Hardware and Software Cybersecurity solutions in MCU-based embedded applications

The context: Cyber Resilience Act

In recent years, the discussion around the topic of *cybersecurity* has taken a central role throughout the entire life cycle of electronic devices. Following the publication of EU Regulation 2024/2847, better known as *the Cyber Resilience Act* [1], all CE-marked products will have to meet the cyber-security requirements as of 11 December 2027. In the field of medical devices, stricter cybersecurity requirements [2] are set out in the *Medical Device Regulation* (MDR) 2017/745 [3].

In this context, both hardware and software solutions must be adopted from the early stages of design, keeping in mind the state of the art in the context of cybersecurity and avoiding non-standard solutions as much as possible.

Hardware Solutions

Starting from hardware solutions, several integrated components can be identified, which meet the most diverse security needs.

1. **Cryptographic accelerators**: these components allow you to accelerate cryptographic operations (AES-128, AES-256, SHA, TRNG, etc.) in hardware, to reduce the computational load on the host CPU. The presence of this type of component is essential to enable the execution of cryptographic algorithms with reduced time.

2. **Hardware Security Module (HSM):** A device that performs several functions, including:
   - store secret information, such as cryptographic keys;
   - execute encryption algorithms, both symmetric and asymmetric;
   - Generate secure cryptographic keys.

   This module can have its own dedicated CPU or share resources with the host CPU. In the context of ARM® architectures, two different types of HSMs can be identified:
   a. *ARM® Trust Zone®*: Technology used both in application processors based on *Cortex-A* architecture [4], and *Arm Cortex-M* processors [5], in order to create a *Trusted Execution Environment* (TEE). The Trust Zone allows to isolate the execution of secure firmware from the non-secure one, within the same processor, through a hardware separation of the two worlds.

   b. *Secure Enclave*: CPU dedicated to the management of secrets and cryptographic operations, different from the host processor and therefore more secure, as it is physically isolated. This

technology is available in dual/multi-core architectures: some examples are *NXP's EdgeLock Secure Enclave* [6] and *Microchip's PIC32CK* [7] and *PIC32CZ* MCUs [8].

3. **Secure Element (SE):** External device that does not share host CPU resources. The SE provides maximum security in terms of attacks, as it is physically detached from the host CPU and implements a secure storage of critical information, such as cryptographic keys and certificates, at the hardware level. The SE can integrate both a cryptographic accelerator and an HSM, thus providing optimized performance in a single component. In addition, the SE is a *tamper-proof* device: any tampering is detected by the system itself, ensuring maximum security.

There are numerous hardware solutions capable of satisfying the most varied needs in terms of cybersecurity. However, these are to be integrated with special software implementations, for all-round protection of the integrated system.

Software Solutions

In the field of firmware and software, several processes can be identified that can implement cybersecurity strategies:

- **Secure Boot**: *Secure Boot* is an algorithm that ensures the execution of authentic firmware, usually executed during the boot phase of the embedded system, i.e. by the bootloader. The latter checks both the authenticity and integrity of the application code, for example through an asymmetric algorithm such as *the Elliptic Curve Digital Signature Algorithm* (ECDSA), which authenticates the firmware by knowing the public key associated with the private key with which the digital signature was generated.

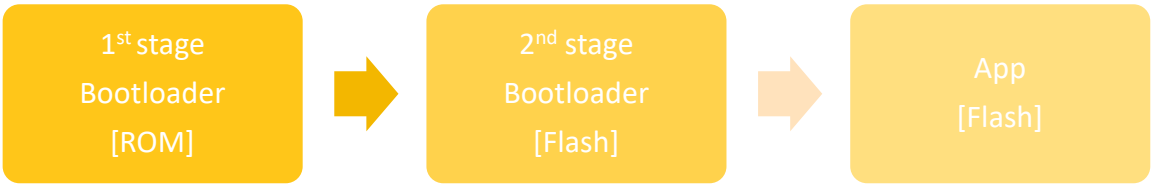

Figure 1: Secure Boot

Given the critical importance of the bootloader during the *Secure Boot* process, it is essential to verify the authenticity of the bootloader itself: the need arises for a primary and immutable bootloader (*first stage* bootloader), which usually resides in the ROM, that takes care of the authentication of the secondary bootloader (*second stage* bootloader), which in turn authenticates the application and can be updated as needed.

- **Secure Update**: the embedded system must be protected not only during boot, but also during software updates, both in the case of updating via a physical connection and *Over-the-Air (OTA)*. The system must therefore accept only authentic software as input, through verification of the digital signature.
  A crucial aspect in this phase is the so-called *rollback protection*: it is suggested to inhibit the programming of a software version older than the current one, to ensure that the version installed is the most up to date in terms of security. A cyber-attack could consist of installing an outdated version, with security bugs, thus exploiting vulnerability in the system with malicious intent.

