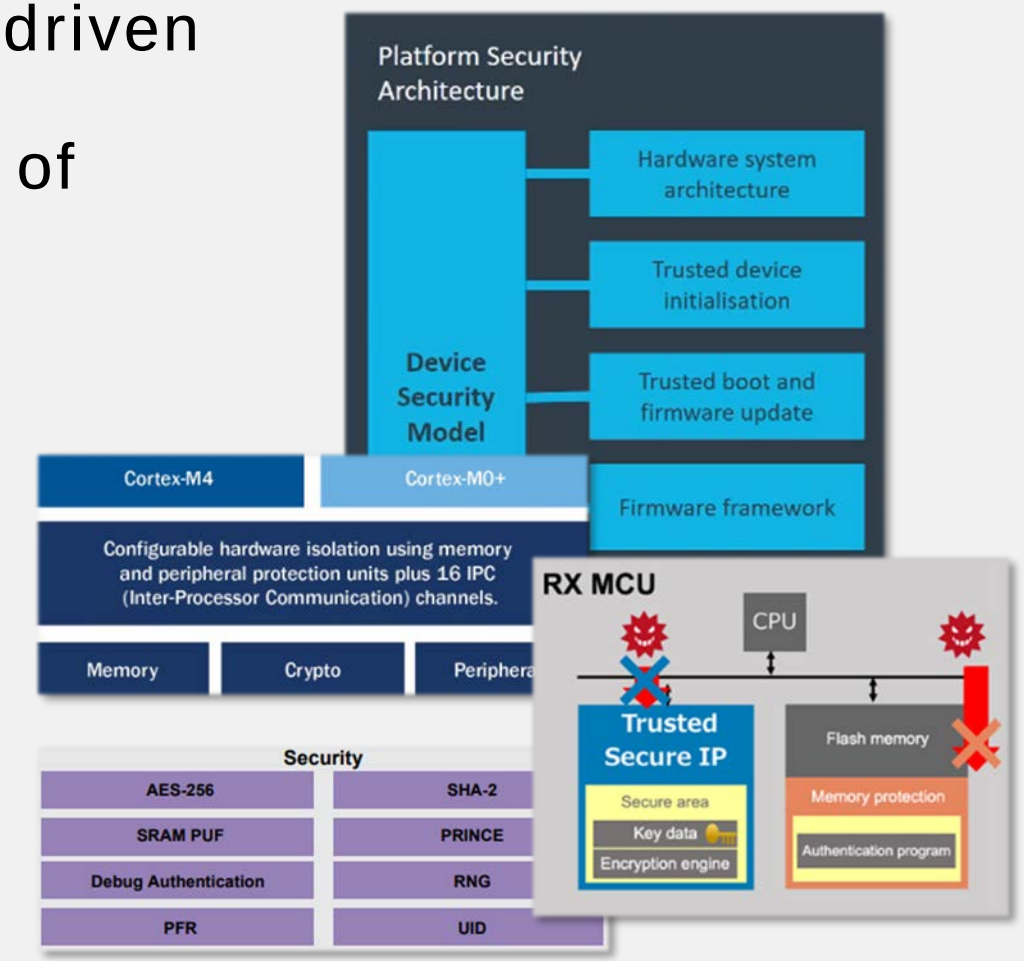
- A **Secure Boot Manager** (SBM) is firmware that forms part of the Root of Trust in the IoT product. The SBM is programmed into the bare metal MCU during the provisioning process. Features required:

  - Check integrity of the SBM and data/secrets/RoT installed during provisioning
  - Ensure secrets & keys (private & public) are held in secure memory
  - Verify integrity of application signatures prior to running
  - Check for pending application updates and validate signatures prior to installation
  - Manage a stable shutdown procedure if there is an application launch failure
  - Provide APIs to enable monitoring of SBM status, basic security functions to be performed and allow software updates

