Hardware Design of an Advanced-Feature Cryptographic Tile within the European Processor Initiative

Pietro Nannipieri, *Member, IEEE*, Luca Crocetti, Stefano Di Matteo, Luca Fanucci, *Fellow, IEEE,* and Sergio Saponara, *Senior Member, IEEE*

Abstract—This work describes the hardware implementation of a cryptographic accelerators suite, named Crypto-Tile, in the framework of the European Processor Initiative (EPI) project. The EPI project traced the roadmap to develop the first family of low-power processors with the design fully made in Europe, for Big Data, supercomputers and automotive. Each of the coprocessors of Crypto-Tile is dedicated to a specific family of cryptographic algorithms, offering functions for symmetric and public-key cryptography, computation of digests, generation of random numbers, and Post-Quantum cryptography. The performances of each coprocessor outperform other available solutions, offering innovative hardware-native services, such as key management, clock randomisation and access privilege mechanisms. The system has been synthesised on a 7 nm standard-cell technology, being the first Cryptoprocessor to be characterised in such an advanced silicon technology. The post-synthesis netlist has been employed to assess the resistance of Crypto-Tile to power analysis side-channel attacks. Finally, a demoboard has been implemented, integrating a RISC-V softcore processor and the Crypto-Tile module, and drivers for hardware abstraction layer, bare-metal applications and drivers for Linux kernel in C language have been developed. Finally, we exploited them to compare in terms of execution speed the hardware-accelerated algorithms against software-only solutions.

Index Terms—AES, ECC, RNG, SHA, RISC-V, EPI, Cryptoprocessor, Hardware, Security, Root of Trust, Chain of Trust, Secure boot

1 Introduction

Common sense can easily state that cybersecurity is a fundamental aspect of everyday life: due to the enormous amount of data and the interconnections between devices and networks, the sources of threats and vulnerabilities aimed at stealing, manipulating and tampering with information can become countless. In other words, each data transfer, whatever the nature, the model and the used medium, can bring a security threat to any of the systems or subsystems interacting with it, and, for extension, to any other entities involved in the chain that can be traced between the origin and the destination of the whole life cycle of such data transfer.

The European Processor Initiative (EPI) [1] is a project born from a consortium of 28 partners, both academic and industrial. The aim of the EPI project is to create a new family of European low-power processors for Big Data, supercomputers and automotive markets, and offer advanced performance on High Performance Computing (HPC) applications and other emerging ones, such as machine learning.

Considering the markets targeted by this project, security plays a fundamental role, especially in the case of automotive one, for which also safety-critical aspects have to be carefully managed. Therefore, a strategy has been required to develop a General Purpose Processor (GPP) able to support the requirements of relevant security standards and certifications, without compromising the computational capabilities.

The adopted solution consisted in creating an isolated and trusted zone that is physically separated from the main processor, and that is dedicated to performing all and only the security-specific functions. This approach granted the possibility to optimize the primary processor for the execution of high-performance computing operations while improving the robustness of the security zone with specific elements dedicated to ensuring the trust of all its parts.

To offer both flexibility and high performance, the secure zone has been equipped with both hardware and software resources, integrating microcontrollers and hardware accelerators in independently replicated architectures, not only increasing the computational capabilities by parallelism but also strengthening the system's safety. The trust of this isolated zone is granted by a layered structure which is built upon the power-on of the EPI GPP, starting from an implicitly trusted component which is called Root of Trust (RoT) [2]. Such an element triggers the execution of a secure boot sequence which, through different stages, authenticates level by level any other component within the boundaries of the secure zone. In addition, dedicated resources have been foreseen to protect the secure zone during the whole lifetime of the EPI chip.

This work describes and illustrates the research activities related to the hardware design and implementation of a suite of cryptographic accelerators capable of supporting and offering a complete and general-purpose set of cybersecurity services with high-security features for long-term protection, including:

- *P. Nannipieri, L. Crocetti, S. Di Matteo, L. Fanucci and S. Saponara were with the Department of Information Engineering,University of Pisa, Italy E-mail: pietro.nannipieri@unipi.it*

Manuscript received XXX; revised XXXX.

- an Advanced Encryption Standard (AES) engine integrating both AES-128 and AES-256 cyphers, to provide at least 128 bits of security strength for both classical security and post-quantum security;
- an Elliptic-Curve Cryptography (ECC) engine supporting operation on elliptic curves with a width of 256 bits and greater than 512 bits, to offer the same security strength of symmetric-key counterpart, at least in terms of classical security. The first standardisation stage for Post-Quantum Cryptography (PQC) asymmetric cryptography algorithms by NIST has recently been released; a future version of the CryptoTile will include support for those algorithms by extending the of the RISC-V processor coupled to the CryptoTile, as preliminary investigated in [3].
- a Secure Hash Algorithm (SHA) engine generating at least 256-bit and 384-bit digests, accordingly to the minimum requirement of 128 bits of strength. It supports both SHA-2 and SHA3 algorithms, for classical and post-quantum security, respectively;
- a True Random Number Generator (TRNG) engine able to meet security requirements for cryptographic applications;

The main innovations of our contribution focus on the development of a Crypto-Tile, used as a fundamental part of an Hardware Secure Module (HSM), paying attention to both features and performances: each one of the implemented cryptographic engines outperforms in terms of performances and efficiency all other available solutions in the state of the art, presenting the first data available for synthesis on a 7nm silicon technology. On top of that, the design of the entire Crypto-Tile introduces innovative features, all natively supported in hardware, such as key management, clock randomisation and access privilege mechanisms. The hardware support of these features, which are usually implemented in software, increases the security, power efficiency and overall performance of the system. This set of cryptographic engines has been developed and integrated into a higher-level module, in addition to other resources for storage, management and protection of security-critical data and assets, bringing to the implementation of a hardware coprocessing unit for the Security Domains of the Security Subsystem of EPI GPP: such coprocessor took the name of Crypto-Tile.

In Section 2, we will analyse the state of the art with the cryptography adopted by the main processor manufacturer and, after that, we present the design choices adopted for the EPI Crypto-Tile. In Section 3 we detail the design process of our Crypto-Tile, including its verification, synthesis and implementation. In Section 4 we present the demoboard we designed to assess the performance of our Crypto-Tile, together with the results achieved in terms of computational effort to compute the target algorithms. Finally, in Section 5 we draw our conclusion on the work carried out, focusing on the main innovation achieved.

2 The Crypto-Tile

2.1 EPI GPP architecture

The usage of hardware acceleration modules becomes fundamental: indeed, much of the energy efficiency and performance improvements in modern digital systems are attributable to their inclusion, as highlighted by [4], [5]. Also, other GPP manufacturers and developers, basing the security functionalities on hardware RoTs, integrated them in their systems [6], [7].

In further support of this, reference can be made also to the work presented in [8]. This paper reports the results in terms of throughput and power efficiency of the comparison between two different implementations for three security algorithms: one solution foresees the usage of hardware accelerators, hypothesizing their adoption as coprocessing units of a microcontroller, the other one instead relies only on software resources. The selected algorithms constitute fundamental building blocks for high-security applications, and in one case they have been fully implemented in hardware, while the other one has employed the OpenSSL library [9], which is a robust, commercial-grade and full-featured toolkit for security protocols.

