Hardware-based Security part II: Implementations

Nicolò MAUNERO

Paolo PRINETTO

CINI Cybersecurity National Laboratory

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Acknowledgments

- The presentation includes material from
  - Gianluca ROASCIO – Politecnico di Torino
- His valuable contributions are here acknowledged and highly appreciated

Organization

- This is the 2nd part of the lecture that, for sake of usability, has been split into two parts:
  - *Part I : Introduction & Basic concepts*
  - *Part II : Implementations*

Outline

- Introduction
- Roots of Trust
- Trust Anchors
- System level solutions
- Architectural level solutions
- Device level solutions

- Security-oriented components
- Proprietary Solutions
- Open Security Platforms
- Built-in Security Features

Part II : Implementations

Prerequisites

- Lecture:
  - *HS_1.4.1 - Hardware-based Security - Part I: Introduction & Basic concepts*

Outline

- Security-oriented components
- Proprietary Solutions
- Open Security Platforms
- Built-in Security Features

Outline

- Security-oriented components
- Proprietary Solutions
- Open Security Platforms
- Built-in Security Features

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*

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
- With respect to software implementation, are
  - Faster
  - Less prone to exploitation than software
- 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 cyphers can be guaranteed

Hardware Cyphers

- As with software:
  - Several Hardware implementations have been proposed
  - They can be clustered as
    - *Block Cyphers*
    - *Stream Cyphers*
    - *Hash Functions*

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*

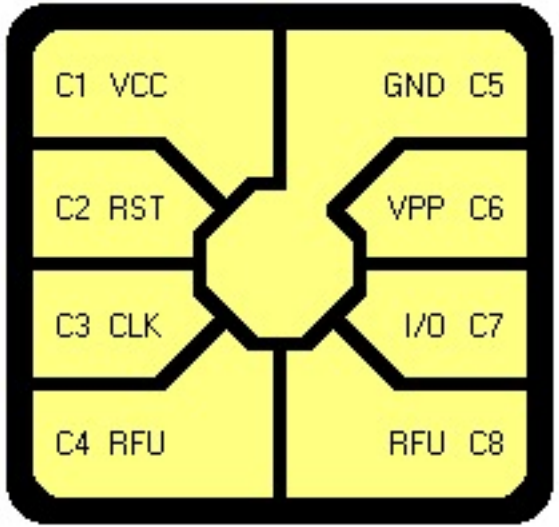
Smart Cards

- 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

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*

Secure Storage Devices -- Guidelines

- A secure storage device should be designed and manufactured in a way to provide certain levels of protection, including:
  - Tamper-evidence
  - Tamper-resistance
  - Secure authentication mechanism
  - Drive's controller security
  - Data encryption capability
  - Secure Erase facility.

[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 mechanism are
  - **PIN pad:** can be subjected to a very simple exploit, some button may present sign 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 input system is not properly manufactured, can be bypassed even without the need of faking the owner fingerprint

Drive's Controller Security

- The drive's controller must be developed in order 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)

Data Encryption

- Core of the security of the device
- Strong and new encryption standard must be used
- Old or badly implemented encryption algorithms can be easily broken by an attacker

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*

See lecture:
BNT#2 - Random Number Generation

Outline

- Security-oriented components
- Proprietary Solutions
- Open Security Platforms
- Built-in Security Features

Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

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)

- 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 through the use of 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

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

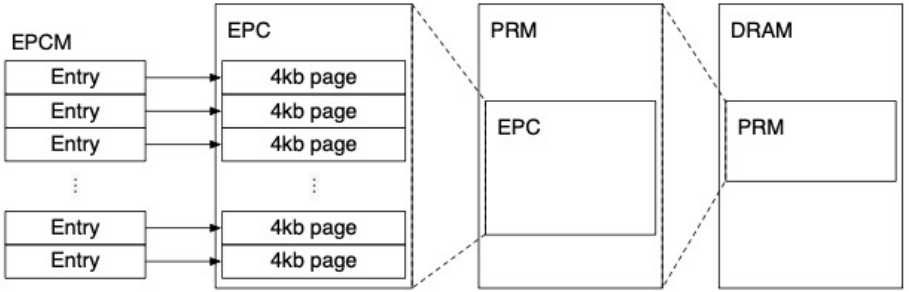
Intel® SGX – MME Overview

- 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 what is called 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

