BETA

Hardware Trust Introduction

Alessandro Savino

Politecnico
di Torino

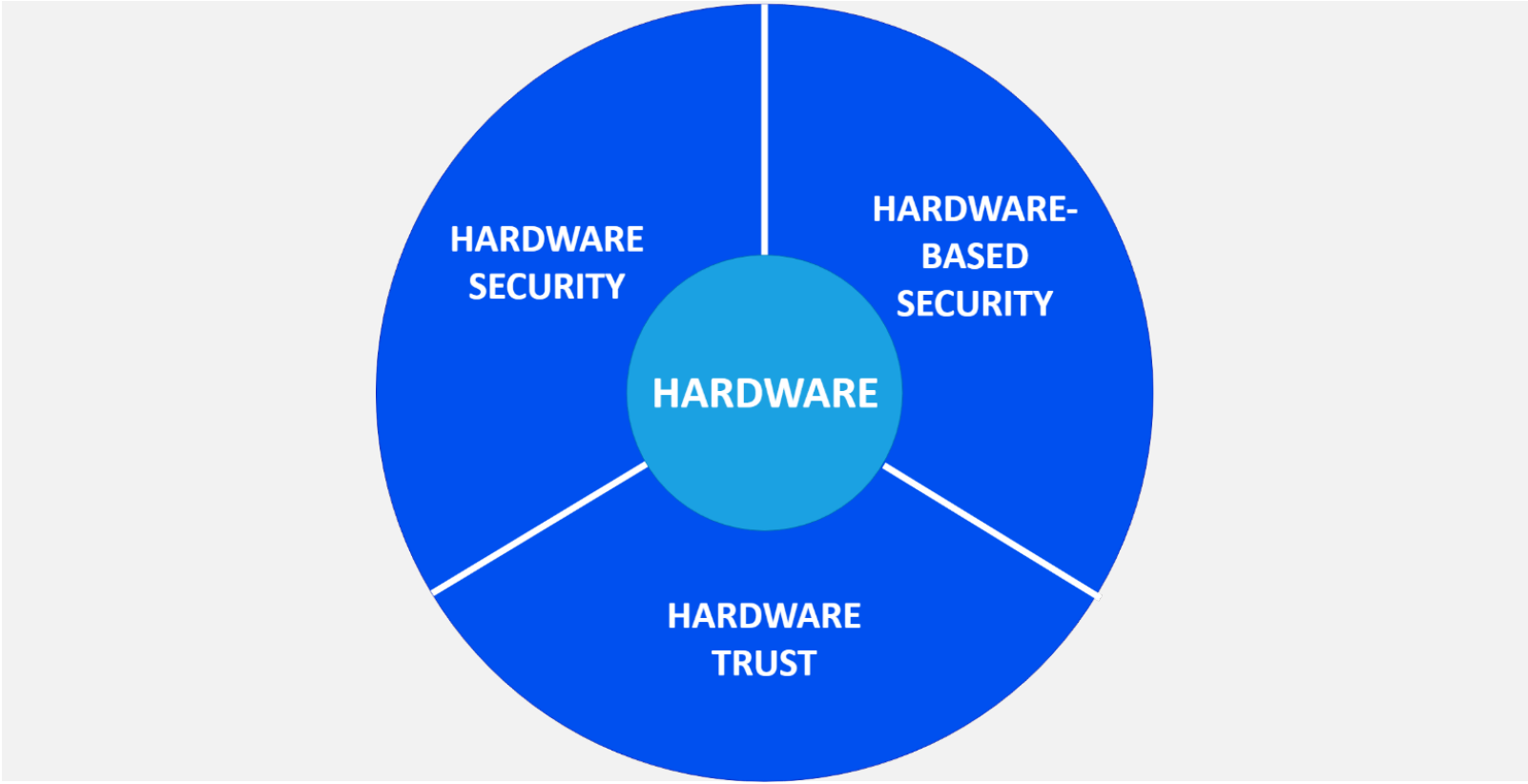
Acknowledgments

The presentation includes material from several contributors whose valuable help is acknowledged and highly appreciated.

› Nicolò MAUNERO

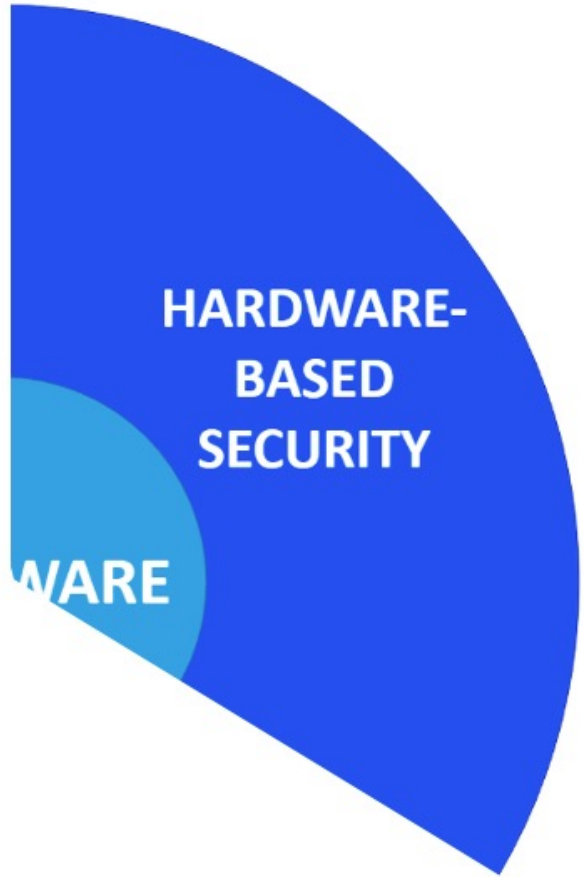
› Gianluca ROASCIO

The Role of Hardware in Cybersecurity



Hardware-based Security

Refers to all those solutions aimed at resorting to hardware devices to protect the system from attacks that exploit vulnerabilities of other system components.

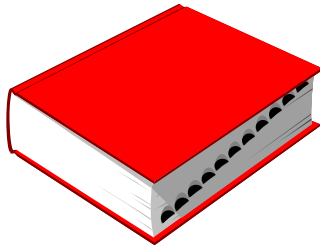


Trust

"A trusted component, operation, or process is one whose behavior is predictable under almost any operating condition and which is highly resistant to subversion by application software, virus, and a given level or physical interference."

[ISO/IEC]

Root of Trust



Component that needs to always behave in the expected manner because its misbehaviour cannot be detected

Root of Trust

Trust in the Roots of Trust can be achieved through a variety of means including technical evaluation by competent experts.

Root of Trust - Role

Is used as basic block for the construction of a **Chain of Trust**

Trust Anchor

”A public or symmetric key that is trusted because it is directly built into hardware or software, or securely provisioned via out-of-band means, rather than because it is vouched for by another trusted entity (e.g., in a public key certificate).”

[https://csrc.nist.gov/glossary/term/trust-anchor]

Hardware Trust Anchor

A Hardware Trust Anchor is a component that securely store and provide a unique secure identifier for the device.

Security & Trust Anchor Candidate

Since attack difficulty is at the highest with the hardware, it presents an excellent anchor for the compute security features

Hardware as the Root-of-Trust (RoT) Design Methodologies

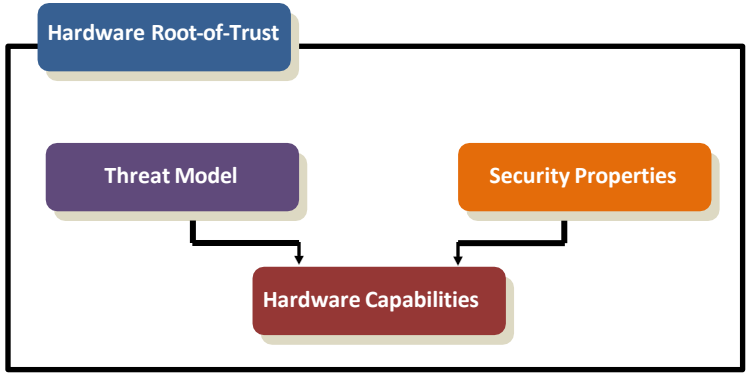
- NIST (NIST SP 1800-19B) defines Hardware Root-of-Trust as "An inherently trusted combination of hardware and firmware that maintains the integrity of information."
- Practically, Hardware Root-of-Trust (HRoT) is defined as the foundational building block(s) of different security schemes, protocols, products, or services within a secure computing system
- Formally, Hardware Root-of-Trust (HRoT) is an immutable hardware component or a set of hardware components (e.g., an encryption engine and/or a dedicated secure processor) considered unconditionally trusted against a well-defined threat model

Hardware Root-of-Trust Properties

Proof of authenticity and/or provenance
- E.g., Uniquely identifiable and verifiable features
  - Physically Unclonable functions (PUFs)

Immutable hardware component(s)
- E.g., Anti-tamper features

Anchor trust against a specific threat model



Roots of Trust in Trusted Computing

Root of Trust for Measurement (RTM)

- Initiates process of recording what software is running
- e.g., implement in an immutable or securely updateable component

Root of Trust for Storage (RTS)

- Implements shielded locations like registers with special integrity or confidentiality
- Implemented in a Trusted Platform Module (TPM)

Root of Trust for Reporting (RTR)

- Uses cryptography to give assurances to third parties
- Built from keys burned into TPM at manufacture time

System level solutions

We shall focus on two significant standards:

- Trusted Platform Module
- Trusted Execution Environments

Trusted Platform Module – TPM

Standard guideline for developing chips with strong cybersecurity features

Trustworthiness of TPM is based on different Root of Trust components and well-defined interactions among them

TPM History

Specification initially released by the *Trusted Computing Group* in 2003
[https://trustedcomputinggroup.org/]

The current version is TPM 2.0, which is standardized under ISO/IEC 11889
[https://www.iso.org/standard/66510.html]
[https://ebrary.net/24701/computer_science/a_practical_guide_to_tpm_20]

TPM types

There are five types of TPM:

- *Discrete,*
- *Integrated,*
- *Firmware,*
- *Software*, and
- *Virtual*.

Discrete TPM is the most common and the most secure form.

TPM adoption

TPM has become standard in many consumer-grade computers over the last few years

In February 2019, the TPM was recommended for securing high-risk industrial devices in the newly released international standard IEC 62443-4-2.

