Secure Multi-tenant Provisioning of IoT Devices by Combining On-chip Cortex-M TrustZone with Secure Element

Dimitrios Bakoyiannis
ISCA-Lab
Hellenic Mediterranean Univ.
dbakoyiannis@hmu.gr

Othon Tomoutzoglou
ISCA-Lab
Hellenic Mediterranean Univ.
otto@hmu.gr

George Kornaros
ISCA-Lab
Hellenic Mediterranean Univ.
kornaros@hmu.gr

Marcello Coppola
STMicroelectronics
France
marcello.coppola@st.com

Abstract

In the aviation, military, self-driving control, and sensitive industrial domains, securely bringing an IoT system and its devices online is a challenging task. To connect to the network, potentially headless IoT devices must re-key and authenticate device credentials, necessitating simple scalability and flow when a trusted platform module (TPM) establishes the chain of trust with an endorsement key and secures all credentials. We present a flexible solution based on trusted boot chain of STM32WL55JC dual core microcontroller extended with STSAFE A110 secure element to enable differentiated multi-tenant services and extended level of security.

1 Secure MCU and Secure Elements

Increased attention to Industrial IoT security has guided strong standardization efforts towards enabling zero trust architecture to IoT environments[5] and the design and IETF standardization of Manufacturer Usage Description (MUD) technologies to enable a scalable and automated means to enforce device specific access control involving network switches and routers[6]. An important concern for the Industrial Internet of Things (IIoT)-based systems is to ensure chain of trust during the booting process of operating systems and trusted execution of applications. Even more challenging is implementing trusted boot and verification checks of applications executing in low-performance and limited memory embedded systems[3][2]. In this scope, hardware security modules and cryptographic hardware accelerators aid to reduce memory usage and offload the CPU from complex computation and enhance the level of security. Several TEEs have been introduced by multiple vendors and researchers for protection against untrusted OSes, such as Intel Software Guard Extension (SGX) supports protecting an application's code as well as data in an isolated virtual address space constituting the so-called enclaves[4]. ARM TrustZone splits CPU into Normal World and Secure World, as well as all other hardware resources, while recently also adopted in Cortex-M microcontrollers[1].

Additional to such secured CPUs, *secure elements* allow each service provider to host its services securely in the advanced secure element that can be activated, personalized and managed remotely to make it easy to run differentiated and loyalty-building services with absolute confidence. A secure element's ability to let third-party partners add their own use cases is one of its largest benefits, since this expands the range of secure element onboarding and security capabilities available to customers. Furthermore, since STSAFE A110 supports eight data partitions, a vendor can store up to eight key pairs per device, to perform key rotation with failsafe without issues. Key pairs and certificates are all generated in a regulatory environment (e.g., the private key can be generated by the secure element itself, not an external party), which allows auditing and an advanced level of trust.

1.1 TEE based on STM32WL55JC MCU

The STM32WL55JC is a dual core microcontroller with Cortex-M4 core (CPU1) as non-secure and Cortex-M0+ core (CPU2) as secure one. The microcontroller unit is equipped with flash, SRAM1 and SRAM2 memories, which can be protected by security and privilege at system level in addition to any privilege protection in the CPUs MCU. Security is defined in flash memory user options and privilege is defined in the Global Trust-Zone Controller (GTZC) registers. Thus, memory regions are protected from being accessed by any non-authorized bus master[5]. A hide protection area (HDP) can be defined in flash memory, which is accessible from device reset or wakeup from standby mode. To protect from any access, the hide protection area is disabled when the flash memory access control registers are set accordingly[5]. To guarantee trusted execution, signature verification is used before an application jumps to its code. We traditionally used software-oriented authentication with the MBEDTLS library which increases the flash memory requirements for the built binary significantly. We replaced the verification method with PKA hardware accelerator which not only reduced memory usage but, also, affected the CPU utilization. Fig. 1 shows the system ROM, the secure system memory that is programmed by ST during the manufacturing process and stores Root Secure Services firmware (RSS) thanks to the RSS library[5]. In particular, the user calls the CloseExitHDP() to close the flash HDP secure memory area and to

jump to the reset handler embedded within the vector table of a user application.

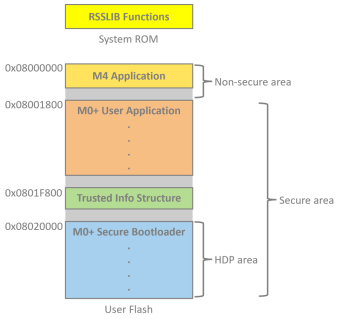

Figure 1. Flash partition and applied secure areas.

Option bytes are a set of registers that allow a user to apply configurations mostly related to the boot mode of the two cores of the STM32WL55JC microcontroller and the security of the flash and SRAM memories. The user can set secure and hide protection areas (HDP) and apply read and/or write protection (RDP, WRP) to sections of code, and select to boot a core from the system ROM, user flash or SRAM. Finally, via option bytes the address offset in flash can be set, where the M0+ core will initiate its execution.

