Silicon-Integrated Security Solutions Driving IoT Security

Stephan Spitz and Alexander Lawall
IU International University of Applied Science, Juri-Gagarin-Ring 152, 99084 Erfurt, Germany

Keywords: Silicon-Integrated Security, Internet of Things (IoT) Security, Industrial Internet of Things (IIoT) Security.

Abstract: Internet of Things (IoT) devices still miss in many cases an ability to prove their identity, verify configuration changes based on a solid root-of-trust or have a data confidentiality protection anchored in hardware. This paper describes how to bridge between service-level security functionalities and a deeply silicon-integrated security solution, which is part of a larger System-on-Chip (SoC) for the benefit of increased security. Such a bridging raises new demands regarding silicon manufacturing, the Secure Operating System design, and also the communication and management interfaces. This is because in comparison to a "classical" Trusted Platform Module (TPM), no dedicated security hardware is available. This article describes the Sytem-on-Chip security integration's impact on increasing the security level of the IoT service layer. "Integrated" refers to a secure enclave, which is no longer located on a separate chip, because it is part of the SoC of a larger device together with many other components on the same piece of silicon e.g. application/modem-processor cores, integrated memory and high-bandwidth I/O interfaces. A further aim of this paper is to create awareness about the capabilities of SoC-integrated security functions so that they can be leveraged by software designers, who are usually not deeply familiar with hardware security.

1 INTRODUCTION

Security mechanisms, which are not anchored in hardware, can be in most cases circumvented by software tools and exploits. Such tools are available for hackers at low cost and usually do not require an investment in expensive hacking hardware. This has been early recognized by initiatives such as the Trusted Computing Group and resulted in several specifications defining how hardware can be leveraged to secure the software stack. Nevertheless, separate security hardware such as a TPM is vulnerable to attacks on its external interfaces e.g. the data communication on the serial bus. Even when external secure elements raise the hurdle for attackers, the attack surface is still high with such externally accessible I/O interfaces. This changes when the secure element becomes integrated with the other building blocks on the same System-on-Chip (SoC). As a result, the interface changes from an external to an internal connection to the bus of the SoC. Consequently, the whole technology stack changes, which is built on integrated security solutions. This starts from seeding the Root-of-Trust (RoT) into the integrated security enclave up to leveraging the seeded cryptographic material in security features such as secure update management or attestation of the IoT device. This paper sketches how to close the chain from the security hardware up to the security-critical services by highlighting the necessary basic cryptographic mechanism. In comparison to many other papers, a holistic approach is sketched linking the worlds of silicon security and cyber security.

2 EVOLUTION FROM SEPARATE SECURITY CHIPS TOWARDS SILICON-INTEGRATED SECURITY

The evolution described in this section is the foundation for linking security services on the application level to deeply integrated security enclaves in a SoC. The opportunities to improve security beyond classical security chips, such as a TPM are high, but a different technology stack is required. In comparison to a separate TPM chip, an integrated security solution requires a Secure OS, which is tightly interfacing with the high-bandwidth I/O capabilities of the SoC, cf. fig. 1. Multiple applications and services, which are executed in the richOS or Real-Time OS (RTOS) on

Spitz, S. and Lawall, A.
Silicon-Integrated Security Solutions Driving IoT Security.
DOI: 10.5220/0012350200003648

In *Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP 2024)*, pages 398-402
ISBN: 978-989-758-683-5; ISSN: 2184-4356

the main CPU Cores, require concurrent security support by services of the Secure OS. As a result, the new Secure OS of an integrated solution needs to be capable of high data rates, multitasking, and concurrent process execution. Secure concurrent communication over the Internet Protocol (IP) e.g. Transport Layer Security (TLS) requires the support of asynchronous encryption and decryption processes performed in the Secure OS.