In the automotive industry, TPM is used to guard a vehicle's software

Hardware Root-of-Trust Usage Model

Security kernel

- Hardware, firmware, and software elements of a trusted computing base implementing the reference monitor concept
- Security kernel must mediate all accesses, be protected from modification, and be verifiable as correct

Trusted Computing Base (TCB)

- Every secure computing system must have some TCB
- Hardware and software necessary for enforcing all security rules
- Vulnerabilities in the TCB can jeopardize the security of the entire system
- Ideally
  - Rooted in hardware and small
  - Should be isolated from the rest of the computing components
  - Its correctness and runtime state should be easily and independently verifiable

Hardware RoT to support the implementation of the trusted computing base

Trusted Computing Base (TCB) Anchor

At the heart of the Trusted Computing Base (TCB) is the Trusted Platform Module (TPM)

- The TPM provides hardware-based authentication, integrity, and attestation to the TCB
- It is designed as a small tamper-resistant chip that provides the following functions
  - A root-of-trust for reporting and storage
  - Measurement and attestation of platform integrity
  - Platform identification and authentication
  - Core and highly constrained cryptographic functions

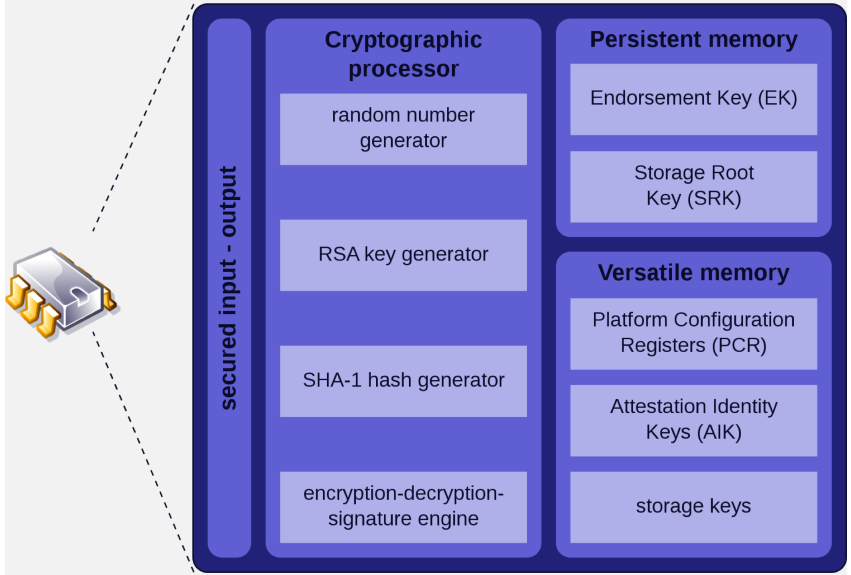
Trusted Computing Base (TCB) Anchor

Trusted Platform Module (TPM) architectures

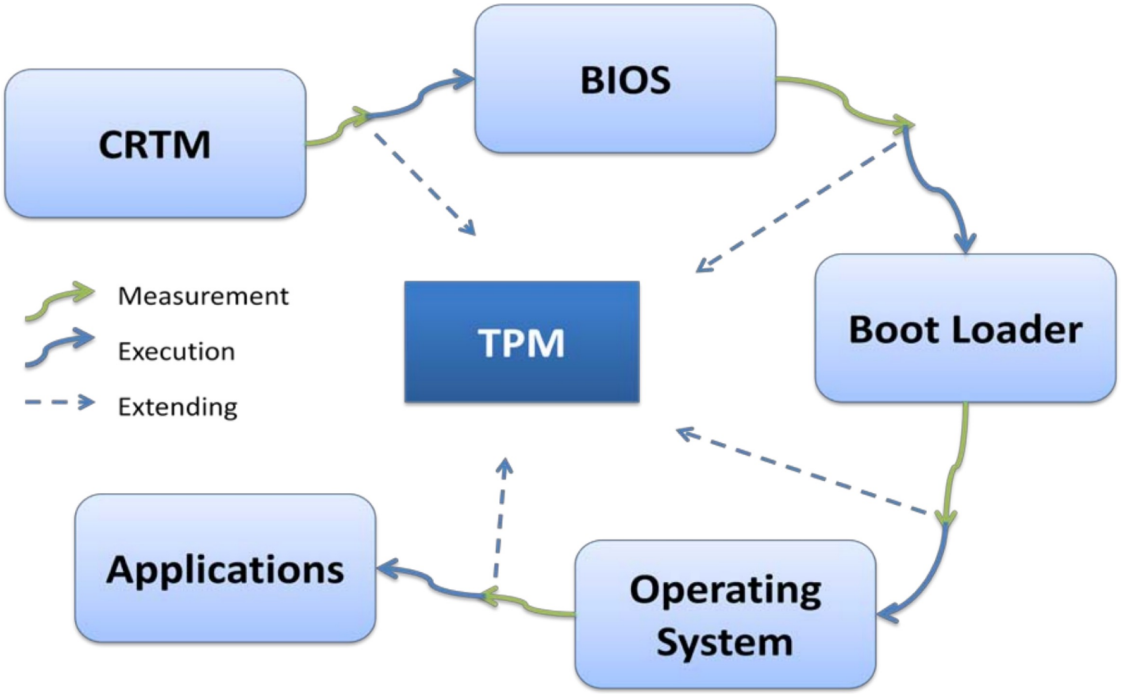
- Storage
- Random number generation
- Cryptographic function and processing

Trusted Platform Module (TPM) types

- Discrete TPMs
- Integrated TPMs
- Firmware TPMs (fTPMs)
- Hypervisor TPMs (vTPMs)
- Software TPMs



Trusted Computing Base (TCB) Anchor



TPM Basic Features

They include, among the others:

- Secure Boot & Firmware Integrity
- Certification
- Attestation and Autentication
- Protected Location
- Integrity Measurments and Reporting

Secure Boot & Firmware Integrity

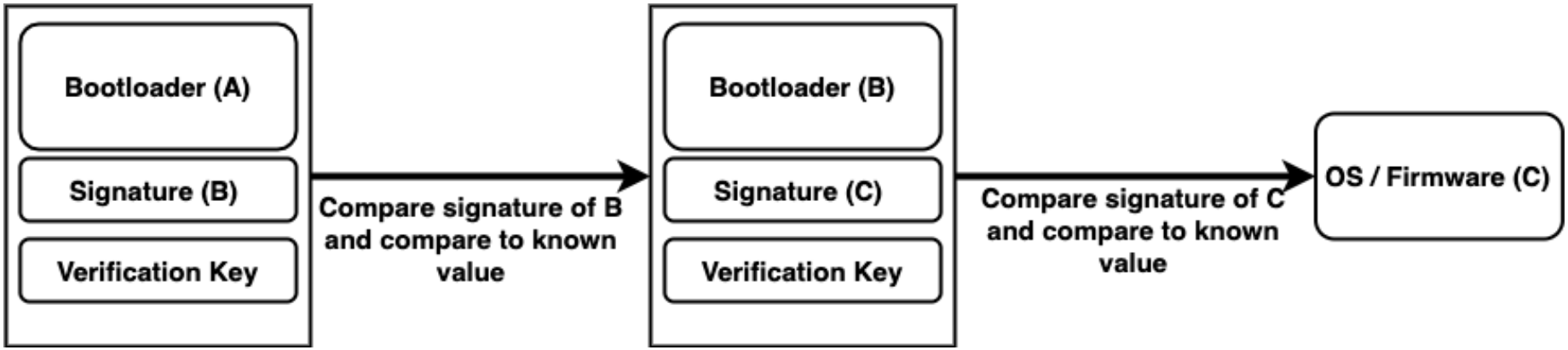
The environment in which the code runs must be controlled

A power-on reset creates an environment in which the platform is in a well-known initial state

Secure Boot is the act of establishing a secure initial state

Secure Boot & Firmware Integrity

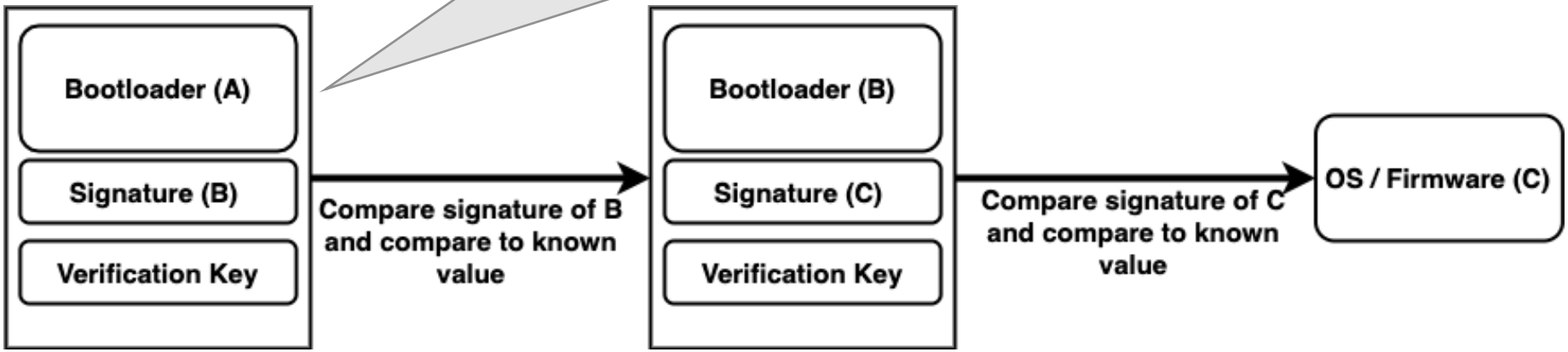
The typical secure boot method verifies the authenticity of each component in the boot chain:



Secure Boot & Firmware Integrity

The typical secure … each component in the boot chain …

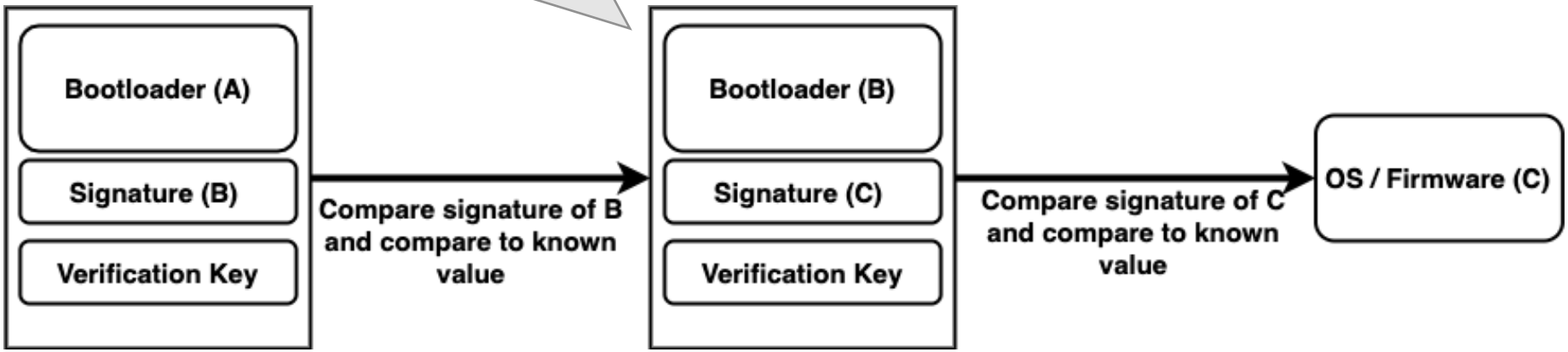
A 1st protected bootloader (A), stored in a secure memory, verifies the integrity and authenticity of a 2nd bootloader (B)