- **Data Encryption**: the encryption of sensitive data is essential to ensure the secrecy of sensitive information, such as cryptographic secrets and passwords, and can concern both data in transit and at rest.
  The National Cybersecurity Agency (ACN) keeps the Guidelines for Cryptographic Functions updated [9], putting the spotlight, for example, on the storage of passwords through *hashing*. Cryptographic hash functions are not invertible: the password is provided as input to the hash function, which calculates the digest, a fixed-length string, from which the password cannot be obtained in plaintext, ensuring its privacy.
  Another example of encryption applies to the secure storage of firmware in memory, making it unreadable in cyber-attacks. The firmware can be encrypted in flash memory and read via *On-The-Fly Decryption* (OTFDEC), a protocol that consists in the decryption of data and code stored in memory in real time, ensuring minimal latency.

- **Secure Communication**: as already highlighted in the previous point, information in transit must also be protected from any attacks that would compromise its integrity and confidentiality. To do so, secure communication protocols shall be used.
  An example is *Transport Layer Security* (TLS), a standard communication protocol that guarantees privacy, authenticity and integrity of transmitted data, operating above the transport layer in TCP/IP networks. Security lies in the *handshake* phase, or negotiation, which creates a secure channel between the two actors in communication: an asymmetric algorithm is used, based on a public key and private key, to share a symmetric key to be used to encrypt the messages exchanged during the communication session. Authentication is central to TLS, as the server must provide the client with a certificate that guarantees its authenticity. In *mutual TLS*, the client must also be authenticated by the server.

It is possible to understand how key and certificate management is central to the implementation of cybersecurity strategies. In order to facilitate the provisioning of connected IoT devices, paid services exist, such as the Microchip TrustMANAGER [10], which implements the Public Key Structure (PKI), allows certificates generation and update, and enables OTA firmware updates.

An example: cybersecurity in an embedded application

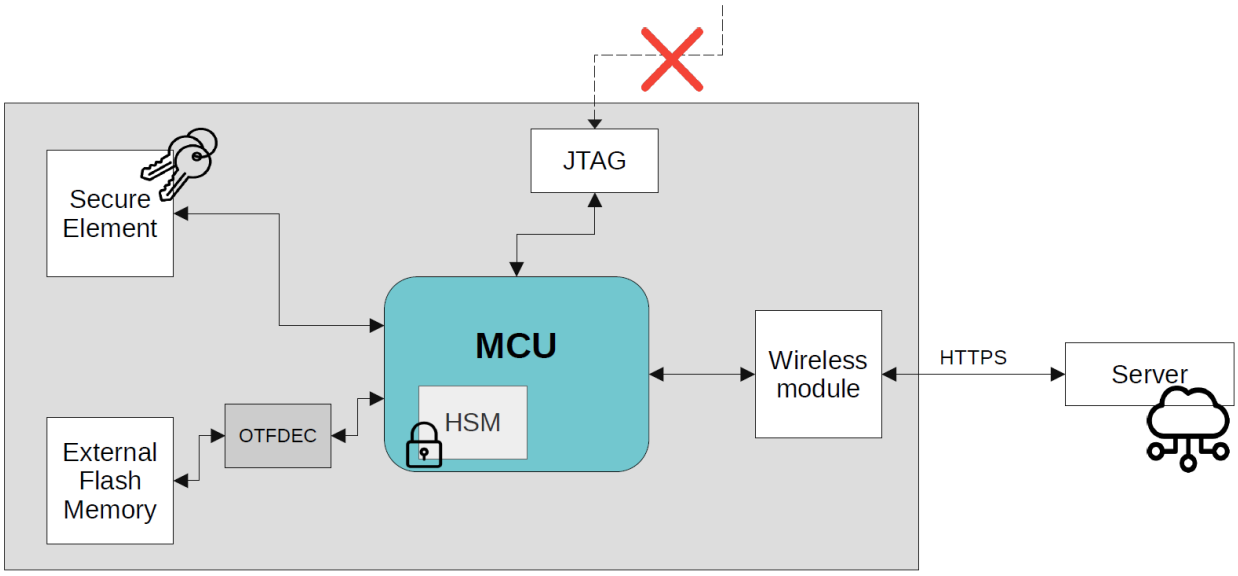

Figure 2: example of embedded application

In conclusion, an example of an application of cybersecurity standards in an embedded system is reported.

- **MCU with HSM**: manages secure boot and secure update;
- **On-the-fly Decription**: enables real-time encryption and decryption of instructions from flash memory, protecting firmware secrecy;
- **JTAG port**: access disabled by removing the JTAG channel, to avoid reading or erasing the code;
- **Secure Element**: secure and tamper-proof storage of cryptographic secrets, hardware acceleration of encryption algorithms;
- **HTTPS:** Secure connection to the Internet with HTTP over TLS, key and certificate provisioning, FOTA.

References:

1. Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).
2. Cybersecurity in medical devices: what does MDR 2017/745 require? - Teoresi Group
3. Regulation (EU) 2017/745 of the European Parliament and of the Council
4. TrustZone for Cortex-A – Arm®
5. TrustZone for Cortex-M – Arm®

6. EdgeLock Secure Enclave | NXP Semiconductors
7. PIC32CK SG/GC Arm® Cortex-M33-Based® Microcontrollers (MCUs) | Microchip Technology
8. PIC32CZ CA Arm® Cortex-M7-Based® Microcontrollers (MCUs) | Microchip Technology
9. Encryption - ACN
10. TrustMANAGER | Microchip Technology

Author: Gaia Di Federico, R&D Electronic Engineer