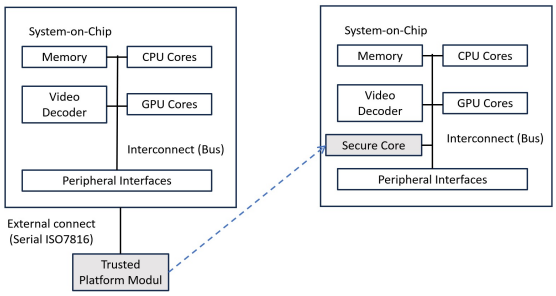

Figure 1: Separate Secure Element attached to a larger System-on-Chip vs. System-on-Chip-integrated Security.

Another difference between a TPM Firmware and a Secure OS for the SoC integration is the boot process. The SoC-integrated Secure OS is powered up together with the whole SoC firmware and richOS/RTOS. This results in a new highly security-critical dependency on the whole boot process and power management of the SoC. The integrity of the Secure OS needs to be verified during the SoC's Secure Boot to ensure that no software is loaded, that has been modified by an attacker.

In comparison to a TPM chip with integrated Non-Volatile Memory (NVM), integrated flash memory and One-Time-Programmable (OTP) memory are expensive resources on a generic purpose SoC. For this reason, many low-end SoCs rely on external flash connected with a high bandwidth interface to the SoC, which is function-wise not a big difference. From a security perspective, the Secure OS code and data are highly exposed in this external flash memory and can only be stored encrypted. The used encryption technology needs to be highly flexible and secure because flexible swap-in and swap-out processes have to be supported. Leakage of information during read and write cycles to external memory has to be avoided, especially during the execution of critical crypto routines. As a result, the Secure OS architecture has to take into consideration the en-/decryption of external storage, especially with the design of crypto routines.

Another memory-related difference is the necessity for monotonic counters in a Secure OS. A retry count for a PIN Unlock Key (PUK) (PIN can be unblocked, but not PUK) is probably the most famous kind of a monotonic counter, but there are many other security use cases, that require a certain security state that cannot be reverted. A TPM, which is a special type of Smart Card controller, implements monotonic counters in a secure NVM protected together with the secure microcontroller. Such a design is with a larger generic purpose SoC not possible. Moreover, expensive OTP memory is in a very low amount available, which is not sufficient to secure all the required state machines and atomic processes. As a consequence, new ways to implement the security feature "monotonic counters" are required on a SoC (Winbond, 2018).

Also, new ways to achieve tamper-resistance of a SoC are required, which a TPM has already per se. Similar to a Smart Card semiconductor, security-relevant structures have to be protected by active measurements such as metal shielding, a scrambled data bus, or sensors for detecting an attack. It is obvious that this bears some challenges for the chip design and also the generic silicon manufacturing process (Arm, 2018).

In addition, the generic SoC manufacturing lines have to be ramped up to create the necessary security foundation cryptographic- and security-wise, especially for the previously mentioned Secure Boot process (IAR, 2018a). The creation of the so-called Root-of-Trust (RoT) forms the security foundation for a Secure Boot and all the other later required security processes, as well as for deploying and loading the Secure OS and the required individualization and personalization processes. Since devices such as wearables, smartphones, or IoT sensors can be personalized in a much later stage, this RoT ensures a robust device identity and allows an attestation service to prove the integrity of the device's software, especially the Secure OS.

3 ARCHITECTURAL CONCEPTS

3.1 Architectural Aspects Related to a Secure OS

All the previously described differences between separate TPM and SoC-integrated security OSs make it obvious that there is an impact on OS architecture, which cannot be ignored (Spitz, 2012). One of the most obvious changes in the Secure OS architecture is the introduction of an OS kernel in conjunction with multitasking capabilities. A hardened µKernel seems to be a good fit for an integrated security solution, especially as the process isolation and minimal size contribute to security and robustness. A standard

µKernel architecture is required for the process isolation of a Memory Management Unit (MMU). With larger SoCs, a MMU is an inherent part also available for the security enclave. Smaller (IoT-)SoCs without a MMU can at least incorporate a kind of Memory Protection Unit (MPU) to support isolated memory for different kinds of services executed by the Secure OS in a pre-emptive way.