Secure Boot & Firmware Integrity

The typic … of each component in the bo…

The 2nd bootloader (B) verifies the integrity and authenticity of the Operating System kernel and of the Firmware



Secure Boot & Firmware Integrity

Typically, the 1st bootloader (A) cannot be modified, whereas the 2nd bootloader (B) can be updated

Certification

A certificate of authenticity should be available for the key shipped with the TPM

- It can be used to associate credentials (certificate) with other TPMs
- A certified key that can be used for signing may be used to attest the platform data that affect the integrity (trustworthiness) of a platform

Certification

Certificate and authentication credential can be stored in a Root of Trust for Storage (RTS) element

Attestation and Autentication

Root of Trust components are usually the entities trusted when attesting to a devices

- Unique identifiers can be stored in a Root of Trust element and used to identify the system
- A unique identifier can be obtained by resorting to **Physically Unclonable Functions** (PUFs)

Attestation and Autentication

Trusted platforms employ a hierarchy of attestation

Extenal entities attest for different charateristics of the TPM

- Genuine and compliant with the standard
- Contains a RTM and exists a trusted path between the RTM and the TPM
- A key pair is protected by a genuine TPM
- …

These various attestation takes the form of keys, certificates, software signature, etc. that are stored inside the TPM

Protected Location

All information on a TPM is in a Shielded Location

- The contents of a Shielded Location are not disclosed unless intended: only the allowed entities can access the secure memory and the Root of Trust functionalities
- When sensitive data are not stored in a Shielded Location on the TPM, they are encrypted

Protected Location

Wherever sensitive data are stored outside TPM, it is in a Protected Location

- Encryption of Protected Locations uses multiple seeds and keys that never leave the TPM
- Tamper resistance devices are used to avoid disclosing sensitive information to common physical attacks.

Integrity Measurments and Reporting

An integrity measurement is a value that represents a possible change in the trust state of the platform

The measured object may be:

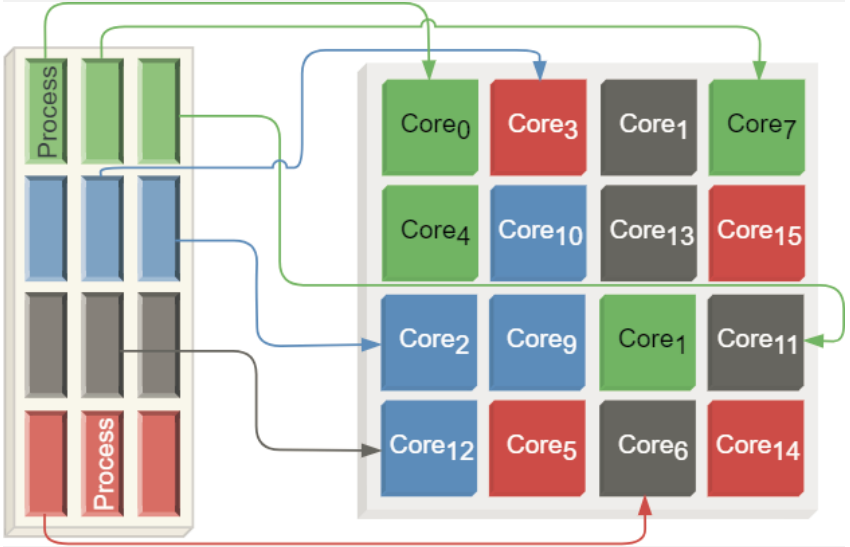
- A data value
- The hash of code or data

The digest of an arbitrary set of integrity measurements is statistically unique

Mixed Criticality Computing Systems

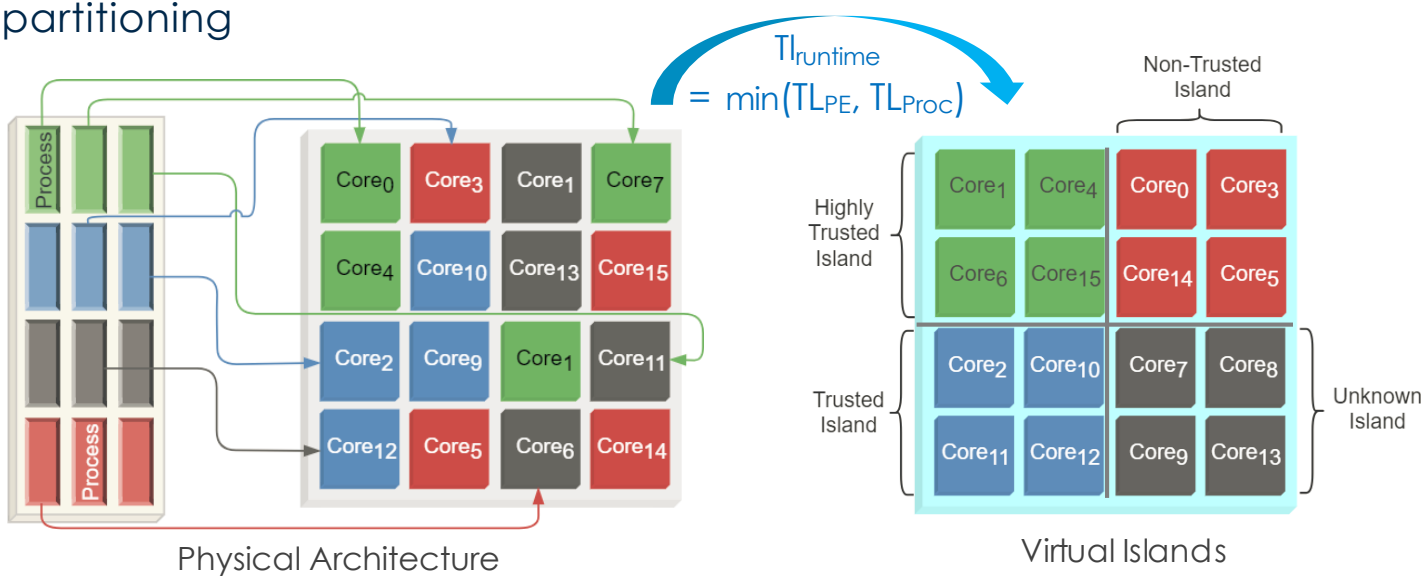
Current situation:

› Trusted/untrusted applications running on trusted/untrusted cores



Trusted Execution Aware Design

- Develop a new trust-aware architectural framework for integrating multiple heterogeneous IPs or tenants, secure to non-secure cores, in the same chip design
  - Hardware virtualization through trusted, non-trusted, and unknown island partitioning



Physical Architecture

Virtual Islands

Trusted Execution Environments

Trusted Execution Environments (TEEs) are secure systems-on-chip areas guaranteeing code and data protection.

- They typically offer the minimal security required by low-end, closed embedded systems, such as IoT and "bare-metal" (i.e., without any Operating System) solutions.

Trusted Execution Environment

TEE was originally an initiative of Global Platform to standardize a part of the processor as a trusted, secure part

- TEE has since evolved and covers, in general, the hardware modifications made to processors to provide isolation and attestation to software

Trusted Execution Environment

TEE is a concept that provides a secure area for the main processor

- "to provide end-to-end security by protecting the execution of authenticated code, confidentiality, authenticity, privacy, system integrity, and data access rights"
- [Global Platform Device Committee, "EE protection profile," version 1.2, Public Release, November 2014, Document Reference: GPD_SPE_021 https://csrc.nist.gov/publications/detail/fips/140/2/final]
- A known example is the ARM Trust Zone

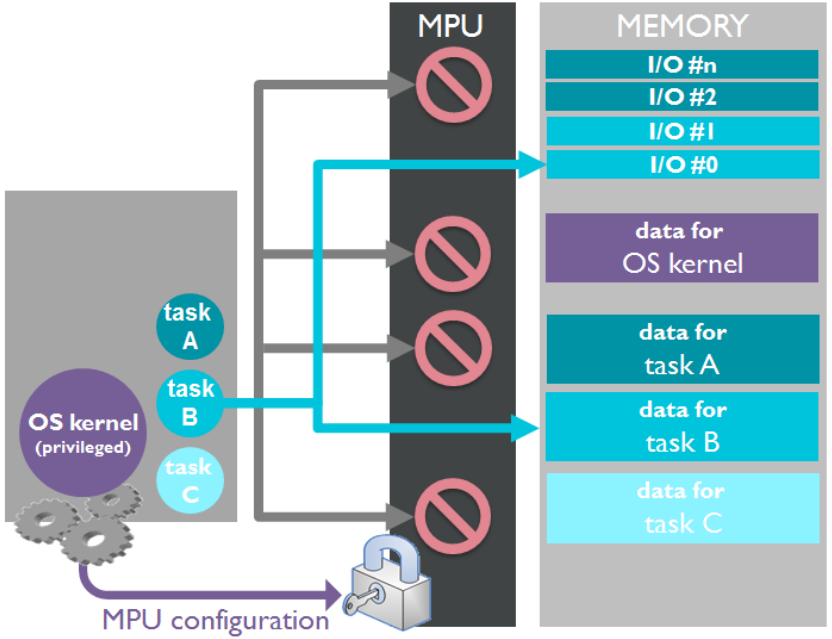
Architectural level solutions

General purpose **Design-for-Security** solutions adopted at the architectural level, mainly to improve the security of the CPUs and the involved memories.

Memory Protection Unit - MPU

Present in a wider and wider number of processors

- Each memory page can be read, written, or executed just by a predefined set of tasks/processes
- Access rights are decided by the kernel, which runs privileged
- The MPU automatically processes addresses sent to the memory without the intervention of the kernel
- Violations cause the immediate abortion of the task



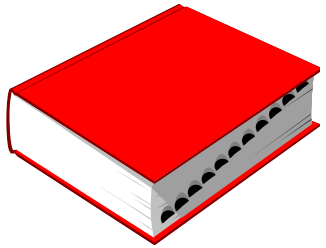
Device level solutions

A set of solutions was adopted at the device level to improve the device's resistance and resiliency to external attacks.