The fact that cryptographic hardware acceleration is becoming fundamental in modern processing systems, pushes for the development of complete HSM. One example of HSM is the CryptoManager Root of Trust (CMRT) Intellectual Property (IP) by Rambus, [10]. It includes a 32-bit RISC-V processor, a Read Only Memory (ROM) unit, hardware resources dedicated to accelerating the security algorithms and managing the security assets, and also private buses and interfaces for the integration of One-Time Programmable (OTP) memories and Secure Random Access Memorys (SRAMs) within the secure zone, and it is developed to assist general-purpose processing units for Internet of Things (IoT), automotive space, connectivity and sensors applications. Several examples can be found in the most important GPP manufacturers and developers as ARM, Intel and AMD. For instance, ARM proposes the so-called TrustZone [11], integrated inside the Cortex-A, [12], and Cortex-M processors, [13], that are high-performance and power-optimized processors for applications such as Artificial Intelligence (AI), machine learning and automotive. Based on hardware security modules to manage cryptographic operations, keys storage and prevent unauthorized access to sensitive resources, the TrustZone is aimed to form a Trusted Execution Environment (TEE) by enabling the main processing unit to execute both general-purpose code and secure software in a time-sliced fashion. The partition of resources dedicated to security is then extended at the software level, by dividing it into two zones that are called, respectively, the Normal World and the Secure World.

A similar solution is implemented also by Intel with its Software Guard eXtension (SGX), [14], [15], [16], [17], integrated on the 3^{rd} generation of Xeon scalable processors [18]. This family of high-performance and low-power processors features a variable number of cores (from 8 up to 40) with frequencies from 2.2 GHz for applications such as HPC, Big Data, AI and networking for cloud-optimized, 5G-ready infrastructures. By exploiting hardware modules and dedicated extensions of the instruction set architecture for cryptographic primitives, [19], such cores allow implementing secure software code thanks to the partitioning of security-sensitive code and data into a so-called *Enclave*, which is executed in a protected region of the CPU protected region.

The strategy used by AMD instead differs from the ones of Intel and ARM, relying on full isolation at the hardware level. AMD integrates a subsystem entirely dedicated to the security and named Platform Security Processor (PSP), [20], that consists of a 32-bit microcontroller ARM Cortex-A5, isolated on-chip ROM and SRAM, an OTP memory for platform-unique key material, local registers, a coprocessor dedicated to the acceleration of security functions, and interfaces to interact with the system memory, input/output and configuration registers. Similar developments have also been made for a heterogeneous system to accelerate cryptography functions using dedicated HSM on commercial devices, e.g. [21].

Based on this investigation, it is possible to make a brief comparison between the several solutions based on the hardware RoT approach, as reported in Table 1, from which it can be noted that there are two main strategies. The first is the complete isolation at the physical level of the security functions which are executed and offered only by a dedicated zone or subsystem of the chip (i.e. the case of AMD PSP, GPP and Rambus CMRT, once integrated into a system). The second strategy is the execution of secure functions using the main processing unit(s) that run also the general purpose software code (i.e. the case of ARM TrustZone and Intel SGX), for which the separation between the secure zone and the non-secure zone of the chip is made at the virtual level of the running software, by authorizing or denying access to the hardware resources and assets dedicated to security. If this latter one allows saving area on silicon and physical resources, the former one allows further optimize the main processing unit(s) for general-purpose software in terms of performance and efficiency.

The EPI CryptoTile approach from the user system operation point of view is to provide a physically isolated portion of silicon capable of hardware accelerating security-based operations, providing two classes of services to its users: I) Secure Boot: the availability of secure RoT allows the EPI processor to be equipped with a secure boot routine, a feature common to all state-of-the-art processors. The realisation of the RoT in a physically separate silicon with dedicated hardware acceleration increases the security of the service. II) Cryptographic services on request: thanks to dedicated communication infrastructures with the non-secure area of the chip (e.g. Mailboxes) the CryptoTile is capable to provide hardware-accelerated cryptographic primitives, with hardware-implemented security features (e.g. keys storage and management, restricted and controlled access to secure assets, generation of high-quality random numbers). In the following sections, we will present in detail the services introduced above, with a focus on their architecture and performance.

Being the purpose of the EPI project the development of a processor with exascale computing capabilities for HPC applications, and also with high-security requirements, the physical isolation between the secure and non-secure zones of the chip is best suited to meet the requirements of both performance and security. Indeed, on one hand, it allows to optimize and enhance the main processing unit(s) for general-purpose computation tasks, while on the other hand, it lets the isolation the security-critical assets and resources and strictly regulates access to them. For instance, assuming the usage of the secure zone also assists the execution of a secure boot routine, such a solution permits to store hardware secret keys at boot time and permits the software of the non-secure zones to use them for security services, but without allowing direct access to them and their value. From this, it can be drawn out the high-level outline of the EPI GPP architecture that is illustrated in Figure 1, and that counts: ARM multi-core processors (e.g. 36), playing the role of main computing units for high-performance operations; a secure zone named Security Subsystem (the green box in Figure 1), which provides the hardware RoT; units for power management, debugging and testing; additional hardware acceleration modules (for the most intensive application-specific functions); peripherals interfaces and controllers for external modules; a global Network-on-Chip (NoC) used to link all the previous elements, featuring high-bandwidth data transfers.

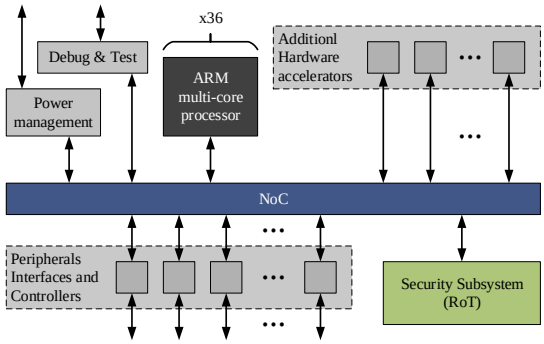

Fig. 1. Simplified outline of EPI GPP architecture: the GPP is an heterogeneous ecosystem formed by different elements, each dedicated to playing a specific role, and that communicate together by means of a NoC.

In the following, we will focus on the Security Subsystem, its architecture and its feature.

2.2 EPI Security Subsystem

The design of the Security Subsystem of EPI implements a physically isolated (hardware) RoT that supports also a secure boot routine. The chosen approach is a mixed solution integrating both software resources (for flexibility) and hardware resources dedicated to accelerating the most computing-intensive parts or routines of the security services. Such methodology better fits the requirements of high performance and security robustness: the hardware resources enhance the security level of critical assets and boost the execution speed of security primitives, while the software resources can offer to the non-secure zone higher-level security services that are built on top of such primitives.

Concerning the software resources to be integrated within the Security Subsystem, a lot of effort in the industrial and academic community is currently granted to the development of secure RISC-V processors to execute it: many contributions (e.g. [22]), including a dedicated working group, are investigating the various possibilities to include security features in the RISC-V processors [23], representing one of the major ongoing developments.