Chip Vendor Security Support

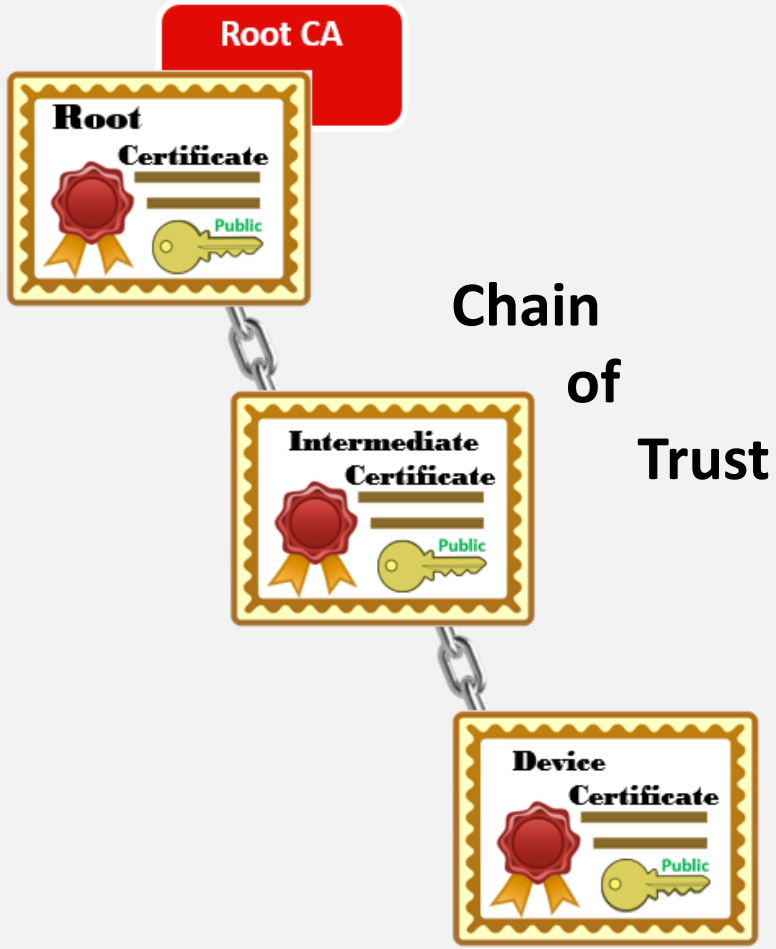
- Semiconductor manufacturers are being driven by industry to develop architectures and peripherals that enable the development of secure applications.

- Examples of secure microcontrollers/technology:
  - ARM®v8-M architecture
    - NXP LPC55S6x (ARM M33 core)
    - Microchip SAM L11 (ARM M23 core)
  - STMicroelectronics (ARM M7 core)
  - Renesas Trusted Secure IP Module (RX core)
  - Cypress PSoC6 (ARM M4 & ARM M0+ cores)



Establishing the Chain of Trust

- With an immutable SBM and a secure memory we can implement the RoT.
- The RoT allows us to take advantage of the PKI and create a **Chain of Trust** (CoT)
  - The CoT can be created using a two-staged PKI
  - The Root CA is operated by the OEM
  - The OEM has the option to create different Intermediate CAs for different kinds of devices.
  - Each device gets an individual device certificate issued by an Intermediate CA of the OEM
  - The Device Certificate creates identity by tying the IoT device unique ID to a Public key.