They include, among others:

- Tamper-evident devices
- Tamper-resistant devices

Tamper-evident devices



Devices that include some indicator of compromise, automatically activated when someone tries to mess with its physical integrity.

Examples of Tamper-evident devices

Physical Level

- Packaging should maximize the evidence of tampering
- Additional internal sensors (e.g., light detectors, temperature sensors, …) could be inserted to detect the presence of laser rays used to perform fault injection attacks

Waksman, Adam, and Simha Sethumadhavan. "Tamper evident microprocessors." In *2010 IEEE Symposium on Security and Privacy*, pp. 173-188. IEEE, 2010.

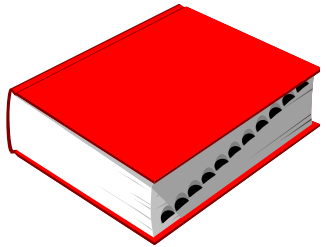
Examples of Tamper-evident devices

Software Level

- Appropriate auditing mechanisms through logging procedures to trace conducted activities and their sources

Waksman, Adam, and Simha Sethumadhavan. "Tamper evident microprocessors." In *2010 IEEE Symposium on Security and Privacy*, pp. 173-188. IEEE, 2010.

Tamper-resistant devices



Device properly engineered in a way to reduce the surface for physical attacks

Examples of Tamper-resistant devices

Strengthening of the hardware physical shielding

- A device built in a way that an attempt of decapsulation will damage or destroy the entire chip (3D SiP)
- Use of stronger coating for device packaging
- Additional metallization layer on top of the actual circuit to prevent micro probing attacks
- The circuit board needs to be dipped in special material (like epoxy) to prevent easy access to the hardware

Ravi, Srivaths, Anand Raghunathan, and Srimat Chakradhar. "Tamper resistance mechanisms for secure embedded systems." In *17th International Conference on VLSI Design. Proceedings.*, pp. 605-611. IEEE, 2004.

Examples of Tamper-resistant devices

Counterfeiting protections

› Hiding device's names and their serial numbers from the packages in order to make more difficult for an attacker to exploit known vulnerabilities of the target devices

Ravi, Srivaths, Anand Raghunathan, and Srimat Chakradhar. "Tamper resistance mechanisms for secure embedded systems." In *17th International Conference on VLSI Design. Proceedings.*, pp. 605-611. IEEE, 2004.

Implementations

Security-oriented components

Set of custom, special purpose components used for performing specific security-oriented operations, including:

- Hardware Cyphers
- Smart Cards & SIM Cards
- Secure storage devices
- Random Number Generators

Hardware Cyphers

Custom devices used to assist or replace software in cryptographic operations

Concerning software implementation, are

- Faster
- Less prone to exploitation than software

It can be isolated from the main processor

- Only a small subset of entities can access their functionalities
- Even if the main system is compromised, the integrity of hardware ciphers can be guaranteed

Hardware Cyphers

As with software also in Hardware, there are different cipher implementations

These implementations follow the software algorithms' counterpart

- Symmetric Key Encryption Algorithms
- Hash Algorithms
- Public Key Algorithms

Smart Cards

Device providing different security solutions, like authentication mechanisms based on something the user has

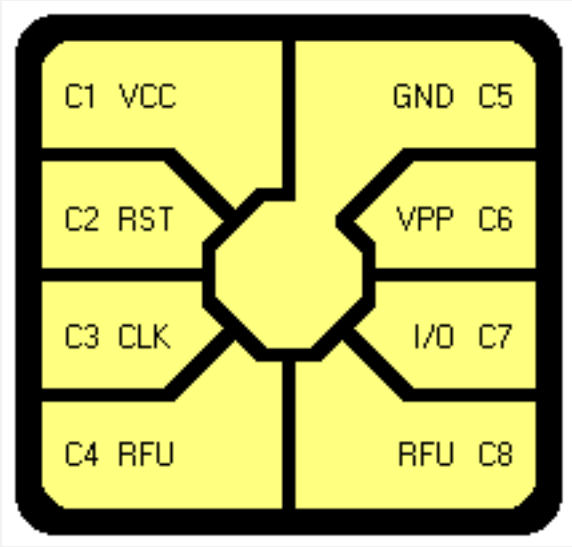
Smart Cards

They consist of a CPU linked to an I/O system

They can provide:

- A set of hardware-accelerated cryptographic algorithms
- Public keys and secret keys
- Secure key generation and storage

ISO 7816 Standards



SIM Card

A *Subscriber Identity Module* or *Subscriber Identification Module* (SIM), widely known as a *SIM card*, is a type of Smart Card

Secure Storage Devices

Securing stored data is a prime concern in many applications

A secure storage device must be designed and manufactured in a way that defines a certain level of protection:

- Tamper-evident devices
- Tamper-resistant devices
- Secure authentication mechanism
- Drive's controller security
- Data encryption
- Secure Erase

[Kaspersky Daily, "Is your encrypted USB drive secure?", https://www.kaspersky.com/blog/encrypted-usb-drives-audit/17948/]

Secure Authentication Mechanism

Hacking the authentication mechanism is far easier than breaking the underlying encryption mechanism

Four possible authentication mechanisms are

- **PIN pad**: can be subjected to a very simple exploit. Some buttons may present signs of usage, hence revealing the input combination
- **Software PIN input**: in this case, PIN must never be stored in software to mitigate replay attacks of eventual software vulnerabilities
- **Wireless badge**: they can be easily cloned
- **Fingerprint**: if the input system is not properly manufactured, it can be bypassed even without the need to fake the owner's fingerprint

Drive's Controller Security

The drive's controller must be developed to prevent unwanted access to the device

The drive's controller must present some protection against brute force attacks:

- Blocking after a series of unsuccessful authentication
- Deleting encryption keys and information stored in flash
- Avoid that passwords, PINs, or encryption keys can be requested to the drive's controller (trivial, but it happens)

Secure Erase

Need for proper data sanitization techniques

- It is not only crucial when a storage device is stolen but also in case of device disposal

An erase operation usually consists of overwriting all data with a given value, either 0 or 1

In solid state drive, data are stored as an electric charge

- A simple overwrite operation may leave some residual charge that, by using appropriate measurement equipment, can be measured to retrieve old data
- Multiple write operations are needed to completely remove any correlation between residual charges and old stored data

Vatajelu, Elena Ioana, Hassen Aziza, and Cristian Zambelli. "Nonvolatile memories: Present and future challenges." In *2014 9th International Design and Test Symposium (IDT)*, pp. 61-66. IEEE, 2014.

Politecnico di Torino

Proprietary Solutions

Proprietary Solutions

Intel® vPro® Platform

AMD Secure Technology™

ARM® TrustZone®

Microsoft BitLocker

Synopsys DesignWare® tRoot™

Apple Secure Enclave Processor

Google Titan

Cisco® Trust Anchor

Intel vPro® Platform

Present in modern Intel CPU, born in 2006 and refined through the years

It aims to provide all the functionalities a modern companies may need

The platform is a superset of underlying products and technologies:

- **Performance**: the top end of Intel's processor product line with high speed wired and wireless networking
- **Manageability**: platform features Intel® Active Management Technology (Intel® AMT) which provides full OS-independent remote control of endpoints over wired or wireless connections.
- **Stability**: It aims to stabilize key system components for 15 months or until the next platform release. This helps a business avoid network or software compatibility problems that may arise when deploying less stable computing infrastructure.
- **Security Features**: hardware-enhanced security features that help protect all layers in the computing stack.

https://www.intel.it/content/www/it/it/architecture-and-technology/vpro/vpro-platform-general.html

Intel vPro® – Security Features

Intel® Hardware Shield provides enhanced protections against attacks below the OS and advanced threat detection capabilities for increased platform security

Hardware Shield includes:

- Intel Runtime BIOS resilience: locks BIOS at runtime, preventing unauthorized access
- Intel Trusted Execution Technology (Intel TXT)
- Intel Virtualization Technology: hardware support for processor virtualization
- Intel Software Guard Extension (Intel SGX)

Intel® Software Guard Extension (Intel® SGX)

Costan, Victor, and Srinivas Devadas. "Intel SGX Explained." *IACR Cryptology ePrint Archive* 2016, no. 086 (2016): 1-118.

Intel's SGX is a set of extensions to the Intel architecture that aims to provide integrity and confidentiality guarantees to security-sensitive computation performed on a computer where all the privileged software (kernel, hypervisor, etc) is potentially malicious

SGX helps protect select code and data using hardware enclaves

The enclave is a secure container that only contains the private data in a computation and the code that operates on it

SGX's major hardware modification is the Memory Encryption Engine (MEE) that is added to the processor to protect SGX's Enclave memory against physical attacks

- It is used to guarantee data integrity and confidentiality during computation. The proof is a cryptographic signature that certifies the hash of the secure container's contents.

Intel® SGX – MME Overview

A modern processor has an internal cache that accommodates a small amount of memory and can be accessed much faster than the system memory

During normal operation, memory transactions are continuously issued by the processor's Core, and transactions that miss the cache are handled by the Memory Controller (MC)

The MEE operates as an extension of the MC, taking over the cache-DRAM traffic that points to the "Protected" data region.