Concerning the hardware resources to accelerate the security primitives, in this work we propose the implemen-

	Rambus CMRT	ARM TrustZone	Intel SGX	AMD PSP
Implementation strategy	Discrete	Integrated within the system	Integrated within the system	Integrated within the system
Isolation approach between secure and non-secure zones	Physical isolation	Virtual isolation	Virtual isolation	Physical isolation
Platform(s)	SoC IPs	Cortex-A, Cortex-M processors	Xeon processors (3^{rd} generation)	Ryzen PRO 5000 processors

TABLE 1
Comparison between major hardware RoT-based security solutions.

tation of a comprehensive suite of cryptographic algorithms able to offer all the basic and most diffused security services, such as confidentiality, data integrity, authentication (of both data and sources) and non-repudiation, including AES, SHA2, SHA3, ECC and RNG, together with logic resources to support system-level advanced security features, such as key management, clock randomisation and access privilege mechanism. Such a suite of cryptographic accelerators took the name of Crypto-Tile and was aimed to empower the RISC-V processor with cutting-edge hardware security features to make able the EPI secure zone support more features than the ones of the previously illustrated solutions (AMD, ARM, INTEL, RAMBUS). To complete the list of architectural resources required for the implementation of the secure zone, they shall be included also an OTP memory for the storage of (an encrypted version of) the secure boot code and a PUF to enable the protection and the verification of the code for the secure boot routine. The usage of Physical Unclonable Function (PUF)-based techniques for secure boot applications can be found in [24], [25], [26], showing schemes and protocols also for the upgrade of firmware in remote IoT devices (similarly to the Firmware Over-The-Air (FOTA) mechanism) or secure boot of bitstreams for programming Field-Programmable Gate Array (FPGA) devices, and all of them confirm how the employment of PUFs is a fundamental element to uniquely authenticate the identity of a device and establish a chain of trusted code and hardware elements. This is because a PUF module is a reliable method to derive a unique physical fingerprint that is extremely difficult to be cloned onto another same hardware component [27]: it essentially relies on a challenge-response protocol [27], i.e. when a stimulus is applied to it (challenge, C), it reacts in a certain way (response, R), by providing a unique response, that varies chip by chip, for the same challenge that is applied to any of the chips. As will be better detailed in the following sections, OTP and PUF are modules that will be imported from third-party libraries while all the remaining has been developed during this work and will be discussed in detail in the following sections, focusing on all the hardware-related security aspects.

From this, a detailed outline of the internal architecture of the Security Subsystem is then reported by Figure 2, which has been developed with the redundancy of logic resources to increase its robustness against flaws and failures, on one hand, and to increase the performance thanks to the parallelism.

Regarding Figure 2, the Security Subsystem of the EPI chip consists of a compact microcontroller named Secure

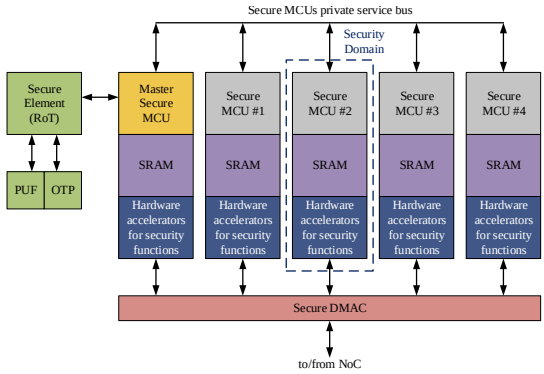

Fig. 2. Preliminary outline of Security Subsystem within EPI GPP.

Element (SE), and that is coupled to the OTP memory and a PUF module, Secure MicroController Units (MCUs), SRAMs and hardware modules to accelerate the security algorithms, plus a Direct Memory Access Controller (DMAC) for connecting the Security Subsystem to the NoC of the EPI GPP and perform high-bandwidth data transfers. The DMAC exploited in the final chip is a commercially available one from ARM, that we substituted with the one provided by Xilinx for our internal test as presented later on. The SE is an additional software resource (i.e. a stand-alone microcontroller) that is part of the RoT because its only function is to load the boot code from the OTP, possibly accessing also the PUF and the Crypto-Tile to decrypt and validate the content of the OTP (decryption and/or verification of signature and/or integrity verification), put it in the SRAM of one of the Secure MCU (which is what for this reason is called Master Secure MCU) and configure it. Once this is done, it exhausts its function and never does anything else until the system is reset. After that, the Master Secure MCU takes over and performs what is called the second boot phase (enabling the other Secure MCUs, etc.).

The triplet formed by Secure MCU, SRAM and hardware acceleration module takes the name of Security Domain, and it is replicated 5 times offering a total of 5 independent Security Domains. This approach not only improves the performance thanks to parallel computing but also enhances the safety of the system because, in case of failure of one (or more) of a Security Domain, the other ones remain available to be used and continue to execute the security services. The 5 Secure MCUs communicate with each other using dedicated private service, and in addition one of these Secure MCUs is privately connected to the SE, which is responsible for providing the RoT, and for this reason, it is

labelled as Master Secure MCU (the yellow one in Figure 2) and the related Security Domain as Master Security Domain.

In the following, we present how such building blocks enable secure boot and we focus on the architecture of the Secure Domain, the inner core of the EPI Crypto-Tile and this work.

2.2.1 Secure boot and construction of the Chain of Trust

The RoT assets (i.e. SE, OTP and PUF) permit to perform of a secure boot routine enabling all the Security Domains and forming a Chain of Trust that offers the security services to the rest of the EPI GPP, i.e. the non-secure zone, as illustrated in Figure 3:

- upon chip power-on, the SE loads the content of the OTP memory inside the SRAM of Master Security Domain, and it uses a challenge C_K to retrieve the response $R_{K\#<n>}$ and thus decrypt the boot code, by using the resources of the Master Security Domain such as the hardware acceleration module (Figure 3b and Figure 3c);
- once the boot code is decrypted and safely stored inside its corresponding SRAM (first boot stage), then Master Secure MCU manages the next boot stages to enable and configure the other 4 Security Domains (Figure 3d and Figure 3e).

The challenge C_K will be the same for any EPI chip, while the corresponding response $R_{K\#<n>}$ will be different chip by chip, thanks to the properties of PUF module. This aspect, in addition to the restricted access to the OTP memory (SE and Master Secure MCU) strengthen the security assets of the EPI chip (secret, certificate, configuration, security policies, life-cycle, ...) against the impact of the bug, applicative malicious code, and external attackers that, even if able to violate one of the EPI chips and retrieve some of the security-critical assets, would not be able to exploit the same information to attack another EPI chip.

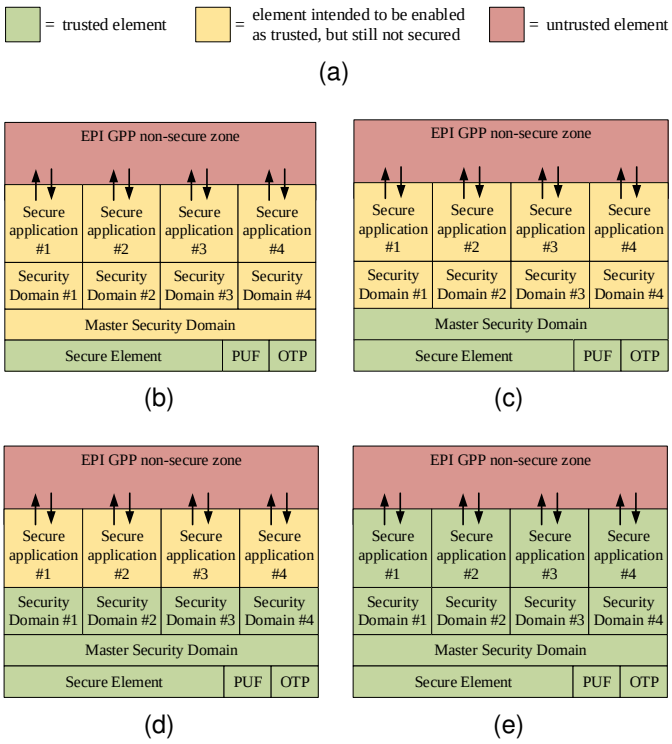

Fig. 3. Secure boot sequence and construction of Chain of Trust in EPI Security Subsystem. According to the legend (Figure 3a), at power-on of EPI chip RoT (i.e. SE, OTP and PUF) is enabled (Figure 3b and it is the only (implicitly) trusted element that performs the first stage of the secure boot, enabling the first layer of the Chain of Trust, i.e. the Master Security Domain (Figure 3c. In the next boot stage, the Master Security Domain enables the other Security Domains (Figure 3d, that become part of the Chain of Trust, and finally each of the Security Domain can provide trusted security services to the rest of the EPI GPP (Figure 3e).