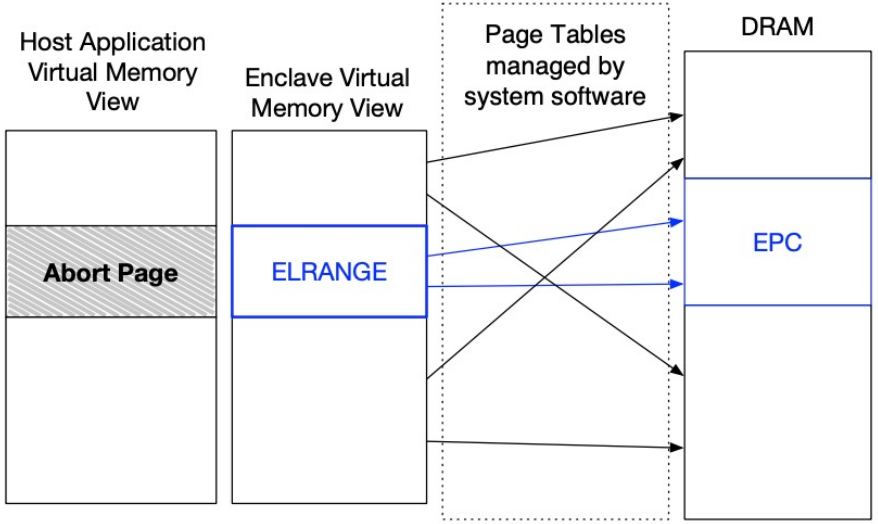
- SGX sets aside a memory region, called the Processor Reserved Memory (**PRM**)
- The PRM holds the Enclave Page Cache (**EPC**) that store enclave code
- **PRM** is a subset of DRAM that cannot be directly accessed by other software e and data
- SGX stores per-enclave metadata in a SGX Enclave Control 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 the use of 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 makes sure that there is no logical processor executing code inside the enclave that owns the page to be removed
  - An enclave is completely destroyed when the EPC page holding its SECS is freed
  - An enclave's SECS page can only be deallocated after all the enclave's 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]

[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] Wojtczuk, Rafal, and Alexander Tereshkin. "Attacking intel bios." *BlackHat, Las Vegas, USA* (2009).
[3] Futral, William, and James Greene. "Introduction to trust and intel® trusted execution technology." In *Intel® Trusted Execution Technology for Server Platforms*, pp. 1-14. Apress, Berkeley, CA, 2013.[4] Rootkits Hoglund, Greg, and James Butler. *Rootkits: subverting the Windows kernel*. Addison-Wesley Professional, 2006.

Intel TXT - Functionalities

- A chain of trust is extended from the Intel processor hardware through the BIOS
- At OS level if a user want 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 perform a secure boot sequence before executing any components
- **Measurements**: keywords indicating a secure boot sequence
  - Integrity of component is checked (e.g. code hash is verified)
  - Sanity checks are performed

Intel TXT - Components

- **CPU and chipset** contains special Intel TXT registers, may 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 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

Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

AMD Secure Technology™

- Dedicated hardware security subsystem that runs independently form the platform's main core processors [1]
- Formerly known as **Platform Secure Processor** [2]
- Integrated into the same SoC of the main processor
- Isolated environment in which security-sensitive components can run without being affected by 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