Development Summary

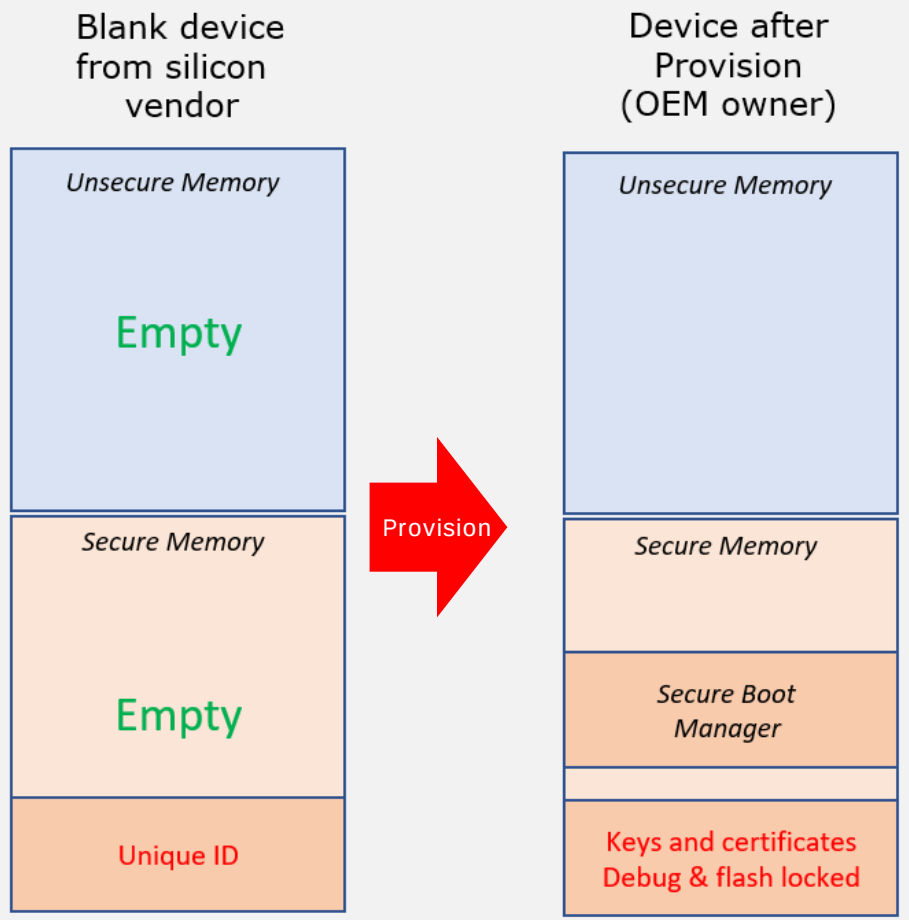
- The creation of a **Trust Anchor** is critical to the development of an IoT product
- Microcontroller architectures are being developed to allow engineers to implement hardware **Roots of Trust**:
  - ARM Platform Security Architecture
- **Secure MCUs** are becoming more readily available
  - Microchip SAM L11, NXP LPC55S6x
    - Arm Cortex M23 and M33 cores
  - Renesas RX (TSIP)
  - STMicroelectronics STM32H7
  - Cypress PSoC6
- A Trust Anchor allows the IoT product to utilise a PKI architecture and develop a **Chain of Trust**

Secure Manufacturing

- The establishment of a Root of Trust includes both hardware and software
  - How can we be sure that the secure device has been manufactured securely?
  - Is the firmware safe or can it be stolen?
  - Who else will have access to the private keys being programmed into the Secure MCUs?
- A “zero trust” philosophy is recommended to help ensure that your IP is safe and that your device is securely manufactured