2.2.2 EPI Security Domain

Continuing with the description of the security strategy of EPI chip, Figure 4 shows the internal architecture of a Security Domain. As illustrated in Figure 4, the elements composing a Security Domain are linked together by different buses, each one with a different function, and such buses are connected by specific bridges. The Configuration bus is a 32-bit Advanced eXtensible Interface 4 (AXI4) bus used by the Secure MCU (which is based on an implementation of a RISC-V processor) to configure the block dedicated to the hardware acceleration of security functions and to connect to the Secure MCU private service bus, for connecting to the other Secure MCUs; in case of Master Security Domain, it establishes also the links between the Master Secure MCU, the SE, the OTP memory and the PUF. The Advanced Peripheral Bus (APB) bus to/from Control Network, which is not present in the outline of the Security Subsystem (Figure 2) for simplicity, constitutes an access point for the environment external to the Security Subsystem for issuing requests of security services. For each Security Domain, one of the 36 ARM multi-core processors shown in the architecture of EPI GPP (Figure 1) can communicate with the Secure MCU using that bus. This depends on the fact that the main resources of the EPI chip are physically distributed in 4 quadrants. The Data bus is a high-speed 128-bit AXI4 bus that provides access to the GPP NoC, as well as connects the DMAC to the hardware acceleration module and the SRAM, to achieve better performance in terms of throughput. Lastly, a dedicated APB bus privately links the Secure MCU to the DMAC to allow the former one to configure and manage the latter one. Again to improve the safety of the system and also to allow the re-usability of this architecture, in case the DMAC is not available or shows a failure, the Configuration bus can be used by the

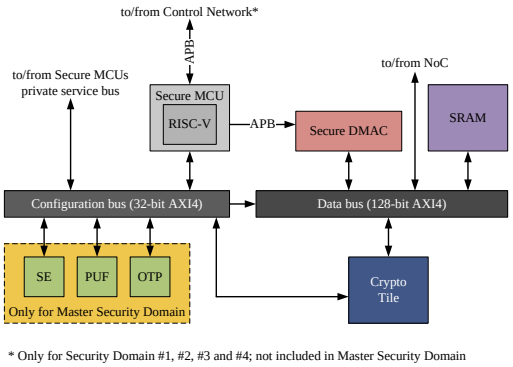

Fig. 4. Security Domain composing the EPI Security Subsystem.

Secure MCU also to transfer data without compromising the trusted zone, because the Configuration bus and each entity connected to it are confined inside this zone.

The Crypto-Tile integrates the cryptographic algorithms and functions that follow. With regard to the class of symmetric-key cryptography, it supports the AES cypher, for both 128-bit keys (i.e. AES-128 version) and 256-bit keys (i.e. AES-256 version), in the modes of operation:

- Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher FeedBack (CFB), Output FeedBack (OFB) and Counter (CTR), for data confidentiality;
- Cipher-based Message Authentication Code (CMAC), for data integrity and authentication of integrity;
- Counter with CBC-MAC (CCM) and Galois/Counter Mode (GCM), for data confidentiality, integrity and authentication;
- XEX-based Tweaked-codebook mode with ciphertext Stealing (XTS), for disk storage encryption and decryption;

Concerning the class of public-key cryptography, the Crypto-Tile embeds logic resources for finite field arithmetic operations on the 256-bit elliptic curve *secp256r1* (or National Institute of Standards and Technologies (NIST) P-256) and the 521-bit elliptic curve *secp521r1* (or NIST P-521), thus offering the same levels of security strength as the AES cypher, i.e. 128 and 256 bits, respectively, and support the following functions and schemes:

- Point Addition (PA), Point Doubling (PD), Point Multiplication (PM) and Point on Curve (PoC), to perform, respectively, the addition, the doubling and the multiplication by a scalar of points on the elliptic curves, and to check if a point belongs to an elliptic curve or not;
- Key pair generation, for the generation of both private and public keys, and Public key generation, for deriving the public key corresponding to a private key;
- generation and verification algorithms of Elliptic-Curve Digital Signature Algorithm (ECDSA);

In addition, the ECC functionalities of Crypto-Tile allow to create of hardware-assisted software drivers and functions for other ECC schemes such as Elliptic-Curve Diffie-Hellman (ECDH) and Elliptic-Curve Menezes–Qu–Vanstone (ECMQV), for keys exchange, and Elliptic-Curve Integrated Encryption Scheme (ECIES), for data confidentiality, by accelerating in hardware the most computationally intensive parts of these schemes.

For the class of hash functions, the Crypto-Tile integrates both the SHA2 and SHA-3 algorithms supporting all the digest sizes, i.e. 224, 256, 384 and 512 bits, i.e.:

- SHA2-224, SHA2-256, SHA2-384 and SHA2-512;
- SHA-3-224, SHA-3-256, SHA-3-384 and SHA-3-512.

This is because the digest size of 256 and 384 bits are required to provide the minimum accepted level of security strength, i.e. 128 bits, in the case of classical security and post-quantum security, respectively, and for SHA2 functions the algorithms to generate a digest of 224 and 512 bits are the same as the ones for generating 256 and 384 bits. Similar consideration can be made also in the case of SHA-3 algorithm which employs the same function for each digest size, the only difference concerns the length of the input data block. Moreover, the Crypto-Tile embeds also dedicated resources to assist in hardware-software drivers and/or functions implementing the keyed-Hash Message Authentication Code (HMAC) scheme.

Finally, concerning the cryptographic class of Random Bit Generators (RBGs), the Crypto-Tile integrates a Cryptographically Secure Pseudo-Random Generator (CSPRNG) composed by a standard-cells TRNG entropy source module whose output is used to seed a hash-based Deterministic Random Bit Generator (DRBG) exploiting a SHA2-256 engine, thus offering a security strength of 256 bits; in addition, it is present also a lightweight Pseudo-Random Number Generator (PRNG) that produces an output stream of 32-bit random numbers, continuously available on-demand, to be used in applications that require high speeds or low latencies and that can accept a low level of entropy.

All the cryptographic algorithms and functions offered by the Crypto-Tile are implemented in hardware and are compliant with the latest versions of revisions of the specifications defined by the corresponding reference standards, as reported by Table 2.

Furthermore, the cryptographic functions above integrated within the Crypto-Tile can be used, maybe also in combination, for supporting and improving hardware-assisted drivers and/or software routines implementing higher-level protocols, such as Transport Layer Security (TLS) (standard Request For Comments (RFC) 8446 - 2018); Secure SHell (SSH) (standard RFC 4253 - 2006; MACsec (standard Institute of Electrical and Electronics Engineers (IEEE) 802.1AE – 2018); Internet Protocol security (IPsec) (standard RFC 6380 - 2011); Wireless Access in Vehicular Environments (WAVE) (standard IEEE 1609 – 2016); Electronic Signatures and Infrastructures (ESI) (standard European Telecommunications Standards Institute (ETSI) Technical Specification (TS) 119 312 - 2021; Intelligent Transport Systems (ITS) (standard ETSI TS 103 097 - 2020.

Cryptographic algorithm	Reference standard
AES-128, AES-256	NIST FIPS 197
ECB, CBC, CFB, OFB, CTR	NIST SP 800-38A
CMAC	NIST SP 800-38B
CCM	NIST SP 800-38C
GCM	NIST SP 800-38D
XTS	NIST SP 800-38E
ECC functions	NIST FIPS 186-4
	SECG SEC 1
	ANSI X9.62
SHA2	NIST SP 800-38C
SHA-3	NIST SP 800-38D
HMAC	NIST SP 800-38E
Entropy source module (of CSPRNG)	NIST SP 800-38C
DRBG mechanism (of CSPRNG)	NIST SP 800-38D
CSPRNG	NIST SP 800-38E

TABLE 2
Reference standards of Crypto-Tile cryptographic algorithms.