Multitasking enables secure asynchronous high-bandwidth communication with internal and external interfaces of the SoC. There are different possibilities to connect a SoC-based security enclave to other peripherals and the main processor. The OS design has to take this into account and provide drivers and protocols for this interaction. Typically, the asynchronous behavior is implemented in the form of a mailbox concept, which allows the exchange of high amounts of data between security-critical and normal processes in a dedicated memory space.

Another security-critical piece of software residing outside the Secure OS is the Secure Boot. The security foundation is formed by the Secure Boot, which contains the necessary RoT to encrypt and verify the integrity of the Secure OS during boot time. Typically the Secure OS is loaded from flash memory in an early bootstage and initialized with the cryptographic keys stored encrypted in the memory. The necessary master keys and integrity protection and verification mechanism are part of the Secure Boot (IAR, 2018b).

3.2 SoC Security Architecture

Measurements against fault injection and side channel attacks are essential for every hardware security solution, which a TPM is. Since silicon, which is manufactured in a generic line, offers fewer capabilities for such measurements, security features incorporated in the hardware and firmware IP become more important, cf. fig. 2.

Security aspects, which can be already part of the hardware IP, are strong hardware-based isolation mechanisms such as arm TrustZone™ or a separate secure processor core such as ARC SEM™ (Synopsys, 2016). In addition, security-critical SoC-based components can be isolated from normal processing components e.g. memory or I/O peripherals, which are only available in a secure execution mode.

The firmware is essential to safeguard the security-relevant components of the SoC e.g. secure memory or secure processing units. The security boundaries are defined during the boot of the platform and integrity checks for all low-level driver and firmware components are necessary. A staged secure boot has to initialize all secure processing components before the standard processing components are booted to guarantee that the standard process has no influence on the security settings, especially access to critical cryptographic keys.

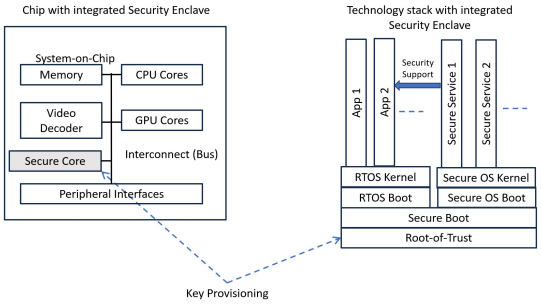

Figure 2: Stack of an Integrated Security Solution.

4 LIFE-CYCLE MANAGEMENT OF INTEGRATED SECURITY SOLUTIONS

4.1 The Role of a Root-of-Trust

A robust RoT, which is already embedded in the SoC during production, creates the security anchor for secure service life-cycle management of the whole device including the Secure OS and its applications (IAR, 2018a). The life-cycle management includes the deployment, change, and complete deletion of security-critical code and data, especially identity-related information. Such identity-related information can concern the identity of the device itself or identities for roles to manage access to the device.

Moreover, a RoT is a pre-condition for establishing a secure channel to the device, which allows the seeding of identities even when the device is in an insecure environment. Based on the RoT an authenticated and secure channel can be established, which allows the download of security-critical data even when the device is already in the field at any point later e.g. for maintenance operations. The RoT enables the necessary security binding between the device and an external trusted entity by a cryptographic handshake. Such a handshake can leverage pre-seeded asymmetric or symmetric keys and can span a key hierarchy for later identity management on the device. It is worth mentioning that the RoT is located outside the secure OS because it has to be established before the Secure OS is deployed on the device. A Secure Boot loader incorporating the RoT is responsible for the integrity of the whole SoC and all the software executed on the device. The SoC must have the necessary capabili-

ties to protect the master keys, which are part of the RoT. These are asymmetric keys, i.e. public-private key pairs, and they are applied as described in the following chapters. Public keys require integrity protection because this information needs not be kept confidential. Private keys should be device individual and highly confidentiality protected.