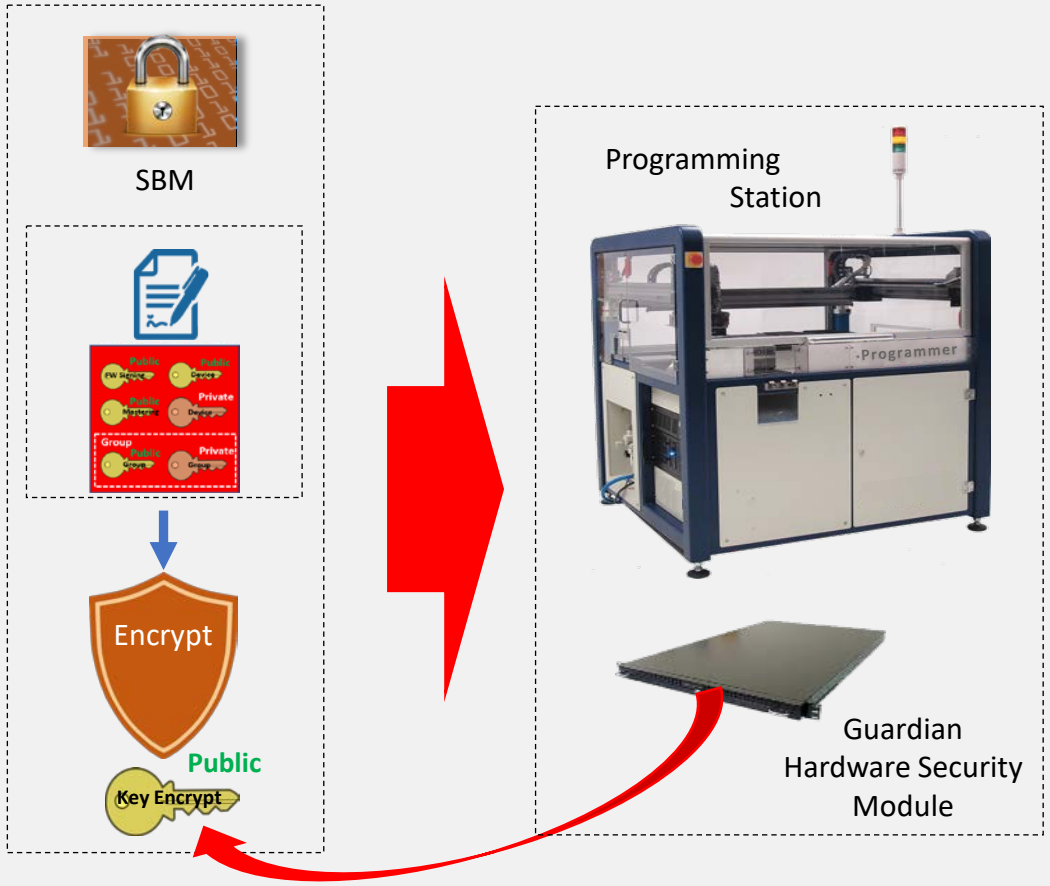
Provisioning

- **Provisioning** is the process of establishing the Root of Trust of the secure MCU by programming.
- The provisioning process is usually undertaken by a programming centre or contract manufacturer.
- For some OEMs, they may choose to provision the RoT directly.
- It is important that a Secure Programming Centre is used as OEM/Customers' private keys, secret information and IP is required to be handled by third parties in order to provision their devices.