Read/write requests to the protected region are routed by the MC to the MEE that encrypts (decrypts) the data before sending (fetching) it to (from) the DRAM.

https://eprint.iacr.org/2016/204.pdf

Intel® SGX – Physical Memory Organization

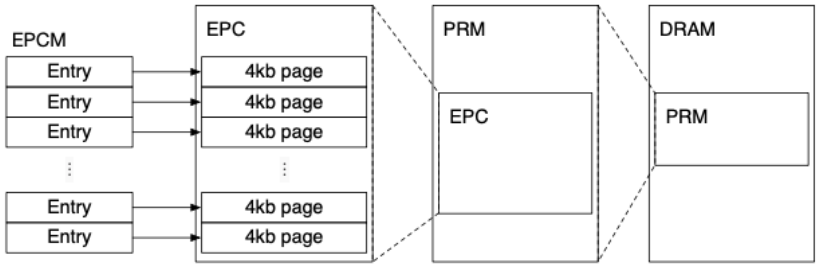
SGX sets aside a memory region called the Processor Reserved Memory (PRM)

The PRM holds the Enclave Page Cache (EPC) that stores enclave code

PRM is a subset of DRAM that cannot be directly accessed by other software e and data

SGX stores per-enclave metadata in an SGX Enclave Control Structure (SECS) associated with each enclave

SECS are stored in a dedicated EPC page



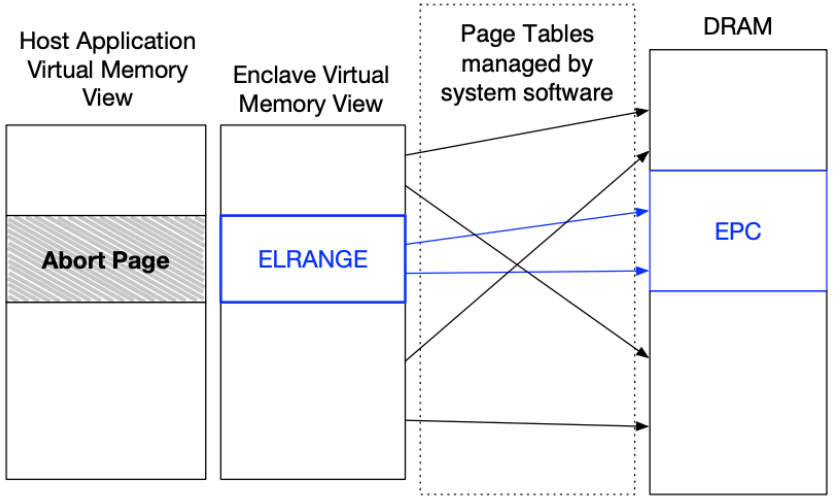
Costan, Victor, and Srinivas Devadas. "Intel SGX Explained." *IACR Cryptology ePrint Archive* 2016, no. 086 (2016): 1-118.

Intel® SGX – Memory Layout

Each enclave designates an area in its virtual address space, called the enclave linear address range (ELRANGE)

ELRANGE is used to map the code and the sensitive data stored in the enclave's EPC pages

Enclaves must store all their code and private data inside ELRANGE and must consider the memory outside ELRANGE to be an untrusted interface to the outside world.



Costan, Victor, and Srinivas Devadas. "Intel SGX Explained." *IACR Cryptology ePrint Archive* 2016, no. 086 (2016): 1-118.

Intel® SGX – Enclave Life Cycle

Creation: An enclave is born when the system software issues the ECREATE instruction, which turns a free EPC page into the SECS for the new enclave. ECREATE initializes the newly created SECS using the information in a non-EPC page owned by the system software.

Loading: ECREATE marks the newly created SECS as uninitialized. While an enclave's SECS is in this state, the system software can use EADD instructions to load the initial code and data into the enclave. EADD validates its inputs before modifying the newly allocated EPC page or its EPCM entry.

Intel® SGX – Enclave Life Cycle

Initialization: After loading the initial code and data pages into the enclave, the system software must use a Launch Enclave (LE) to obtain an EINIT Token Structure. The token is then provided to the EINIT instruction, which marks the enclave's SECS as initialized.

- The LE is a privileged enclave provided by Intel and is a prerequisite for using enclaves authored by parties other than Intel.
- The LE is an SGX enclave, so it must be created, loaded, and initialized using the same process as other enclaves.
- However, the LE is cryptographically signed with a special Intel key that is hard-coded into the SGX implementation, and that causes EINIT to initialize the LE without checking for a valid EINIT Token Structure.

Intel® SGX – Enclave Life Cycle

Teardown: After the enclave has done the computation it was designed to perform, the system software executes the **EREMOVE** instruction to deallocate the EPC pages used by the enclave.

- Before freeing up the page, EREMOVE ensures that no logical processor is executing code inside the enclave that owns the page to be removed.
- An enclave is destroyed when the EPC page holding its SECS is freed.
- An enclave's SECS page can only be deallocated after all the pages have been deallocated.

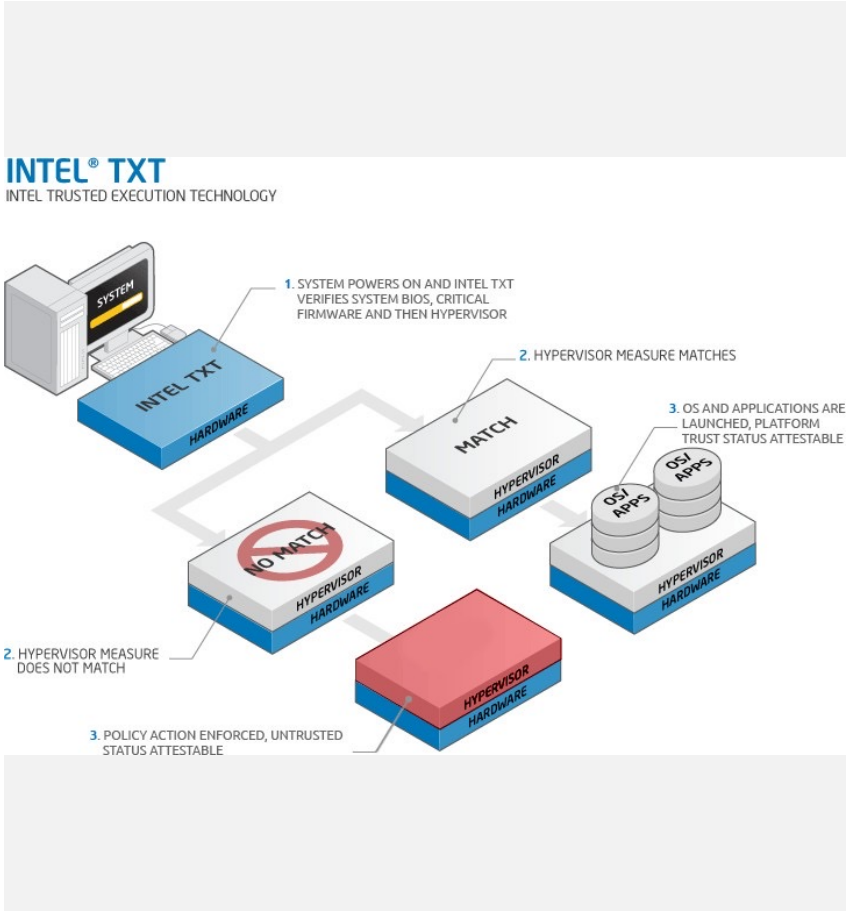
Intel® Trusted Execution Technologies (Intel® TXT)

Provide a chain of trust that is rooted in the microprocessor's hardware [1]

Its main purpose is to notify the user and system software of a possible attack and prevent a verified launch if an attack is detected

Can defend against:

- BIOS Attacks [2]
- Reset attacks [3]
- Rootkits [4]



Intel TXT - Functionalities

A chain of trust is extended from the Intel processor hardware through the BIOS

At the OS level if a user wants to enter secure mode, a secure boot of the software is initiated by the OS

A chain of trust can be extended from the hardware to the highest level of software

This chain of trust always performs a secure boot sequence before executing any components

Measurements: keywords indicating a secure boot sequence

- The integrity of component is checked (e.g. , code hash is verified)
- Sanity checks are performed

Intel TXT - Components

CPU and chipset contain special Intel TXT registers, many of them readable and/or writable only by ACM and CPU microcode

Authenticated Code Modules (ACMs): can only be created by Intel and are digitally signed using a private key only known to Intel. The public key is hardwired into hardware registers in the chipset, and only a module signed with the matching private key is allowed to execute.

- BIOS ACM: On startup, it measures the BIOS boot block on exiting the BIOS locks some registers, preventing hostile software from accessing them
- SINIT ACM: called by the OS to perform a measured launch
- Both run in a special internal CPU memory, preventing DMA access

Intel TXT - Components

TPM: registers in the TPM are used to store measurements of components involved in the boot process.

NV Indices: nonvolatile indices track state information required by the verified launch process. They play a key role in Intel TXT:

- Securely pass information and states between ACMs
- Securely maintain state between platform resets and power cycles
- Protect OEM and user policies from malicious alteration

AMD Secure Technology™

Dedicated hardware security subsystem that runs independently from the platform's main core processors [1]

- Formerly known as Platform Secure Processor [2]

Integrated into the same SoC of the main processor

An isolated environment in which security-sensitive components can run without being affected by the main software running

Provide the immutable hardware Root of Trust that can be used as the basis for providing the chain of trust from the hardware up to the OS.

Provide the Root of Trust for the Hardware Validated Boot (HVB), a secure boot process that verifies the integrity of the system BIOS

[1] Arthur, Will, David Challener, and Kenneth Goldman. "Platform Security Technologies That Use TPM 2.0." *A Practical Guide to TPM 2.0*. Apress, Berkeley, CA, 2015. 331-348.
[2] https://www.amd.com/en/technologies/security

AMD PSP - Components

Dedicated 32-bit microcontroller: ARM with TrustZone technology

Isolated on-chip ROM and SRAM: the ROM contains the initial immutable PSP code

DRAM carved out via hardware barrier and encrypted

Secure off-chip NV storage access

AMD PSP - Components

Hardware logic for secure control of CPU core boot

Cryptographic coprocessor (CCP), made up of:

- Random Number Generator
- Engines for standard algorithms (AES, RSA, …)
- Key storage block, composed of two areas:
  - One dedicated to storing system keys that can be used by privileged software but are never readable
  - One where keys can be loaded, used and evicted during normal operation by software running either on the PSP or on the main OS.

AMD PSP – Hardware Validate Boot

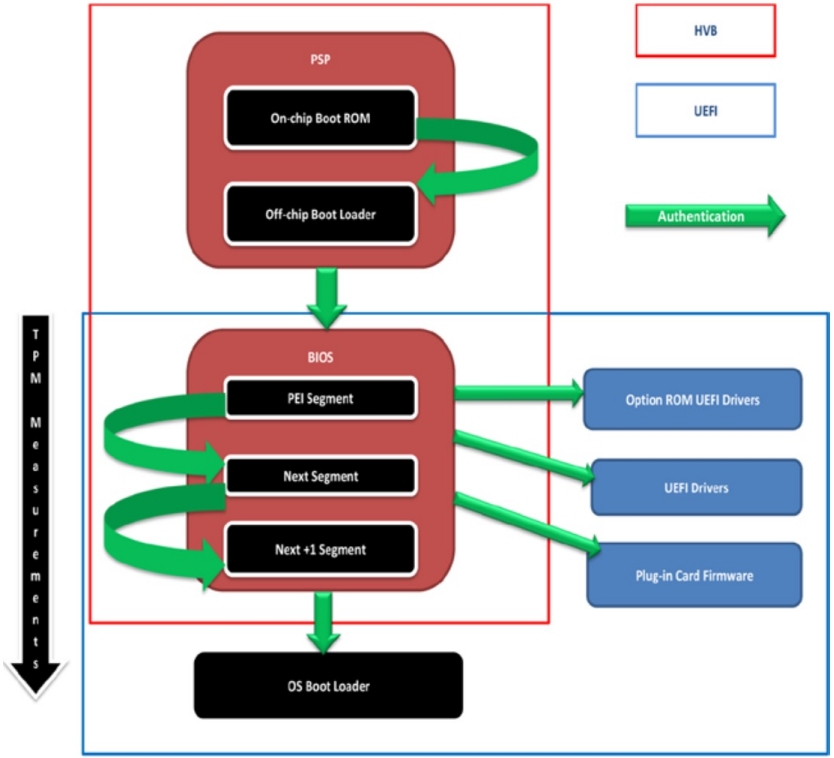
It is an AMD-specific form of secure boot

- Roots the trust to hardware in an immutable PSP on-chip ROM
- Verifies the integrity of the system ROM firmware (BIOS)

The PSP ROM validates a secure boot key and then uses the key to validate the PSP firmware

The PSP firmware loads and starts the system application execution.