4.2 Life-Cycle Management Operations

The secure life-cycle management of a device contains the following aspects:

- Verification of the SoC/device identity and integrity also from remote, cf. (Sundar et al., 2019)
- Integrity protection of code and data during loading and runtime, especially protection of the Secure OS, cf. (Wang et al., 2019)
- Secure remote download of a user identity (personalization)
- Secure disabling of the SoC/device e.g. for end-of-life, over production control, and grey market prevention
- Establishment of end-2-end secure communication channels for any kind of life-cycle operation e.g. configuration or firmware update
- Authentication and authorization of user access, configuration changes, download of code, or any other administrative actions
- Delegation of rights and permission, especially delegation of control to third parties e.g. with the change of ownership

For a more detailed description the following example processes, describing how a robust RoT can be leveraged, have been chosen:

- a) Identification of the SoC or device
- b) Authorisation of an administrative action, code or data
- c) Authentication of the issuer of an administrative action
- d) Confidentiality protection of code, data, and administrative commands

4.2.1 Identification of a System-on-Chip

In scenario a), the RoT can comprise an identity and a private key, which is securely stored in the bootloader of the SoC during manufacturing, cf. fig. 3. Now the Secure OS in the device gets a request to authenticate the device and attest its identity. This request is in conjunction with a random number, which the Secure OS encrypts by using the private key stored in the RoT. The verifying entity outside the device can decrypt the response with the help of the public key. The device has proven its identity if the result is the same as the previous random challenge.

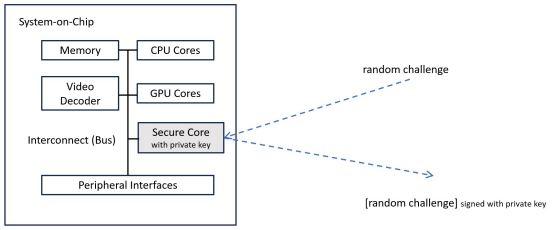

Figure 3: Device Identity Verification.

4.2.2 Authorisation and Authentication of an Administrative Action

In scenarios b) and c), it is vice versa i.e. the RoT holds a public key, cf. fig. 4. Now the external entity has a private key and can authorize an administrative action, deploy code or read/write data in the Secure OS. The issuer of the administrative action is in this scenario automatically identified, because the private key can be uniquely assigned to a person, legal entity, IT system, etc.. This assignment is done in the form of a Public Key Infrastructure (PKI). It is also worth mentioning that in this case, the public key on the device has to be protected from modifications and exchange, but confidentiality protection is not required. The RoT has just to ensure integrity protection and not only the Secure OS on the device can have read access to this public key.

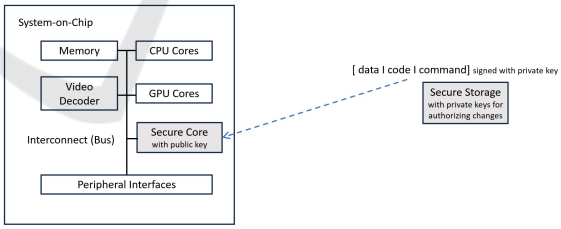

Figure 4: Authorisation of Remote Management.

4.2.3 Confidentiality Protection of Data

Scenario d) requires a hybrid encryption scheme i.e. a combination of a symmetric and an asymmetric encryption, because asymmetric algorithms are performance-wise not suited to encrypt larger amounts of data, cf. fig. 5. In this combination, the symmetric key is just temporarily generated and sent with the encrypted data to the device. The symmetric key itself is again encrypted with the public key of the device. The Secure OS has access to the private key of the device stored in the RoT and can in the first

step encrypt the ephemeral symmetric key and in the second step encrypt the data using this symmetric key. This scenario is highly relevant when personalization data is sent to the device e.g. a subscriber profile for an integrated SIM (ETSI, 2017) or payment credentials for an Integrated Secure Element (iSE).