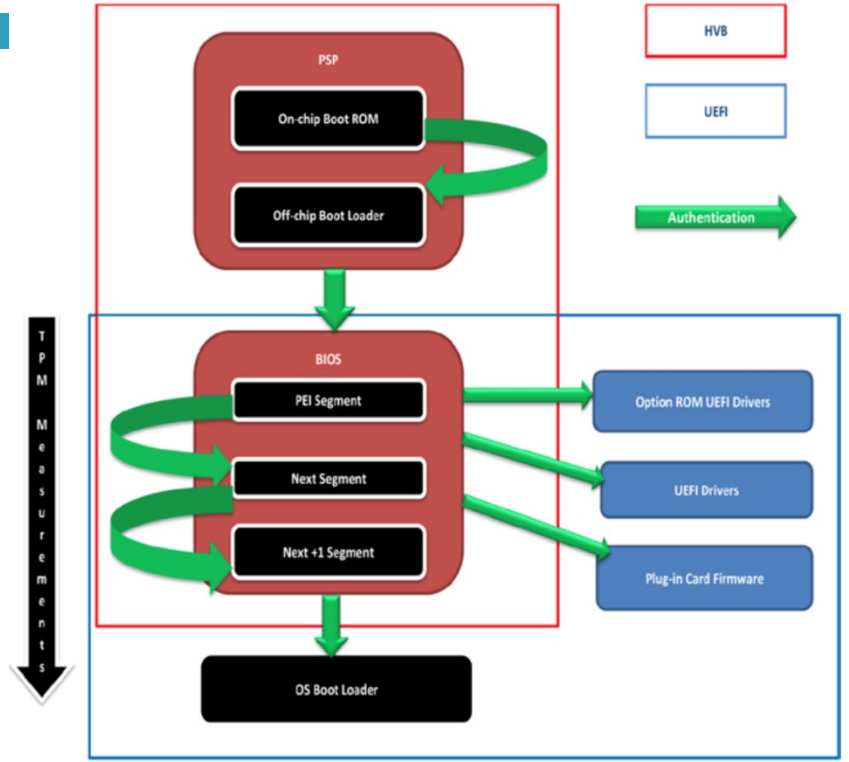
- Dedicated 32-bit microcontroller: ARM with TrustZone technology
- Isolated on-chip ROM and SRAM: the ROM contains the initial immutable PSP code
- DRAM carved out via hardware barrier and encrypted
- Secure off-chip NV storage access

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

- 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.

Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

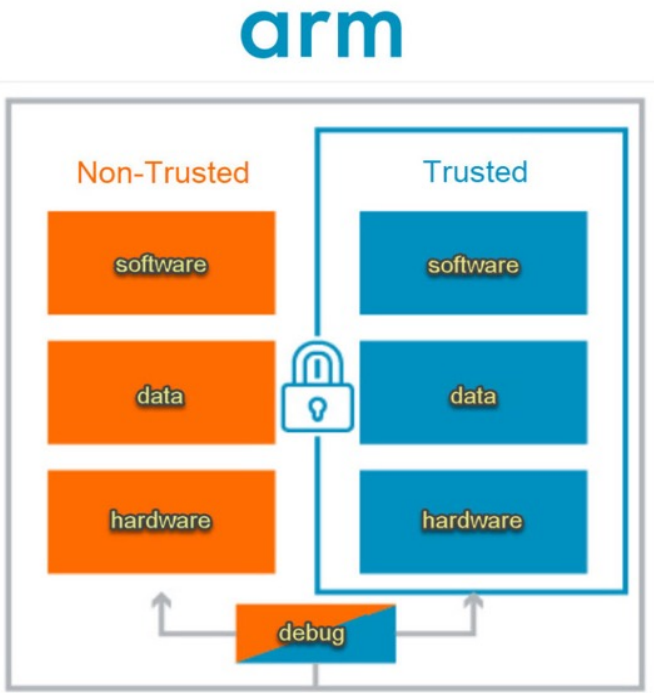
Custom Solutions – ARM® Trust Zone®

- 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 two worlds are kept separate by the SoC hardware 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 features
- 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
- Security separation is enforced by the chip hardware 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 that accompanies all read and write transactions to system bus masters, including memory devices
- **The Monitor**: Alongside the two explicit operating modes there is a third processor mode called *Monitor mode* that runs a third separate software stack
  - Small amount of firmware required to coordinate the two worlds
  - Transition from SWd and NWd (and viceversa) have to be allowed by the Monitor
  - The Monitor is able to access all the crucial security data in the sytem

ARM® Trust Zone® - Components

- **Interrupts:** interrupts from secure peripherals can be routed directly to the SWd without ever passing through any untrusted code at any privilege level
  - interrupts are caught by the Monitor, 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

Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

Microsoft BitLocker

- Introduced in 2006 with Windows Vista and present still in 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 at rest
- 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 be used to access the stored data
- Each access mechanism can be used to decrypt a copy of the VMK which in turn is used to decrypt the FVEK which in turn is used to decrypt the protected data

BitLocker – Access Mechanism

- **TPM:** This not require any interaction with the user for unlocking BitLocker and accessing the volume. If the TPM is missing or other component in the boot process have been compromised it is not possible to access the system volumes
- **TPM plus a PIN**: in addition to the TPM, BitLocker ask to the user the insertion of a PIN. It is not possible to access the encrypted volume if the PIN it 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 *Startup Key*