1.2 STSAFE-A110 Secure Element

In our work, a trusted application, through the secure bootloader, is enabled to access immutable data records such as LoRaWAN credentials, administrative configurations or certificates and signatures that differentiate access rights and roles for LoRa and NFC provisioning. Such immutable data can be saved in a safe flash region, but even a reliable program running in a secure environment has the potential to inflict harm accidentally or most likely due to improper code operation. The secure bootloader must be the only entity allowed to modify critical data and trusted applications must only be able to acquire read rights. We expanded the secure application execution via attaching STSAFE-A110, a HW Secure Element (SE) that provides cryptographic services such as authentication, encryption and secure storage. The major reason for integrating the SE is to benefit from its secure storage, which only permits data alteration to authorized entities by use of MAC symmetric keys. In our case, the MAC key is shared between the STSAFE-A110 and the secure bootloader, in other words critical actions can only be granted to the latter. A solution would be to store critical data to the flash area of the bootloader but since this area is hide protected and considered as a custom ROM it must not be modifiable.

2 Implementation

The bootloader's primary job is to redirect CPU execution to a certain address offset in memory where an installed program binary is located. Both the bootloader and the trusted user application execute on the M0+ core which is the only privileged CPU allowed to access secure memories and peripherals. The M0+ core, though, can be released only by the M4 core. As a result, one more binary application is involved for the M4 core that should, at least, include the system clock configuration and the M0+ release operation. The secure bootloader uses the STM root certificate which enfolds the public key that can be used to verify the STSAFE-A110 leaf certificate. If that verification succeeds it is then clear that the secure element is trustworthy.

This certificate is available to the bootloader and is stored in the STSAFE-A110 secure data partitions. The reason for storing it in the STSAFE-A110 is that an administrative entity may choose to replace it with a fresh certificate for the trusted developer and the secure bootloader HDP secure area should not be allowed to be modified once security is enabled. In other words, storing the certificate in the HDP area is not recommended. At this point, the secure bootloader reads the developer's certificate from STSAFE-A110, parses the public key found in that certificate and extracts the Trusted Info Structure (TIS) from the flash. Based on the TIS it calculates the hash of the user application, checks that the signature is generated for this hash and finally according to the developer's public key it verifies that the signature belongs to the trusted developer.

Table 1 summarizes the cost of employing different secure element functionality integrated with the user application, while Figure 2 presents the power consumption to perform authenticaton on the M4 of STM32WL55JC.

Table 1. STSAFE A110 FUNCTIONS OVERHEAD

Code Configuration	Flash (KB)	RAM (KB)
Base System (1)	14.71	2.05
Base System + Pairing	48.17	10.91
Base System + Authentication	118.81	10.96
Base System + Data Partition	49.09	10.91
Base System + Wrap/Unwrap	48.03	10.91
Base System + All Cases Above	123.94	10.69

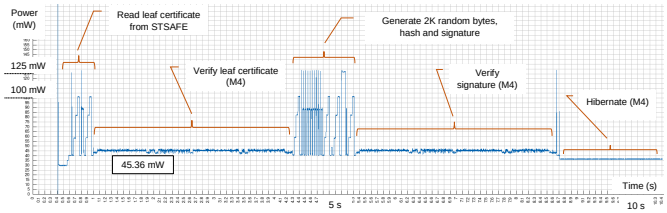

Figure 2. Power consumption for STM32WL55JC (M4) to perform authentication with the aid of STSAFE-A110

Acknowledgments

This work was supported in part by the EU H2020 Project AVANGARD under agreement 869986.

3 References

[1] ARM. Trustzone technology for armv8-m architecture, version 2.1. https://static.docs.arm.com/100690/0201/armv8_m_architecture_trustzone_technology_100690_0201_01_en.pdf.

[2] D. Bakoyiannis, O. Tomoutzoglou, G. Kornaros, and M. Coppola. From hardware-software contracts to industrial iot-cloud block-chains for security. In *2021 Smart Systems Integration (SSI)*, pages 1–4, 2021.

[3] D. Mbakoyiannis, O. Tomoutzoglou, and G. Kornaros. Secure over-the-air firmware updating for automotive electronic control units. In *34th ACM/SIGAPP Symp. on Appl. Comp.*, SAC '19, page 174–181, 2019.

[4] F. McKeen et al. Innovative instructions and software model for isolated execution. In *Proc. of the 2nd Int'l Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, 2013.

[5] "STMicroelectronics". Stm32wl5x advanced arm-based 32-bit mcus with sub-ghz radio solution, rm0453 rev 2, 2021.

[6] S. Symington et al. Trusted internet of things (iot) device network-layer onboarding and lifecycle management (draft), 2020. NIST Cybersecurity, https://doi.org/10.6028/NIST.CSWP.09082020-draft, NIST.