The PSP then initiates BIOS execution



Arthur, Will, David Challener, and Kenneth Goldman. "Platform Security Technologies That Use TPM 2.0." *A Practical Guide to TPM 2.0*. Apress, Berkeley, CA, 2015. 331-348.

Custom Solutions – ARM® Trust Zone®

TrustZone provides a facility to create a virtual second processor inside a single system-on-chip (SoC)

Two special operating mode

- the Normal World (NWd), which runs the main OS and user interface
- the Secure World (SWd), which runs a trusted software stack implementing security features

The SoC hardware keeps the two worlds separate so that the main OS can't interfere with programs or data in the SWd.

Users can retain trust in the integrity and confidentiality of SWd data even when they can't trust the state of the device as a whole

https://developer.arm.com/ip-products/security-ip/trustzone

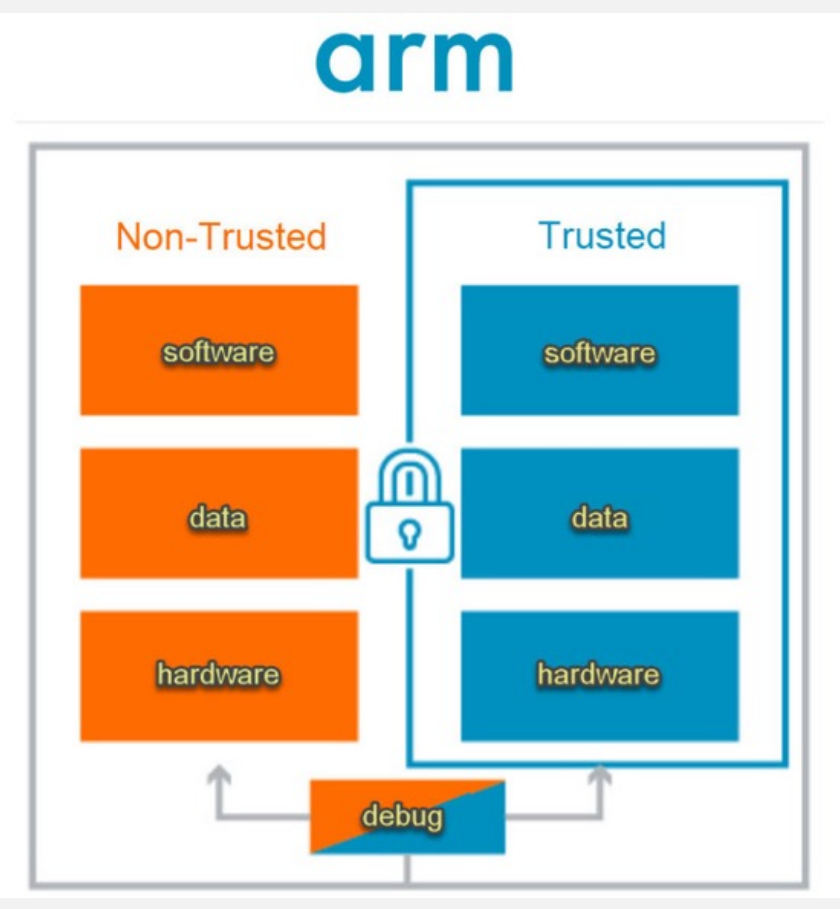
Custom Solutions – ARM® Trust Zone®

Given the business model of ARM, TrustZone is an architectural feature

An architectural feature is something that is baked into the architecture specification and is implemented through standard mechanisms and signals

Is not implemented as software or any auxiliary module/IP block

The chip hardware enforces security separation and does not rely on software or logical access control systems



ARM® Trust Zone® - Components

The NS bit: is the central manifestation of TrustZone in the processor architecture. It's a control signal accompanying all read and write transactions to system bus masters, including memory devices.

The Monitor: Alongside the two explicit operating modes, a third processor mode called Monitor mode runs a third separate software stack.

- A small amount of firmware required to coordinate the two worlds
- The Monitor must allow the transition from SWd to NWd (and vice versa).
- The Monitor can access all the crucial security data in the system

ARM® Trust Zone® - Components

Interrupts: interrupts from secure peripherals can be routed directly to the SWd without passing through any untrusted code at any privilege level.

- the Monitor catches interrupts, and the Monitor decides (based on a configuration table) which driver (SWd or NWd) should receive the interrupt
- When entering a secure transaction, the SWd can reserve the peripheral, meaning it receives all the interrupts
- Upon completion, the SWd can release the peripheral, informing the Monitor that it should send interrupts on to the NWd driver instead

Microsoft BitLocker

It was introduced in 2006 with Windows Vista and is present today, Windows 10.

Volume Encryption feature of the Microsoft Operating System

BitLocker securely stores a series of keys on each protected volume

- Key security is assessed using a TPM

It is possible to use BitLocker even if the system does not have a TPM, but the security of the stored cryptographic keys will be affected.

https://docs.microsoft.com/it-it/windows/security/information-protection/bitlocker/bitlocker-overview

BitLocker – Key Management

The BitLocker key management system uses a series of keys to protect the data.

The key used to protect the data of a volume, the Full Volume Encryption Key (FVEK), is stored on the protected volume.

- To prevent unauthorized access, the FVEK is encrypted using another key.
- The key used to encrypt the FVEK is the Volume Master Key (VMK)

VMK is also stored on the protected volume alongside with several other copies

Each copy of the VMK is encrypted using a different key.

The different keys allow different access mechanisms to access the stored data.

Each access mechanism can decrypt a copy of the VMK, which in turn is used to decrypt the FVEK, which is used to decrypt the protected data.

BitLocker – Access Mechanism

TPM: This does not require interaction with the user to unlock BitLocker and access the volume. If the TPM is missing or other components in the boot process have been compromised, it is impossible to access the system volumes.

TPM plus a PIN: in addition to the TPM, BitLocker asks the user to insert a PIN. It is impossible to access the encrypted volume if the PIN is wrong or not inserted.

TPM plus an external (USB) device (aka "Startup Key"): in addition to the protection granted by the use of the TPM, part of the cryptographic key is stored on an external USB drive. This key is called the Startup Key.

BitLocker – Access Mechanism [Cont.d]

Recovery Password: if BitLocker enters recovery mode, the user is asked to enter a recovery password to access data.

Unprotected key saved to the protected volume: if BitLocker is enabled, a user can disable it. The user can still access old data without the need to decrypt them.

- The operating system writes a 256-bit clear key to the volume's metadata along with a copy of the VMK encrypted with that key
- The system can decrypt the VMK and FVEK without any other information

BitLocker – Interaction with TPM [1/3]

The TPM keeps several Platform Configuration Registers or PCRs.

A specific function can only modify PCRs, which sets a PCR to the hash of its old value and a supplied data string.

There is no other way to set the value of a PCR, so if a PCR has value x after a sequence of extends, the only way to reach the value x again is to perform the same sequence of extends after a power-up.

The seal/unseal functions of the TPM allow selective access to cryptographic keys based on PCR values.

- The seal function encrypts a key into a string, which can only be decrypted by that same TPM.
- The TPM will decrypt the string if the selected PCRs have the value specified during the seal operation. In other words, we can store a key in an encrypted string so it can only be accessed when selected PCRs have a particular value.

BitLocker – Interaction with TPM [2/3]

At power-up, the processor starts running the BIOS from ROM. The first part of the BIOS cannot be modified.

This part extends the BIOS PCR with the entire BIOS code and continues the BIOS start-up.

After BIOS initialization, the BIOS reads the Master Boot Record (MBR) of the hard disk, extends the boot sector PCR with the sector's data, and executes the boot sector code.

BitLocker – Interaction with TPM [3/3]

The boot sequence of a PC contains several more iterations, but in each case, the newly loaded code is first measured using an extend function before it is executed.

The boot sequence switches to using BitLocker encryption at the first opportunity

- Before the switch, PCRs are used to measure what code is running
- At the switch point, the TPM unseals the BitLocker volume encryption key
- After the switch, all further data is read from the encrypted volume.