BitLocker – Access Mechanism [Cont.d]

- **Recovery Password**: if BitLocker enters in 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 decide to disable it. The user can still access old data without the need of 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
- PCRs can only be modified by a specific function, 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, then the only way to reach the value *x* again is to perform the exact 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 is used to encrypt a key into a string which can only be decrypted by that same TPM
  - The TPM will decrypt the string if and only if the selected PCRs have the value that was specified during the seal operation. In other words: we can store a key in an encrypted string so that 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 continue the BIOS start-up
- After BIOS initialization the BIOS reads the Master Boot Record (MBR) of the hard disk and extends the boot sector PCR with the sector's data, and then executes the code in the boot sector

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

Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

Synopsys – DesignWare® tRoot™

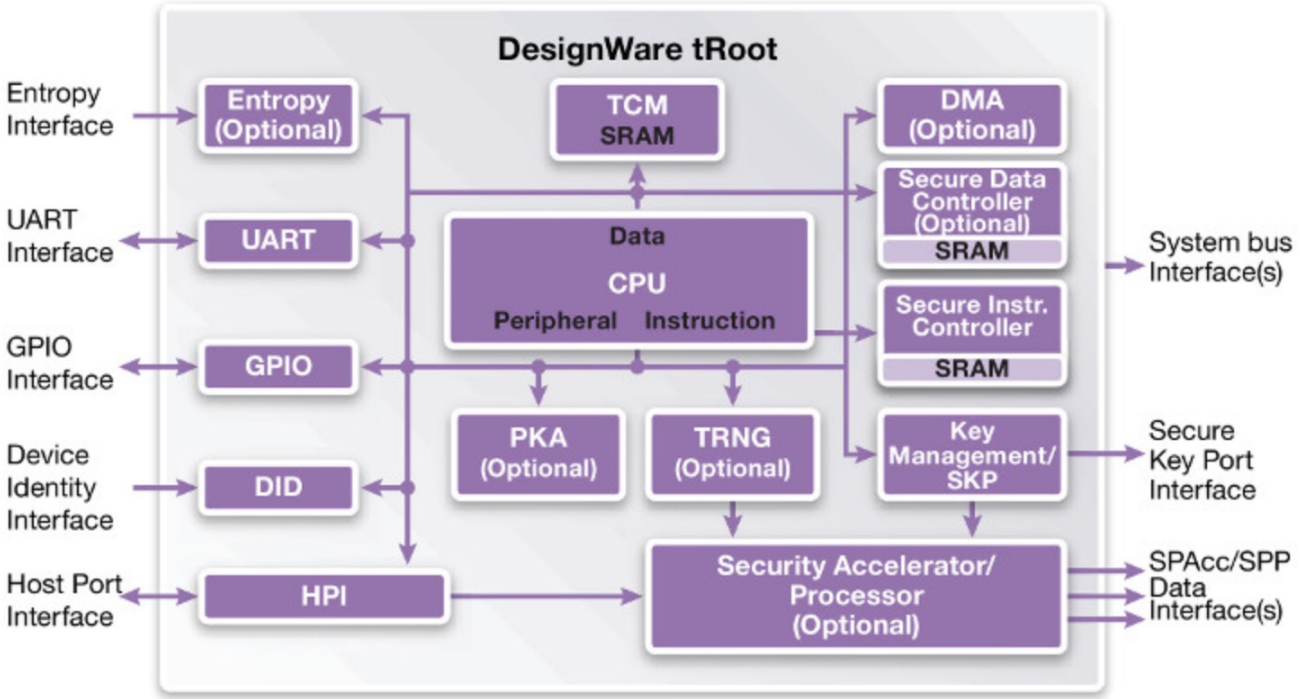
- Is an IP that could be incorporated into an SoC and run without a ROM
- Synopsys implementation of a Root of Trust deivice
- Code can be stored in any unsecured non-volatile storage and the tRoot module's firmware can be expanded despite its fixed physical implementation, because the module can securely 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



Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

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 own operating system, **SEPOS**, and fully operates in its own protected memory space in physical memory
- An attacker who has gained full control of main OS cannot easily gain access to 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 been around since the iPhone 5S, little information exists on its inner workings. No part of SEPOS is documented by Apple nor 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 simply called SEPOS. It is responsible for starting all applications that run in SEPOS, as well as maintaining contextual information about every running task
  - task's 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

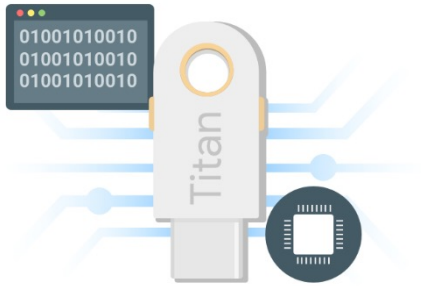
Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

Google – Titan

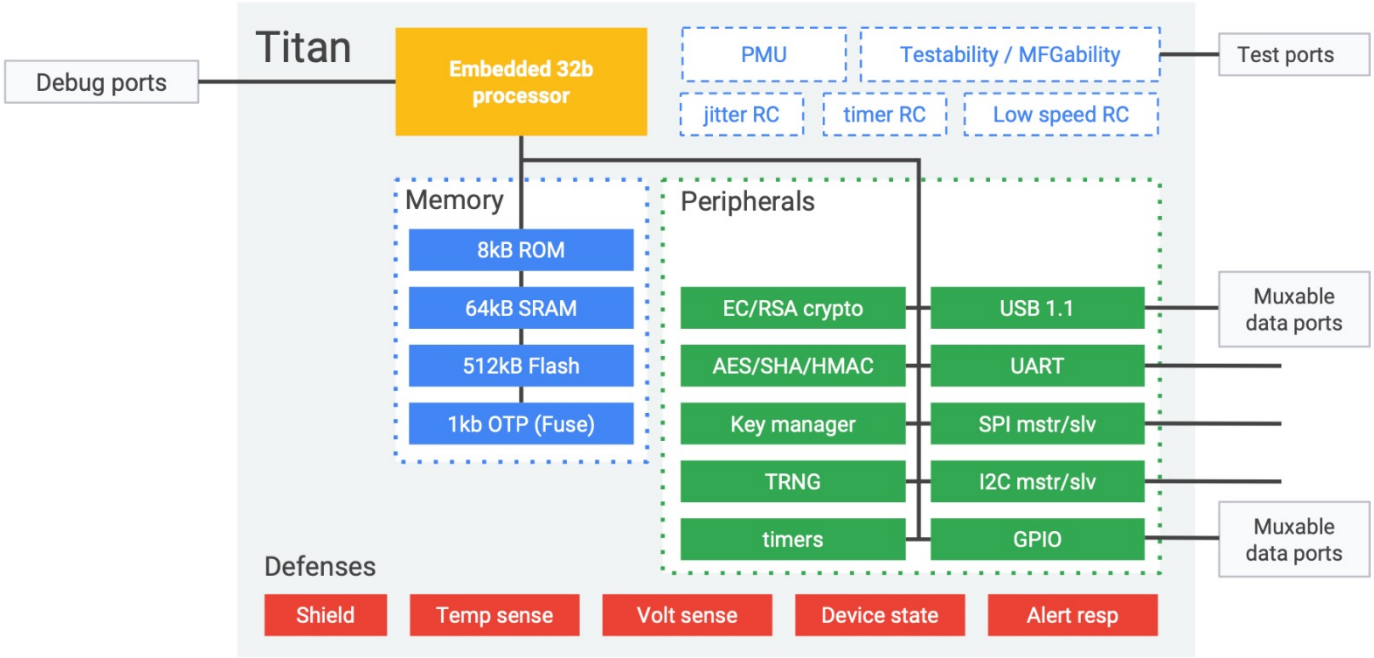
- Secure and anti tamper chip
- Integrated directly into Google Cloud Platform
- Available as a token for 2FA
- Secure keys generation and storage
- Implement a secure boot process within the chip to ensure integrity