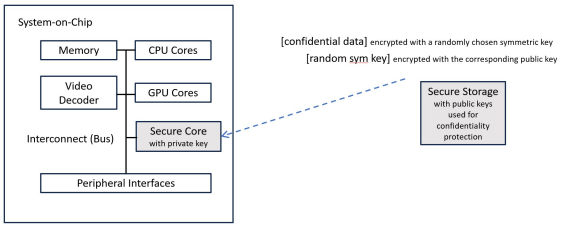

Figure 5: Confidentiality Protection.

5 CONCLUSION

Due to the advances in silicon-integrated security, a new generation of deeply embedded security solutions is rising, which will have an impact on the whole industry involved e.g. personalization processes are no longer bound to secure premises of highly specialized manufacturers such as the Smart Card industry. The value chain is changing and different technologies are becoming more relevant for integrated security solutions such as security Hardware IP or a robust Root-of-Trust deployed in the system from the beginning.

Moreover, paradigm shifts with security services, OS- and silicon-security-architecture offer opportunities for new businesses, but also replace existing technologies and processes in the classical Smart Card industry. This journey has just started and will offer the end-user more convenient and robust security solutions in the end. End-users currently see the tip of this iceberg with the disappearance of SIM cards, payment capabilities in smartphones, remote feature enablement with cars, and many other security-critical IoT functions. Of course, the robustness of smartphones and IoT devices against any kind of attack is significantly increased by a good security design, which is anchored in the silicon. Security design aspects will move in the foreground with an integrated security enclave because security moves tighter to the central processing of the data in the main CPU. These advantages can be only leveraged when already considered in the service design.

Last, but not least is worth mentioning that integrated security enclaves are more vulnerable to sophisticated side-channel attacks than tamper-resistant external secure elements. However, recent developments show that is possible to implement mechanisms to achieve tamper resistance already by the silicon (Arm, 2018). However, this is an additional research topic, which makes no difference for the security chain, which has been sketched in this paper.

REFERENCES

Arm (2018). Cortex-m35p a tamper-resistant cortex-m processor with optional software isolation using trustzone for armv8-m. In *https://developer.arm.com/products/processors/cortex-m/cortex-m35p*. Arm.

ETSI (2017). iuicc poc group primary platform requirements, approved release. In *https://www.gsma.com/newsroom/wp-content/uploads/UIC.03_v1.0.pdf*. ETSI.

IAR (2018a). Building a supply chain of trust: Understanding secure mastering. In *https://www.iar.com/support/resources/articles/secure-mastering*. IAR.

IAR (2018b). Establishing a supply chain of trust: Start with secure development. In *https://www.iar.com/support/resources/articles/secure-development*. IAR.

Spitz, S. (2012). Mobicore® secure os for arm® trustzone® soc. In *https://prezi.com/rgrvv8vv-t4s/mobicore-secure-os-for-arm-trustzone-soc/*. Prezi.

Sundar, S., Yellai, P., Sanagapati, S. S. S., Pradhan, P. C., et al. (2019). Remote attestation based software integrity of iot devices. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–4. IEEE.

Synopsys (2016). Designware arc sem security processors. In *https://www.synopsys.com/dw/ipdir.php?ds=arc-sem*. Synopsys.

Wang, W., Zhang, X., Hao, Q., Zhang, Z., Xu, B., Dong, H., Xia, T., and Wang, X. (2019). Hardware-enhanced protection for the runtime data security in embedded systems. *Electronics*, 8(1):52.

Winbond (2018). Winbond introduces trustme™ secure flash memory aligned with platform security architecture from arm. In *https://www.winbond.com/hq/about-winbond/news-and-events/news/news00452.html*. Winbound.