Synopsys – DesignWare® tRoot™

Is an IP that could be incorporated into an SoC and run without a ROM

Synopsys implementation of a Root of Trust device

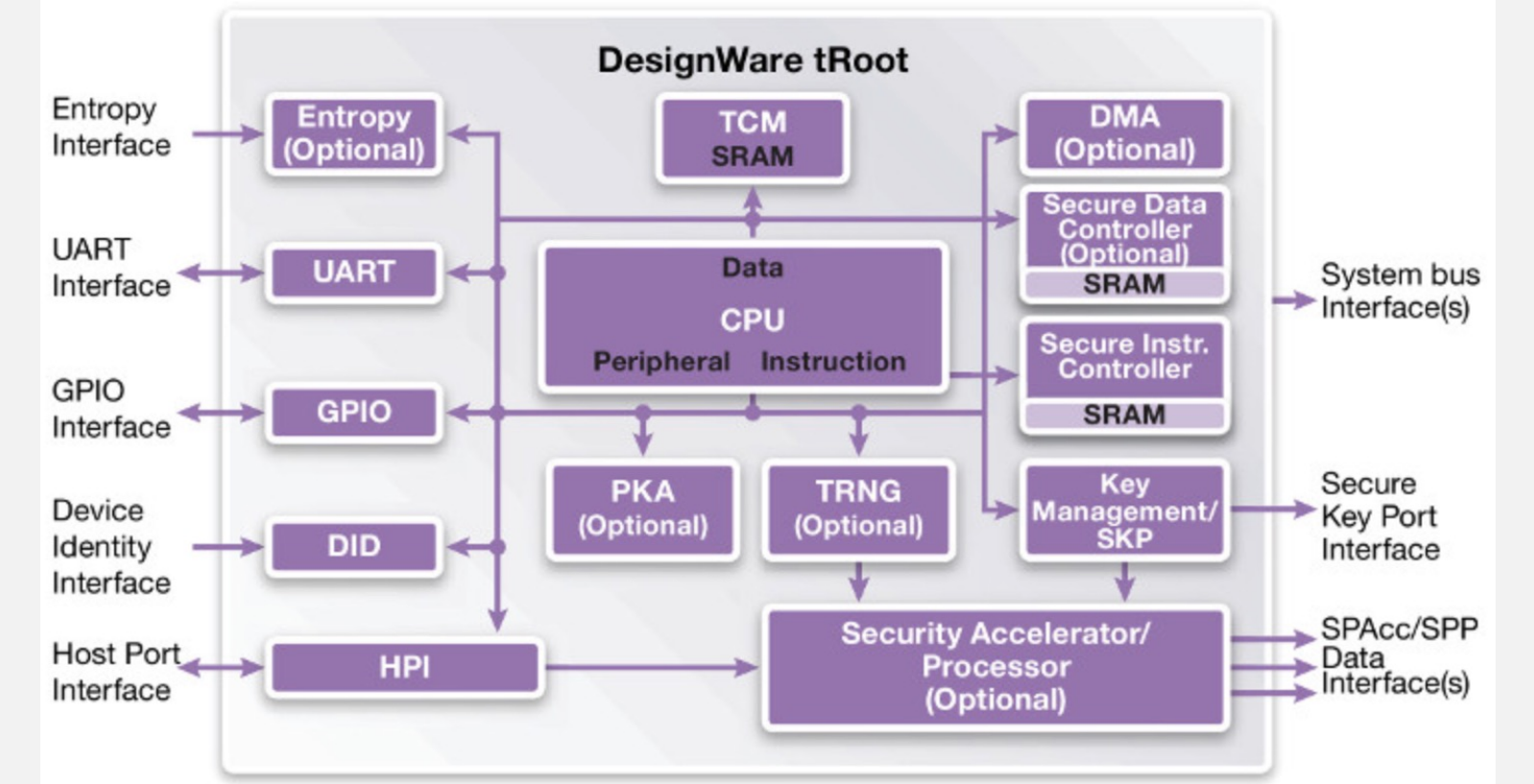
Code can be stored in any unsecured non-volatile storage, and the tRoot module's firmware can be expanded despite its fixed physical implementation because the module can securely share system memory with other devices on-chip.

https://www.synopsys.com/dw/ipdir.php?ds=security-troot-hw-root-of-trust

DesignWare® tRoot™ - Features

- FIPS 140-2 compliant anti-tamper module
- Secure boot and access control
- Secure identification and authentication
- Secure storage for keys and sensitive data
- Secure communications with other on-chip components
- Secure in-field firmware updates
- Run-time integrity protection

DesignWare® tRoot™ - Architecture



Apple Secure Enclave Processor (SEP)

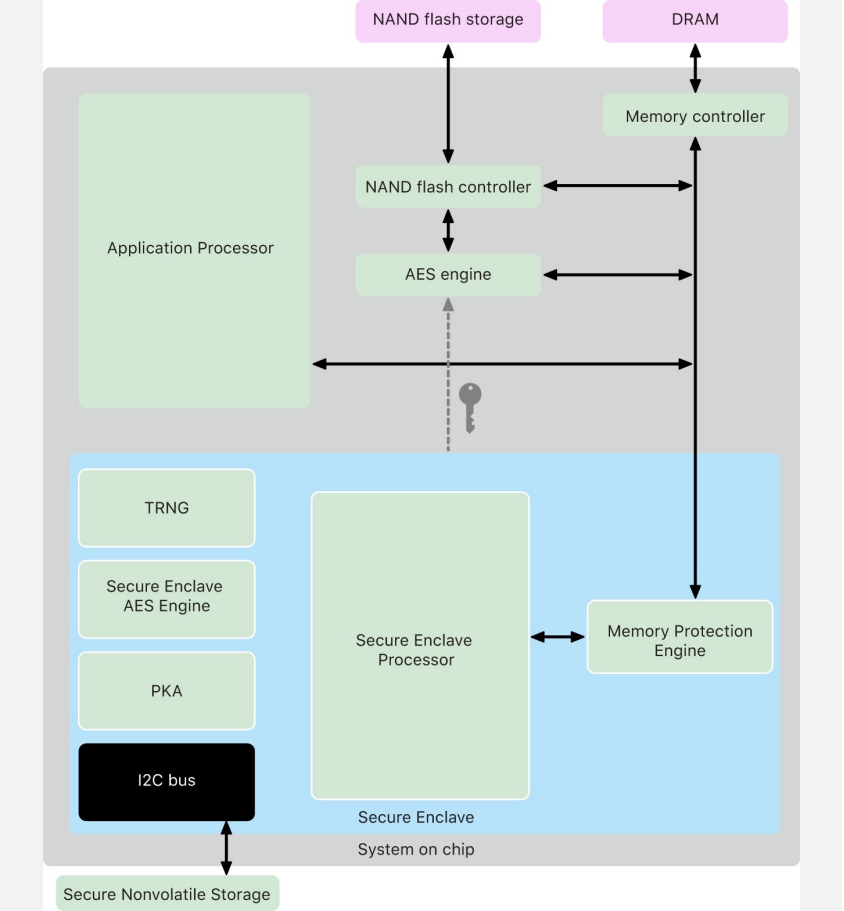
External chip integrated into the motherboard

Secure generation and storage for cryptographic keys

Random number generation

Execution of cryptographic functions

Isolation from the main processor



https://support.apple.com/it-it/guide/security/sec59b0b31ff/web

Apple Secure Enclave Processor (SEP)

Notably, SEP is used to support services that process highly sensitive data, such as Touch ID and Apple Pay

The Secure Enclave Processor runs its operating system, SEPOS, and fully operates in its own protected memory space in physical memory.

An attacker who has gained full control of the main OS cannot easily access SEP and its data.

Since the main OS has no direct access to SEPOS, it communicates through a mechanism known as the secure mailbox.

The secure mailbox is implemented as a shared memory region between the application processor and the secure enclave processor, where messages are passed using an interrupt-based delivery system

While SEP has existed since the iPhone 5S, little information exists on its inner workings. No part of SEPOS is documented by Apple or by any third party.

Apple SEP - SEPOS

At the heart of the secure enclave processor runs SEPOS, an L4 microkernel-based operating system [1].

In SEPOS, only the root task can invoke privileged system calls.

In SEPOS, the root task is called SEPOS. It is responsible for starting all applications that run in SEPOS, as well as maintaining contextual information about every running task

- virtual address space
- privilege level
- running threads
- …

[1] Gernot Heiser, Kevin Elphinstone. L4 Microkernels: The Lessons from 20 Years of Research and Deployment. ACM Transactions on Computer Systems. https://www.nicta.com.au/publications/research-publications/?pid=8988

Apple SEP – SEPOS Drivers

SEPOS includes several drivers that are designed to support services and applications such as the True Random Number Generator (TRNG) and the AP/SEP endpoint driver (AKF):

- AES SEP
- AES HDCP
- AES AP (CMC)
- Endpoint management (mailbox)
- Key management
- Power management
- True random number generator
- ….

Apple SEP – SEPOS Services

Like drivers, also hosted by their own application, but are implemented in a much simpler way.

- Key generation service
- Test service
- Anti replay service
- Entitlement service

Apple SEP – SEPOS Applications

SEPOS also runs several applications designed to support various applications, services, and frameworks implemented in the application processor.

- ART Manager/Mate: anti-replay token manager/mate is an application that handles anti-replay tokens
- Secure Biometric Engine: is responsible for handling biometric information
- Secure Credential Manager: manages user credentials so none of the internal data structures are exposed to the main OS (AP)
- Secure Key Store: manages the secure key storage , isolating internal data structures from the main OS
- SEP Secure Element (SSE): is an application that handles requests for the Secure Element

Google – Titan

Secure and anti-tamper chip

Integrated directly into the Google Cloud Platform

Available as a token for 2FA

Secure key generation and storage

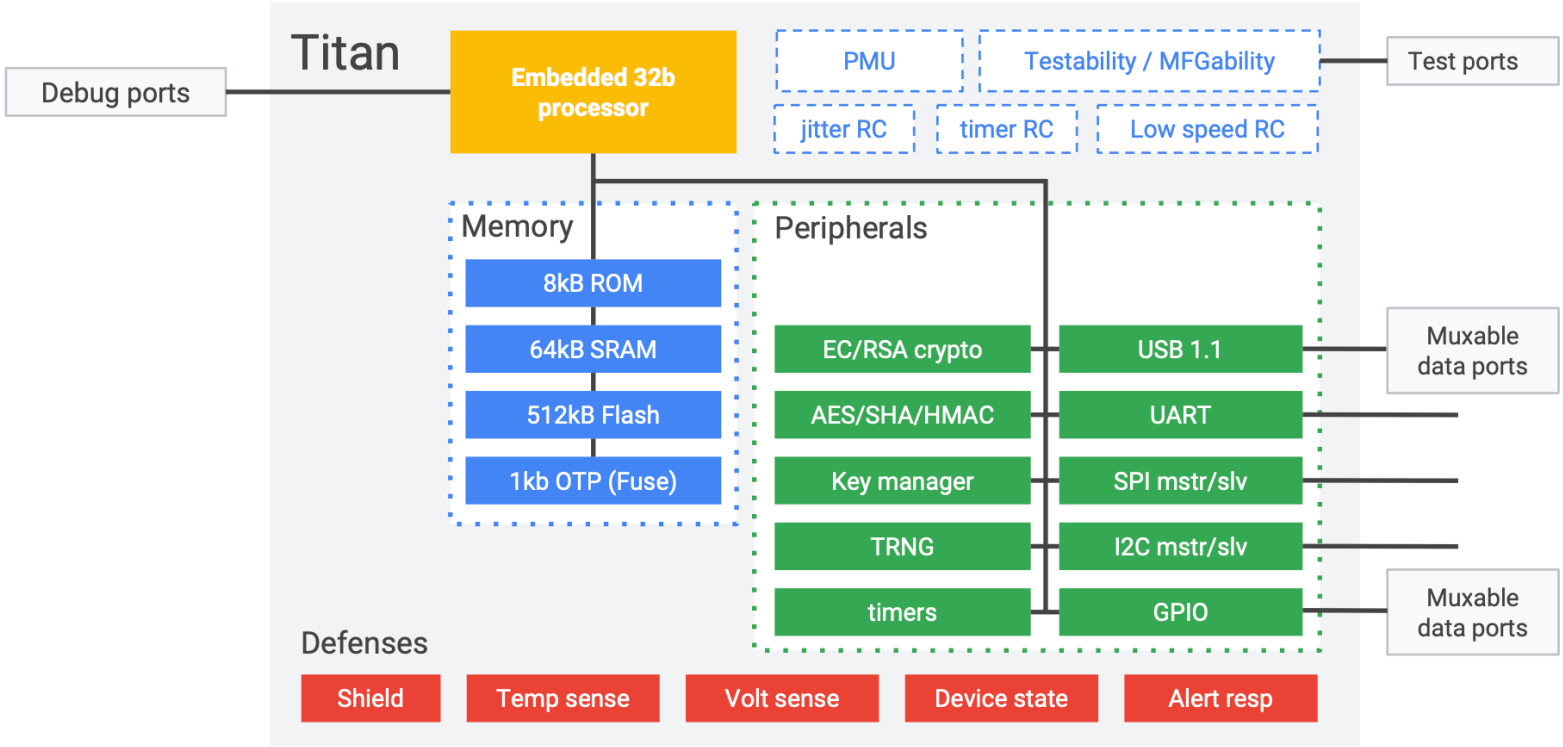
Implement a secure boot process within the chip to ensure integrity