- 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

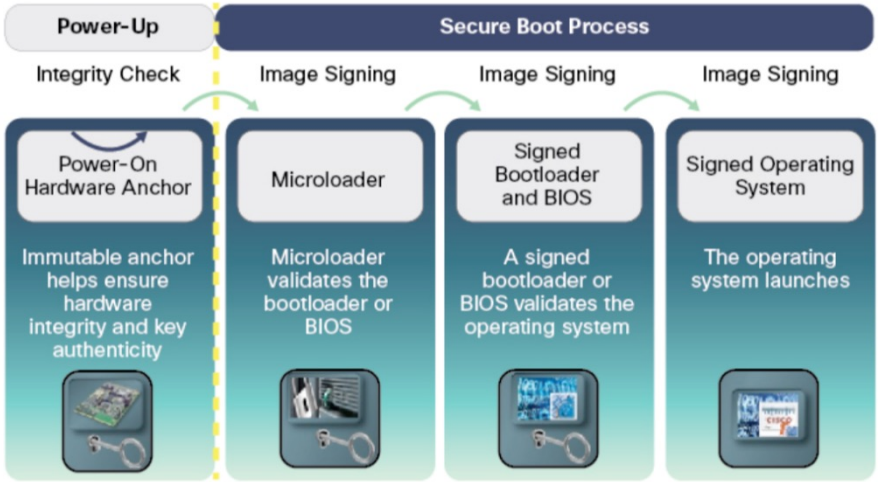


Proprietary Solutions

- *Intel® vPro® Platform*
- *AMD Secure Technology™*
- *ARM® TrustZone®*
- *Microsoft BitLocker*
- *Synopsys DesignWare® tRoot™*
- *Apple Secure Enclave Processor*
- *Google Titan*
- Cisco® Trust Anchor

CISCO® Trust Anchor

- Tamper resistant chip integrated in many CISCO products
- Provides several functionalities to the running OS, from secure storage to crypto services
- Basis of a chain of trust that guarantee 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 true 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

Outline

- Security-oriented components
- Proprietary Solutions
- Open Security Platforms
- Built-in Security Features

Open Security Platforms

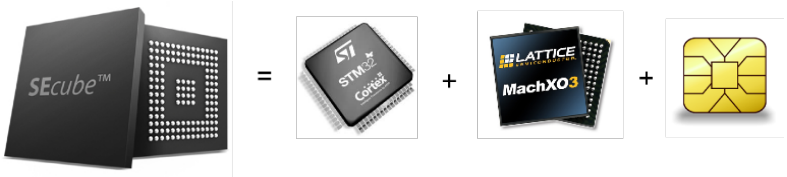
- Platforms designed with cybersecurity in mind
- Packed with strong cybersecurity features:
  - Hardware accelerators for cryptography
  - Anti tamper
  - Secure boot process
  - …

Open Security Platforms

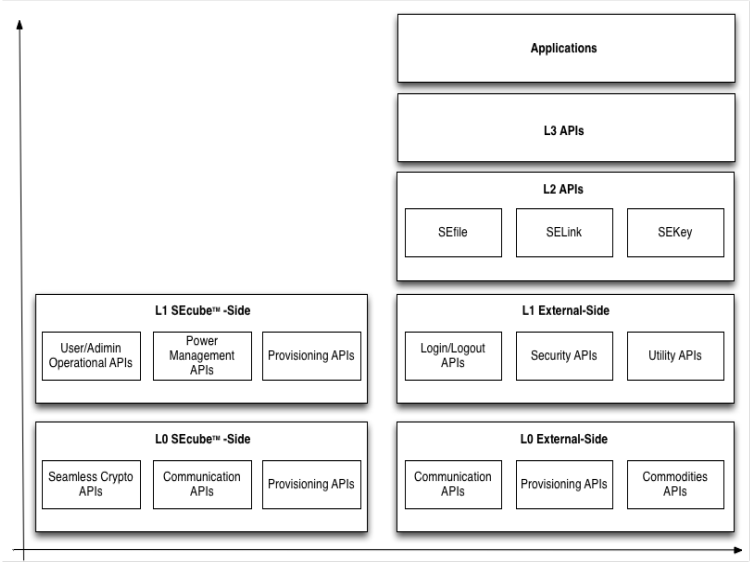
- SE*cube*™
- USB Armory
- …

Hardware Platforms – SE*cube*™

- System-In-Package developed by Blu5™ Group
  - Cortex-M4 microcontroller
  - Flexible and fast FPGA
  - SmartCard certified EAL 5+
- Strong Cybersecurity features and capabilities