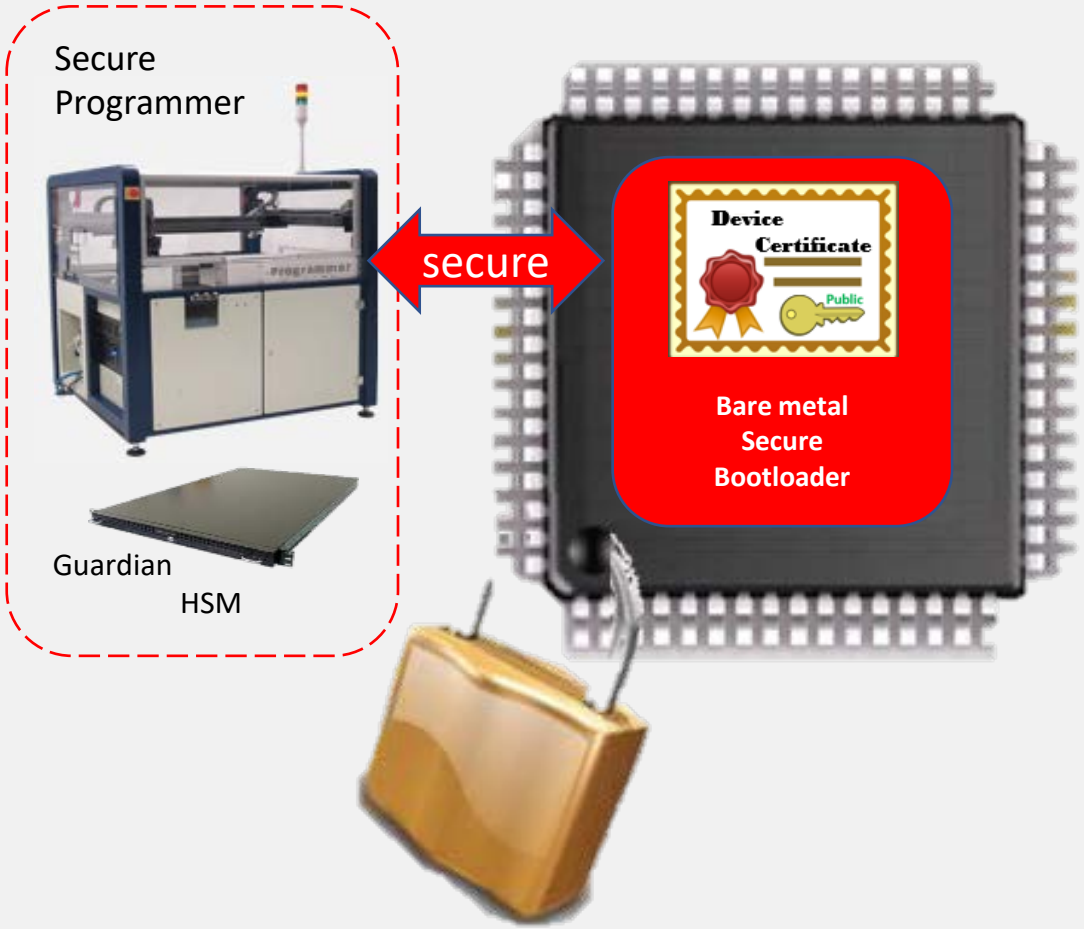
Secure Provisioning

- **Secure Provisioning** services utilise the functions of Hardware Security Modules (HSMs).
- HSMs allow the secure transportation of secrets, firmware, configuration and production information directly to an MCU programming station.
- Many distributors are now providing these secure programming services that incorporate HSMs.



Secure Provisioning

- Currently many secure MCUs have open channel provisioning access
  - JTAG
  - SWD
- More advanced secure MCUs include cryptographic key pairs incorporated into the bare metal devices.



Mastering

Mastering happens at the end of the development process.

- The process of exchanging all keys and certificates used during the development process with production environment security keys and certificates, all in the context of the “zero trust” philosophy.

- Adds headers, signatures and encryption to enable the software image to be efficiently transferred over a number of potential mediums and into the Software Update Slot of the Secure Boot Manager (SBM).

Image Preparation

- A signature of the mastered software image is created using the OEMs signing (Private) key.

- The software image file is encrypted using an ephemeral symmetric session key which enables it to be transported freely over non-secure channels.

But how does the software encryption session key get to the destination?

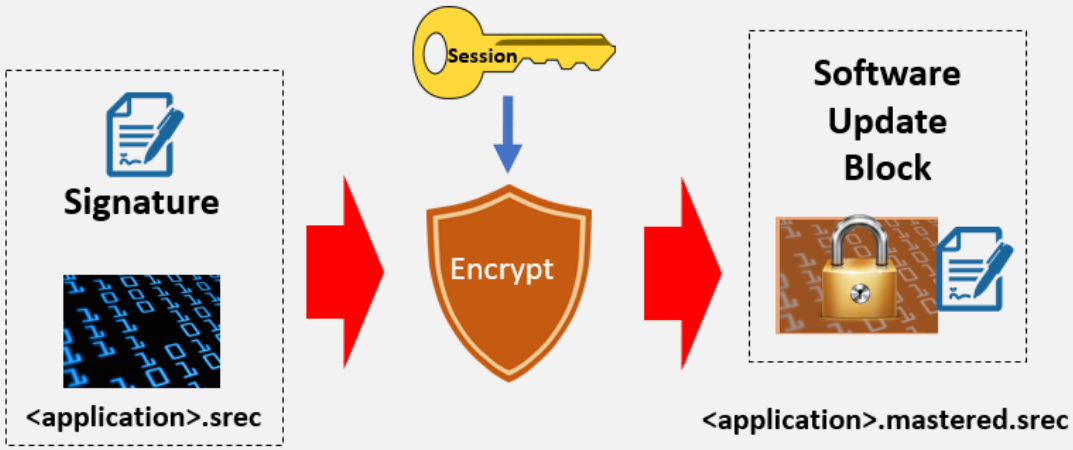


Image Deployment

- For remote mastering, the session key is encrypted using the remote Hardware Security Module (HSM) public key.
- The secure facilities' HSM public key can be retrieved from its certificate.