2.3 Crypto-Tile Specification

Figure 5 shows the outline of the internal architecture of Crypto-Tile, that mainly consists in a 32-bit AXI4 interface (I/F), which is Slave Memory-Mapped, for accessing each register of Crypto-Tile through the Configuration bus; a Global Management Unit, for the global configuration (Cfg), control (Ctrl) and status of Crypto-Tile; 4 independent cryptoprocessors, one for each class of cryptographic algorithms, i.e. AES, ECC, SHA and Random Number Generator (RNG), and that constitute each a coprocessing unit for the main processor they are connected to (notably the Secure MCU or the SE); each cryptoprocessor integrates local registers for configuration, control and status, an Finite State Machine (FSM), to manage the Cryptographic Operation (CO), local registers for data (input and output) and a cryptographic engine for the acceleration in hardware of algorithms and functions; in addition, the AES cryptoprocessor and the ECC cryptoprocessor embed local logical resources for the key slots, respectively, for the symmetric keys and the pairs of private and public keys; 4 (independent) 128-bit AXI4 interfaces, that are Slave Memory-Mapped, one for each cryptoprocessor, to provide the access to the data registers for high-bandwidth transfers.

Concerning the AXI4 interface to the Configuration bus, it can be accessed only by the Secure MCU when using the Supervisor or User Privilege Level, by distinguishing among the (admitted) Privilege Levels at address level: i.e. each Privilege Level of Secure MCU can access only a specific region of the address space of the Crypto-Tile. In addition, also the SE can obtain access to the address space of Crypto-Tile, by setting a dedicated and custom signal which has been added to the interface of 32-bit AXI4 bus, according to the specification of AXI4 standard. Furthermore, even if not highlighted in Figure 5, the Crypto-Tile integrates also a module dedicated to the management of the local clock

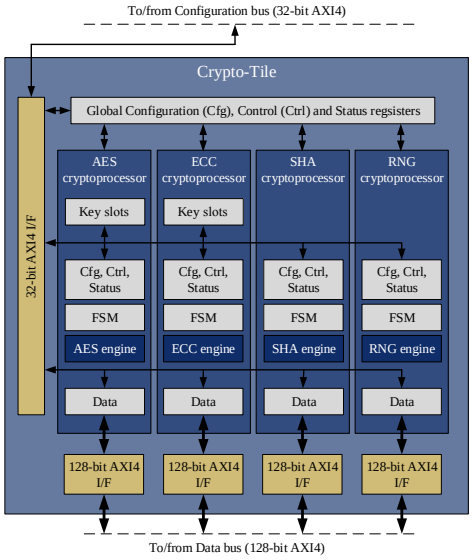

Fig. 5. Outline of Crypto-Tile IP architecture. The grey boxes represent registers and related logic resources; the dark blue boxes represent the cryptographic engines; the dark golden boxes represent the AXI4 interfaces logic. The narrower (black) lines with arrows indicate 32-bit buses, while the wider (black) lines indicate 128-bit buses.

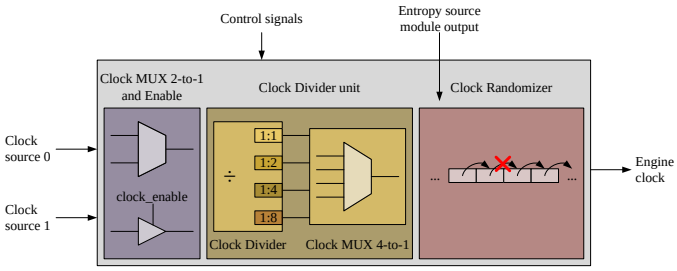

Fig. 6. High-level architecture of Crypto-Tile local clock manager.

and reset signals for the cryptographic engines, to spare power when the corresponding cryptoprocessor is disabled (using the global configuration), and also for the division and the randomization of clock signals. Such module is named clock subsystem, and, together with the other main blocks composes the Crypto-Tile.

The Global Management Unit is in charge of handling the global configuration and control of Crypto-Tile and reporting its global status. Each time a part of the global configuration is intended to be updated, it is necessary to provide a specific unseal value that enabled the modification of the settings: also the unseal value can be changed during the initial configuration routine. Rather, some parts of the global Crypto-Tile configuration can be modified only once, after the reset of the system, and they cannot be modified until the next reset event, as, for example, the settings related to the debug capabilities.

The Global Management Unit allows also to selectively and independently enable and configure each cryptoprocessor, by indicating which of the supported cryptographic algorithms can be used or not: after the reset, all the cryptoprocessors are disabled by default and until (or each time that) a cryptoprocessor is disabled the clock signal to the corresponding cryptographic engine is not provided, to reduce the power consumption. This clock gating mechanism is managed by the clock subsystem module which is described later. Concerning the global status of Crypto-Tile, the Global Management Unit integrates logic resources to log errors, especially the ones related to an abnormal state which is defined as a panic state and that can be triggered thanks to dedicated registers with two different levels of severity: the partial one or the full one. Once the panic state alert is enabled, any cryptographic operation is abruptly interrupted and all data are flushed, except for the key slots, whose content is erased based on the set global configuration and the level of panic.

The clocking subsystem of Crypto-Tile consists of four independent local clock managers, one for each cryptographic engine, and whose high-level architecture is reported in Figure 6.

As illustrated by Figure 6 the clock manager for the local clock signals to cryptographic engines is mainly composed of a cascade of three stages: a clock multiplexer (MUX), a clock divider and a clock randomizer. The first stage is responsible for selecting between two clock sources and applying the clock gating; the second stage locally derives the divided clock signals, dividing the output clock signal of the previous stage by 1, 2, 4 and 8; the third and last stage performs the randomization of the clock signal generated by the second stage, by randomly and temporary gating

it to skip none clock cycle or 1 clock cycle every 2, 4, 8, 16 or 32 clock cycles, in mean. The skip event is triggered using a PRNG module which generates a random number and an internal counter: when the counter matches the random number, the clock cycle is skipped. To enhance the unpredictability of PRNG integrated within the local clock manager module, its internal state is mixed with the output of the entropy source module composing the CSPRNG engine of RNG cryptoprocessor.

The control signals to the clock manager module are driven by both the global configuration settings of the Global Management Unit, concerning the main enable signal used by the clock muxing stage for gating the clock sources, and the settings of configuration local to the cryptoprocessor integrating the engine for which the clock signal is generated, affecting the clock division parameter and the clock randomization. The two clock sources consist of a global clock signal generated by a Phased-Lock Loop (PLL), which is regulated by the Power management unit of EPI chip (refer to Figure 1), and a local clock signal derived by a ring oscillator.

Concerning Figure 5, the AES cryptoprocessor consists in several logic resources that are built around and wrap the cryptographic engine dedicated to the acceleration in the hardware of AES cypher algorithms (i.e. the AES engine). Such resources count the registers for local configuration, control and status of AES cryptoprocessor, to manage the AES engine using an FSM. Detailed information can be found in [28].

Similarly to the AES cryptoprocessor, also the ECC cryptoprocessor integrates logic resources for local configuration, control and status, key slots, and data registers operating in the same way, and that surround the ECC engine which is responsible for the hardware acceleration of public-key algorithms, providing the security services. Detailed information can be found in [29].

The SHA cryptoprocessor features an architecture similar to the ones described respectively for the AES and ECC cryptoprocessors. The main difference lies in the absence of key slots, while the engine of this cryptoprocessor has been developed according to the same approach of finding the best trade-off between performance and complexity, accelerating in hardware all the SHA2 and all the SHA-3 functions. The cybersecurity services offered by this engine, the cryptoprocessor which integrates it, and its features and main characteristics are described in the publication [30].