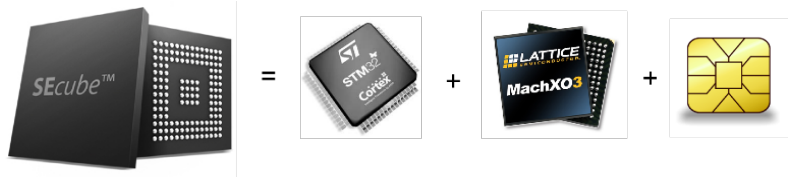


https://www.secube.eu/



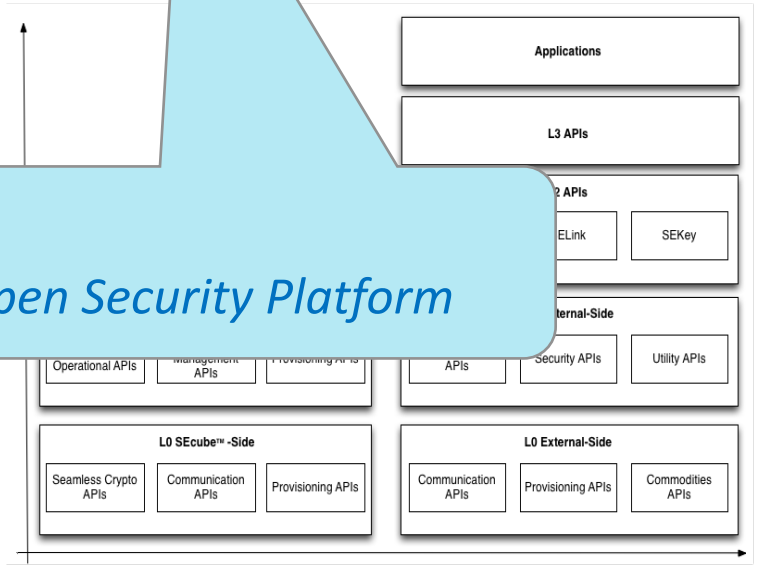
Hardware Platforms – SE*cube™*

- System-In-Package developed by Blu5™ Group
  - Cortex-M4 microcontroller
  - Flexible and fast FPGA
  - SmartCard certified EAL 5
- Strong Cybersecurity fe
  capabilities



https://www.secube.eu/

See lecture:
Secube™ - Open Security Platform



Hardware Platforms – USB Armory

- Open hardware platform developed by Inverse Path
- Open source software
- Strong security features:
  - Secure Boot
  - ARM® TrustZone® enabled

INVERSE PATH

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

Outline

- Security-oriented components
- Proprietary Solutions
- Open Security Platforms
- Built-in Security Features

Built-in Security Features

- Functionalities present in most of the modern microcontroller
- Mostly introduced for safety
- A 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 sake of simplicity in the sequel we will focus only on solutions provided by ST-microelectronics
- But the presented feature are available, more or less, in all modern microcontroller

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 status to determine what causes reset
- ***Read While Write***: Efficient tamper detection logging

Integrity and Safety

- *Clock Security System (CSS):* Independent clock sources and clock 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 is operating in 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, protect against replay attacks
- *Hashing Functions & HMAC:* Hash algorithm provides a way to guarantee the integrity of information, verify digital signatures and message authentication codes

Crypto

- *Symmetric Cryptography:* various cryptography algorithm 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 tamper protection active even in Low Power modes
- *RTC (Alarm Timestamp):* Timestamp on tamper event

Tamper Protection

- *Backup Register:* For Confidential data storage. Tamper automatically deletes registers content
- *GPIO Configuration Locking:* Lock of selected GPIO. Impossible to unlock until next reset. Capability to lock communication channels after tamper detection

Privileges Permission Management

- *Memory Protection Unit (MPU):* Divides the memory map into a number of 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 of the protected area

Memory Protection

- *Read Protection (RDP):* Global memory access control management. Prevents memory dumps, safeguarding user's IPs
- *Write Protection (WRP):* Each sectors 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-bitUnique ID:* Enables product traceability. Can be used for security key diversification

Secure Firmware Update

- *Software SFU:* Secure firmware upgrade capability. Allows for different type of software update with integrity check capabilities

Hardware-based Security part II: Implementations

Nicolò MAUNERO

Paolo PRINETTO

CINI Cybersecurity National Laboratory

https://cybersecnatlab.it