Provide True Random Number Generation

Provide cryptographic functionalities: AES, SHA/HMAC, accelerator for EC and RSA

https://cloud.google.com/titan-security-key

Google – Titan Architecture



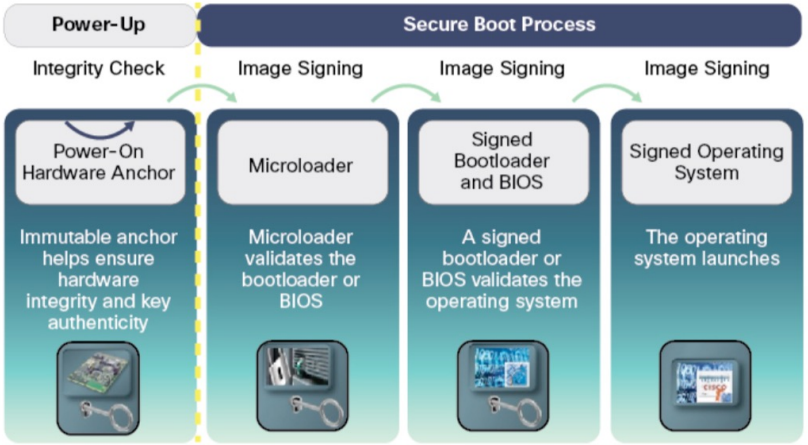
CISCO® Trust Anchor

Tamper-resistant chips integrated into many CISCO products

Provides several functionalities to the running OS, from secure storage to crypto services

Basis of a chain of trust that guarantees the integrity of running CISCO software

An FPGA is used as the root of trust to validate the bootloader image for the next stage in the secure boot process



https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/trustworthy-technologies-datasheet.pdf

CISCO® Trust Anchor - Characteristics

Nonvolatile secure storage: highly secure storage for keys, passwords, customer credentials, and other critical security information for the device. Allocating secure storage outside the Trust Anchor module is also possible.

Random Number Generation: provides a NIST SP 800-90A and B certifiable RNG

Entropy Source: extracted from a truly random source within the Trust Anchor

CISCO® Trust Anchor - Characteristics

Secure Unique Identifier (SUDI): is an X.509v3 certificate which maintains the product identifier and serial number.

- Can be used as an unchangeable identity for configuration, security, auditing, and management
- The SUDI credential can be either RSA or Elliptic Curve Digital Signature Algorithm (ECDSA) based
- The key pair is cryptographically bound to a specific Trust Anchor chip and the private key is never exported

Crypto Services: SHA256/512 algorithms for code signing and integrity checks, RSA and/or ECC.

- The SUDI can be used for asymmetric key operations allowing remote authentication

Politecnico
di Torino

Built-in Security Features

Open Security Platforms

Platforms designed with cybersecurity in mind

Packed with strong cybersecurity features:

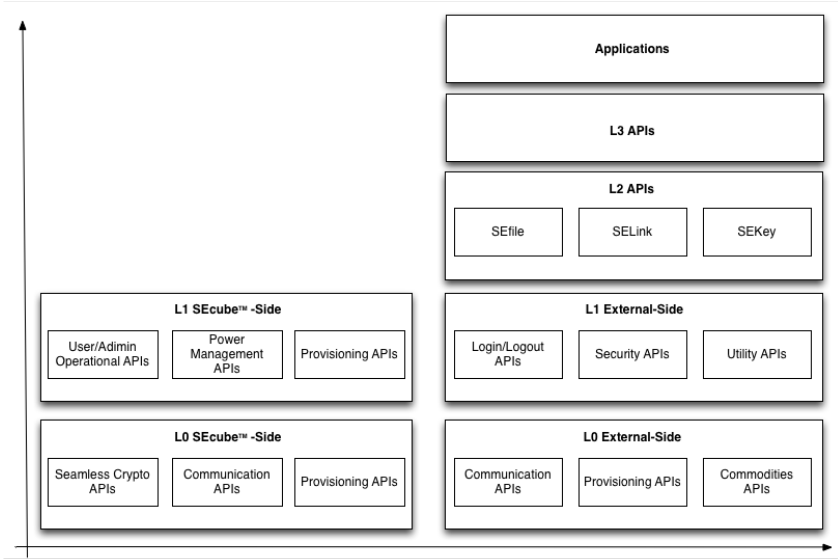
- Hardware accelerators for cryptography
- Anti-tamper
- Secure boot process
- …

Hardware Platforms – SEcube™

System-In-Package developed by Blu5™ Group

- Cortex-M4 microcontroller
- Flexible and fast FPGA
- SmartCard certified EAL 5+

Strong Cybersecurity features and capabilities



https://www.secube.eu/

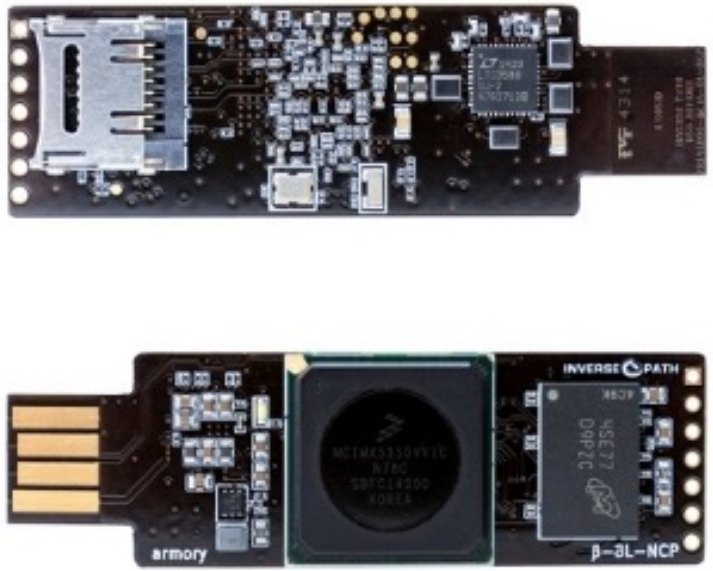
Hardware Platforms – USB Armory

Open hardware platform developed by Inverse Path

Open source software

Strong security features:

- Secure Boot
- ARM® TrustZone® enabled



https://inversepath.com/usbarmory_mark-one.html

Built-in Security Features

Functionalities present in most modern microcontrollers

Mostly introduced for safety

Proper exploitation could significantly increase the system protection against the common threats in the embedded system landscape

Built-in Security Features

There exist nowadays different microcontroller manufacturers.

For the sake of simplicity in the sequel, we will focus only on solutions provided by ST-microelectronics.

But the presented features are available, more or less, in all modern microcontrollers.

ST - Built-in Security Features

Integrity and Safety
Crypto
Debug Lock
Tamper Protection
Privileges Permission Management
Memory Protection
Traceability
Secure Firmware Update

http://www.emcu.it/SILICA-STDay-2016/X/Presentazioni/2_STM32&SecureElements.pdf

Integrity and Safety

CRC: Used to verify data transmission or storage integrity. Computes a signature of the software at runtime

Power Supply Integrity Monitoring: Ultra-safe supply monitoring. Several flag statuses to determine what causes a reset

Read While Write: Efficient tamper detection logging

Integrity and Safety

Clock Security System (CSS): Independent clock sources and recovery systems. Internal clock for secure program execution

Error Correction Code (ECC): Robust memory integrity. Hardened protection against fault injection attacks

Parity Check: Memory content integrity check

Integrity and Safety

Temperature Sensor: Check if the device operates in the expected temperature range. Protection against temperature or laser attacks

Watchdogs: Independent watchdog and window watchdog for software timing control

Crypto

Random Number Generator: On-chip entropy generation. Ensure strong keys and protect against replay attacks.

Hashing Functions & HMAC: The Hash algorithm provides a way to guarantee the integrity of information, verify digital signatures and message authentication codes

Crypto

Symmetric Cryptography: various cryptography algorithms implemented in both hardware and software

Asymmetric Cryptography: RSA signature function. ECC (Elliptic Curve Cryptography)

Debug Access Protection

JTAG or SWD: Prevent unauthorized access to the device through debug interfaces.

Taper Protection

Anti-Tamper: Protect against a wide range of physical attacks on HW system outside the MCU

Backup Domain: Maintains active tamper protection even in low power modes.

RTC (Alarm Timestamp): Timestamp on tamper event.

Tamper Protection

Backup Register: For Confidential data storage. Tamper automatically deletes registered content

GPIO Configuration Locking: Lock the selected GPIO. It is impossible to unlock until the next reset. It is possible to lock communication channels after tamper detection.

Privileges Permission Management

Memory Protection Unit (MPU): Divides the memory map into several regions with privilege permissions and access rules.

Firewall: Even more restrictive than MPU. Made to protect a specific part of code or data from the rest of the code executed outside the protected area.

Memory Protection

Read Protection (RDP): Global memory access control management. It prevents memory dumps and safeguards users' IPs.

Write Protection (WRP): Each sector can be protected against unwanted write operations.

Memory Protection

Proprietary Code Protection (PCROP): Each Sector can be configured in "execute only".

Mass Erase: Safely remove IPs and confidential data. Force factory reset.

Traceability

Device electronic 96-bit Unique ID: Enables product traceability. Can be used for security key diversification.

Secure Firmware Update

Software SFU: Secure firmware upgrade capability. Allows for different types of software updates with integrity check capabilities.