The RNG cryptoprocessor has an architecture similar to the SHA one, i.e. integrating logic resources for configuration, control and status, data registers a cryptographic engine (the RNG engine), without including any hardware resources for key material. The RNG engine implements a CSPRNG, to provide random sequences of bits (or numbers) with an entropy level that can be considered sufficient for cryptographic applications requiring a high level of security (or security strength). In addition, the RNG cryptoprocessor embeds also a PRNG unit to offer a stream of (32-bit) random numbers continuously available on-demand, and that can be used in applications that require high speeds or low latencies and that can accept a low level of entropy. Detailed information can be found in [31].

3 Crypto-Tile Design Flow

3.1 Verification

As the first step of the verification phase, all the engines have been tested (independently) using the official test vectors released by the NIST through its Cryptographic Algorithm Validation Program (CAVP), whose purpose is to assure that a cryptographic algorithm implementation adheres to the specifications detailed in the associated cryptographic algorithm reference standards. For each Federal Information Processing Standards (FIPS)-approved and NIST-recommended cryptographic algorithm, it offers a suite of validation tests (called the algorithm's validation system) to test the algorithm specifications, components, features, and/or functionality of that algorithm. Such validation suites are the:

- Advanced Encryption Standard Algorithm Validation System (AESAVS), for ECB, CBC, CFB, OFB and CTR modes of AES cipher;
- CMAC Validation System (CMACVS), CCM Validation System (CCMVS), GCM Validation System (GCMVS) and XTS-AES Validation System (XTSVS), respectively for CMAC, CCM, GCM and XTS modes of AES cipher;
- ECDSA Validation System (ECDSA2VS) , version 2 of the validation system for ECDSA, related to FIPS 186-4;
- Secure Hash Algorithm Validation System (SHAVS) and Secure Hash Algorithm-3 Validation System (SHA3VS) , respectively for the SHA2 and SHA-3 functions;
- DRBG Validation System (DRBGVS), for the DRBG mechanisms described in Special Publication (SP) 800-90A;

Then the functional verification of Crypto-Tile followed by defining a Test Plan and developing a dedicated environment [32] in SystemVerilog language to run digital simulations using the Questa simulator by Mentor Graphics. The verification environment mainly consist is a battery of SystemVerilog test cases, each one including a common testbench that integrates both the device under test, i.e. the Crypto-Tile, and 5 different instances of a non-synthesizable Hardware Description Language (HDL) module aimed to emulate the Master entity for the AXI4 Slave interfaces of the Crypto-Tile.

3.2 Synthesis and Implementation

The synthesis process of Crypto-Tile was performed using Design Compiler by Synopsys and Vivado by Xilinx, targeting, respectively standard-cell and FPGA technologies.

3.2.1 FPGA technology results

In the case of FPGA, the full implementation flow was performed, i.e. logic synthesis, placement and routing steps. The target device for this flow was a Xilinx Virtex UltraScale+ FPGA VU37P (device XCVU37P-L2FSVH2892EES9837), that is manufactured using a 16 nm low power FinFET+ process technology from TSMC. The main programmable logic element of this device is named Configurable Logic Block (CLB), and it integrates several

Entity	Max. Frequency [MHz]	CLB (162960)	CLB LUTs (1303680)	CLB Registers (2607360)	DSPs (9024)
Crypto-Tile (top-level)	150	27507 (16.9%)	144892 (11.1%)	93503 (3.6%)	64 (0.7%)
AES engine	170	1253 (0.8%)	6036 (0.5%)	2460 (0.1%)	0 (0.0%)
ECC engine	95	15151 (9.3%)	79219 (6.1%)	37626 (1.4%)	64 (0.7%)
SHA engine	190	3433 (2.1%)	10290 (0.8%)	10787 (0.4%)	0 (0.0%)
RNG engine	260	2294 (1.4%)	10154 (0.8%)	7122 (0.3%)	0 (0.0%)

TABLE 3
Implementation results on FPGA VU37P.

logic resources, including LookUp Tables (LUTs) (referred to as CLB LUTs), flip-flops (referred to as CLB Registers) and other logic blocks; the FPGA counts 162960 CLBs, each one provided with 8 LUTs (for a total of 1303680 CLB LUTs) and 16 flip-flops (for a total of 2607360 CLB Registers, or bits). It is equipped also with other embedded hardware resources as Random Access Memory (RAM) blocks, Digital Signal Processors (DSPs), PLLs and others.

The implementation flow followed for the Crypto-Tile counted an incremental approach based on the sweep of frequency to determine the maximum supported frequency for each clock domain inside the design of this IP. Concerning its architecture (Figure 5), the design of Crypto-Tile can be split is 5 main clock domains: one of AXI4 resources (interfaces, registers, ...) and a clock domain for each distinct engine, for a total of 4 (i.e. AES, ECC, SHA and RNG engines). In addition, using Vivado, we set the synthesis and implementation strategies oriented to the optimization of the performance (of timing and power, notably the synthesis strategy denoted as *Flow_PerfOptimized_high* and the implementation strategy denoted as *Performance_ExtraTimingOpt*, respectively). We also applied the timing constraints in addition to the ones for the correct synthesis of the RNG engine. Firstly, each engine has been implemented separately, and once determined their maximum frequencies, the implementation of the whole Crypto-Tile design followed. In this last case, further timing constraints have been applied, such as the false paths between the different clock domains (to declare them asynchronous each other) and the input and output delays on the ports of Crypto-Tile top-level, both with a minimum delay and a maximum delay corresponding, respectively, to the 10% and the 20% of the clock period. Table 3 reports the final results of this activity.

Concerning Table 3, the percentage data between round brackets do not express the relative consumption of resources of a sub-module of the Crypto-Tile concerning the top-level, but they report the percentage utilization for the total amount of available resources of the VU37P FPGA. Moreover, Table 3 emerges that the ECC engine occupies about half of the whole design of Crypto-Tile: more precisely, the 55.1% if considering the CLBs, the 54.6% if considering the CLB LUTs, and the 40.2% if considering the CLB Registers. Anyway, it is not possible to make a fair comparison among the sub-blocks composing the Crypto-Tile design, because some of them consume also other dedicated hardware resources in addition to the logic programmable ones: for example, the ECC engine consumes also 64 DSPs, and they correspond also to the total of DSPs used by the whole Crypto-Tile. Or, even if not shown in Table 3, the Vivado report counted also the usage of 11 global clock buffers by the clock subsystem. This number can be explained by counting a total of 8 clock buffers for the engines (2 clock buffers for each of the four engines, for the clock and the reset signals, respectively) and 3 clock buffers for the AXI4 clock domain: 1 clock buffer for the 32-bit AXI4 clock domain, 1 for the 128-bit one, and 1 for the common reset signal of such clock domains.

3.2.2 Standard-cell technology results

Also for standard-cell technology, an approach similar to the illustrated for the FPGA was used, by splitting the design into different 5 clock domains and determining the maximum frequency supported by each of them. It is possible to have also a unique clock domain if this is required at the system level. Design Compiler by Synopsys was used, limiting to the only synthesis step, and target technology was the one proposed by the EPI project, i.e. the *H300 BASE SVT C8* of the 7 nm TSMC process *CLN07FF41001 SVT*, and released by ARM as part of the package of logic products named *Artisan 7nm TSMC CLN07FF41001*. The operating conditions and the technology corner case used in the synthesis were 0.90 V for the voltage supply, and 125° C for temperature and slow process. Moreover, it has been used the *Zero* model as a wire load model.