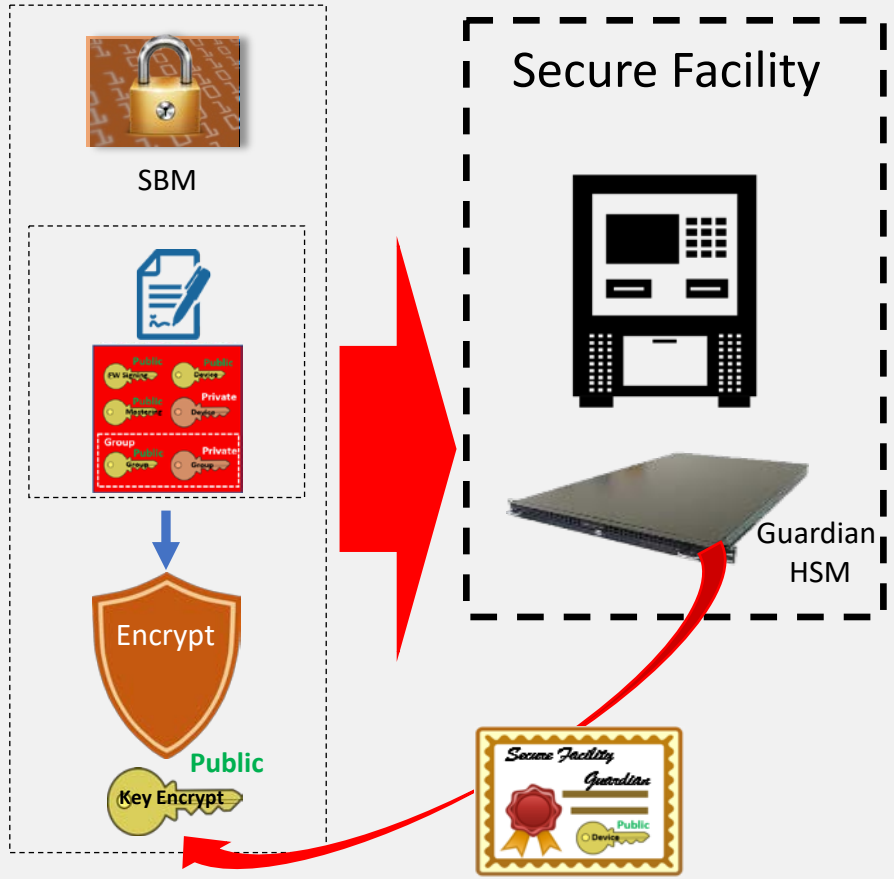
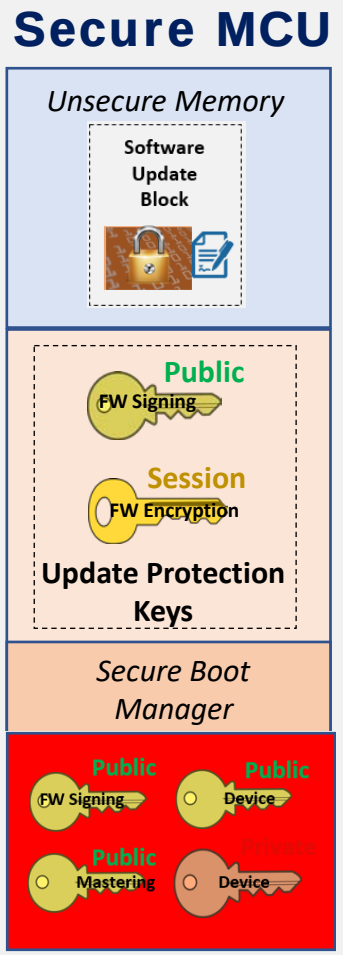


Image Final Delivery

- The encrypted software image and associated protection keys are programmed into the target device memory.

- The SBM will then:
  - Decrypt the secret keys using its private device key.
  - Check the authenticity of the secret keys using the mastering tools public key.
  - Check the authenticity of the software update block by using the OEM's public key.
  - Decrypt the software update block by using the session key.
  - Program the application software into flash.



Summary

- Semiconductor manufacturers are aware of their customers' requirements for secure features and are designing MCUs to meet those needs.
- MCUs with immutable boot and secure memory enable the Chain of Trust.
- PKIs deliver the elements essential for a secure and trusted environment.
- A "zero trust" philosophy is recommended when considering manufacturing partners in the supply chain for a secure device.

Key Points

- The Chain of Trust is anchored directly in hardware
- Software can be protected using development tools that leverage the Chain of Trust

Questions?

Thank you

Want to learn more?

Visit the Secure Thingz' table during the event today