Referring to Table 4, similarities can be found concerning the case of FPGA implementation and its corresponding results (Table 3). In particular, the ECC engine consumes an amount of area/resources which is about the 50% of the total design area, while the SHA and RNG engines show approximately the same area consumption.

3.2.3 Gate-level simulations and power estimation

One of the outcomes of the synthesis activity on the 7 nm standard-cell technology was the gate-level netlist (in Verilog format). Such files are permitted to run off the gate-level timing simulations, as further verification of the synthesis process and the timing constraints used in it. To emulate a reasonable context close to the real one (i.e. to the final layout of Crypto-Tile), for the gate-level simulation step, they have been used the maximum frequencies for the engines (i.e. 2.425 GHz, 1.525 GHz, 3.725 GHz and 4.325 GHz, respectively for the AES, ECC, SHA and RNG engines), while for the AXI4 clock domain(s) it was used a frequency of 3.7 GHz.

To give an overall indication of the power consumption of each CO, implementing ad-hoc test cases, further power

Entity	Max. Frequency [GHz]	Area	
		Absolute [kGE]	Percent [%]
Crypto-Tile (top-level)	3.7	1325.16	100.0
AES engine	2.425	56.01	4.2
ECC engine	1.525	658.90	49.7
SHA engine	3.725	128.32	9.7
RNG engine	4.325	127.16	9.6

TABLE 4
Post-synthesis results for Crypto-Tile synthesis on 7 nm technology.

simulations have been conducted to isolate the dynamic power consumption of the engine performing the CO so that the total power consumption of a CO can be computed by summing the base power consumption of the Crypto-Tile ($P_{Base} = 68.3mW$) and the dynamic power consumption of the corresponding engine (P_{Dyn}): the results of this procedure are summarized in Figure 7.

Concerning data in Figure 7, the dynamic power (i.e. the sum of internal and switching powers) has been reported by grouping similar operations which showed similar consumptions, i.e. all the AES COs for all the key sizes (i.e. 128 and 256 bits), case AES (about $23mW$), the SHA2-224 and SHA2-256 functions, case SHA2-224/256 (about $94mW$), the SHA2-384 and SHA2-512 functions, case SHA2-384/512 (about $102mW$), all the SHA-3 functions, case SHA-3 (about $72mW$). Instead, for ECC and RNG COs, only the most significant cases have been documented, illustrating the dynamic power consumption of ECDSA generation (for both 256-bit and 521-bit curves/keys), case ECDSA generation (about $172mW$), ECDSA verification (for both 256-bit and 521-bit curves/keys), case ECDSA verification (about $131mW$), and for random number generation using an internal seed (from the entropy source module, about $132mW$) or an external seed (about $115mW$), case RNG (internal seed) and RNG (external seed), respectively. Such data can also be combined to compute the (total) power consumption of the Crypto-Tile when running multiple COs: for example, assuming an AES and a SHA2-256 operation are performed concurrently (one using the Secure MCU interface and the other one using the Secure DMAC interface), then the power consumption of Crypto-Tile can calculate as the sum of $68.3mW$ (base power consumption), $23mW$ (the AES engine dynamic power consumption) and $94mW$ (the SHA engine dynamic power consumption), for a total of about $185mW$; this assuming the unused engines (i.e. ECC and RNG) are disabled.

The power simulations presented in Section 3.2.3 have been used also to evaluate the resistance of Crypto-Tile and the engines embedded within into Side-Channel Attacks (SCAs), in particular to power analysis attacks, by emulating the power consumption of such engines and reconstructing the power traces (or waveforms) they show when executing a cryptographic operation. Detailed information is available in [28], [29], [30].

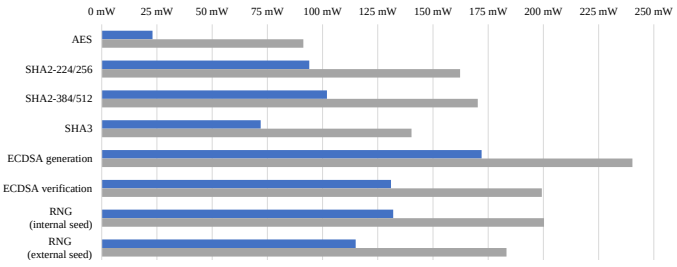

Fig. 7. Summary of post-synthesis COs power consumption on 7nm technology. The blue bar indicates the dynamic power consumption of the engine executing the CO reported (P_{Dyn}), while the grey bar indicates the total power consumption for that CO (P_{CO}): the base (total) power consumption of the Crypto-Tile is indicated with $P_{Base} = 68.3mW$, hence the total power consumption (of the Crypto-Tile) when executing a CO can be calculated as $P_{CO} = P_{Base} + P_{Dyn}$.

4 UNIPI CRYPTO-TILE DEMOBOARD

As a final activity, a demoboard was implemented to validate the integration of Crypto-Tile into a software system, i.e. by coupling it to a processor, emulating the element of the Security Subsystem of EPI GPP. This permitted not only the development of the software drivers for the Crypto-Tile but also to measure and validate other characteristics related to its implementation.

4.1 Demoboard and speed tests

The demoboard of Crypto-Tile has been implemented using a VCU128 board by Xilinx, that features the Virtex Ultrascale+ technology VU37P FPGA illustrated in Section 3.2.1, and it is equipped also with other hardware resources such as a DDR4 memory with a capacity of 4.5 GB. The CVA6 [33] has been chosen as the processor, also named Ariane CPU. It is an open-source, six-stage, single-issue, in-order CPU based on the 64-bit RISC-V instruction set and implemented in SystemVerilog HDL, developed by the OpenHW Group [34]. Figure 8 shows the system architecture of the demoboard.

Referring to Figure 8, it can be noted the similarity between the implemented system and the Security Domain of EPI GPP, highlighting the processing unit (block *CVA6_system*), corresponding to the Secure MCU, and accordingly, the Crypto-Tile module, represented by the block *cryptotile_cdma_system*), which internally includes also two Secure DMAC modules. The SRAM, is implemented using the DDR4 memory onboard the VCU128, together with its controller unit (i.e. the block *ddr4_0*). Finally, the AXI4 buses are realized using the remaining blocks (i.e. *smartconnect_memory_peripherals* and *smartconnect_0*). In addition to the I/O link to the DDR4 memory (i.e. port *ddr4_sdram*), they are present also in other I/O lines: the port *default_100mhz_clock*, for the provisioning of the main system clock source at 100 MHz, and the ports *rs232_uart_0* and *rs232_uart_1* which, exploiting a UART module onboard the VCU128 device, are employed to establish serial communication with a host system and hence to allow the interaction with the CVA6 processor using a terminal. The system in

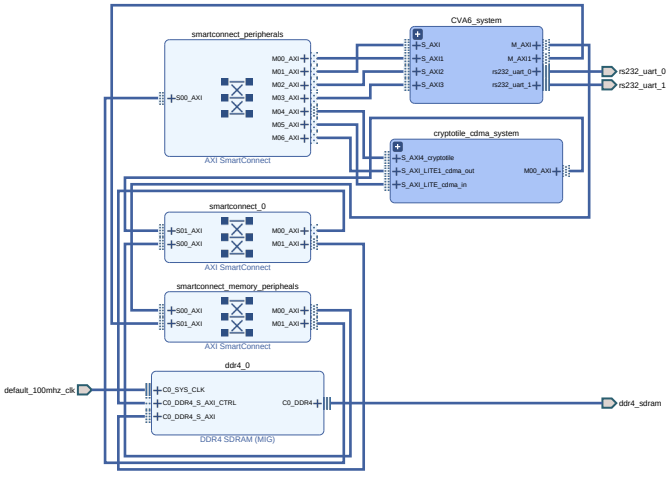

Fig. 8. System architecture of Crypto-Tile demoboard on the Xilinx VCU128 Board.

Name	CLB	CLB LUTs	CLB Registers	BlockRAM tile	DSPs	Global Clk Buffer	PLL	MMCM
demo_cva6_cryptotile	50188	249914	203302	71.5	94	24	3	2
vio_reset	33	97	234	0	0	0	0	0
system_ila	574	1931	3071	5	0	0	0	0
smart connect	7101	28683	50622	0	0	0	0	0
proc_sys_reset_ddr4	7	15	33	0	0	0	0	0
ddr4	4277	18833	20774	25.5	3	5	3	1
cryptotile_cdma	29553	156031	101890	0	64	18	0	1
CVA6	9668	44333	26678	41	27	0	0	0

TABLE 5
Post-implementation hierarchical FPGA resources usage for demoboard. The reported hardware resources are referred to as the VU37P FPGA onboard the VCU128

Figure 8 has been implemented setting a frequency of 100 MHz for the main system clock (i.e. for the Ariane CPU, the DMAC, the AXI4 systems buses, the peripherals interfaces, ...) and the maximum supported frequencies illustrated in Section 3.2.1 for the Crypto-Tile engines; the other modules of the Crypto-Tile wrapping the engines (i.e. the crypto-processors wrapping logic for control, the registers, the Global Management Unit, the AXI4 interfaces, et al.) are clocked with the system clock of 100 MHz. Table 5 shows the resource utilization of VU37P FPGA for the VCU128 demoboard system. The demoboard has been primarily used as a development environment to design and test the software drivers required to integrate the Crypto-Tile into a software system and to use it as a peripheral of the processor. The drivers have been written in C language and after the debug and verification phases, they have been used to build software routines (benchmarks) to measure the execution time of cryptographic algorithms, comparing the results obtained with the software-only solution with the one obtained by accelerating the same algorithm in hardware with the Crypto-Tile. Such comparison shows the advantages in terms of performance that can be gained by using the proposed mixed software/hardware solution (i.e. the RISC-V processor coupled to the Crypto-Tile accelerators suite) concerning a pure software solution (i.e. RISC-V processor and cryptographic software libraries, such as the wolfCrypt one [35], [36]) for the implementation of the Security Domain element of the Security Subsystem, as discussed in Section 2.2. Therefore the results reported in Tables 6, 7 and 8 give a measure of how much time the Security Domain element requires to execute a cryptographic function. The cases analyzed are:

- software-only solution (labelled as *SW-only*, and consuming a time T_S);
- mixed software/hardware solution (labelled as *HW*, and consuming a time T_H);
- mixed software/hardware solution exploiting Direct Memory Access (DMA) for data movement (labelled as *HW DMA*, and consuming a time T_{HD}).

The additional overhead in terms of time required by the Security Subsystem to serve a request for a security service from the non-secure zone of the EPI chip is not considered in this comparison. Anyway, it will be the same for both the implementation solutions for the Security Domain because it will be a direct consequence of how the Security Subsystem (and the Security Domains composing it) is integrated within the EPI GPP. Such investigation will be the subject of future works; however, we did not expect significant bottlenecks in the communication between the non-secure zone and the Security Subsystem of the EPI chip thanks to the connection links and in particular thanks to the connection to the high-bandwidth NoC for high-speed data exchange after the configuration of a cryptographic operation.

AES-128		Payload		Execution time [us]			Ratio (T_S/T_H)	Ratio (T_S/T_{HD})
		AAD	P/C	SW-only (T_S)	HW (T_H)	HW DMA (T_{HD})		
ECB	Enc	-	160 B	283.64	33.24	4.14	8.53	68.51
	Dec			256.56	36.48	4.56	7.03	56.26
CBC	Enc	-	160 B	175.1	31.9	4.05	5.49	43.23
	Dec			161.3	35.92	4.91	4.49	32.85
CFB	Enc	-	160 B	174.06	32.55	4.18	5.35	41.64
	Dec			164.48	36.9	4.29	4.46	38.34
OFB	Enc	-	160 B	156.49	32.49	4.18	4.82	37.44
	Dec			152.2	36.21	4.28	4.20	35.56
CTR	Enc	-	160 B	73.32	13.77	4.14	5.32	17.71
	Dec			63.74	15.42	4.28	4.13	14.89
CCM	Enc	32 B	24 B	172.41	15.48	6.27	11.14	27.50
	Dec			176.7	13.72	6.05	12.88	29.21
GCM	Enc	90 B	51 B	417.82	24.91	9.51	16.77	43.93
	Dec			383.4	26.29	9.52	14.58	40.27
XTS	Enc	-	32 B	72.63	8.7	2.17	8.35	33.47
	Dec			114.72	8.86	2.16	12.95	53.11

AES-256		Payload		Execution time [us]			Ratio (T_S/T_H)	Ratio (T_S/T_{HD})
		AAD	P/C	SW-only (T_S)	HW (T_H)	HW DMA (T_{HD})		
ECB	Enc	-	160 B	353.73	33.89	4.13	10.44	85.65
	Dec			322.33	36.73	4.48	8.78	71.95
CBC	Enc	-	160 B	272.44	32.63	4.15	8.35	65.65
	Dec			236.66	36.79	4.87	6.43	48.60
CFB	Enc	-	160 B	239.27	32.33	4.24	7.40	56.43
	Dec			215.51	36.8	4.25	5.86	50.71
OFB	Enc	-	160 B	207.17	32.57	4.31	6.36	48.07
	Dec			199.49	36.21	4.29	5.51	46.50
CTR	Enc	-	160 B	92.88	13.62	4.17	6.82	22.27
	Dec			81.49	15.95	4.31	5.11	18.91
CCM	Enc	32 B	24 B	218.83	16.05	6.27	13.63	34.90
	Dec			230.5	13.09	6.05	17.61	38.10
GCM	Enc	90 B	51 B	445.08	26.25	9.51	16.96	46.80
	Dec			417.4	26.78	9.52	15.59	43.84
XTS	Enc	-	32 B	90.15	8.93	2.17	10.10	41.54
	Dec			180.61	9.46	2.16	19.09	83.62

TABLE 6
SW-only vs. HW-accelerated: AES-128/256 algorithms. The symbols *P/C* and Additional Authenticated Data (AAD) in the *Payload* column indicate, respectively, the byte length of plaintext or ciphertext and one of the additional authenticated data (used only by the CCM and GCM modes.

Results of the comparison for the AES algorithms are reported in Table 6. For AES-128 and AES-256 algorithms the execution time T_H of the hardware-accelerated version is of the order of tens of μs, while the software-only counterpart takes a time T_S of the order of hundreds of μs for the same operation, with a variable ratio T_S/T_H in the range 4 to 19 (indicating the hardware-accelerated version is 4 to 19 times faster than the software-only implementation). In addition, the *HW DMA* solution takes a time T_{HD} of the order of μs, with a variable ratio T_S/T_{HD} in the range 14 to 85. The payload size used for these tests is indicated in the *Payload* column, using symbols *P/C* for the byte length of plaintext or ciphertext and AAD for the additional authenticated data, that are used only in case of CCM and GCM modes. Table 7 shows the results of speed tests for the SHA algorithms. Table 7 reports that for a payload of about 8 kB, the execution time T_S of the software-only version of the SHA algorithms takes about $0.7\ ms$ up to $1.5\ ms$ to process data, while the hardware-accelerated counterpart employs a time of about $0.07\ ms$, being from 9 up to about 21 times faster than the first solution. The *HW DMA* solution takes about $0.05\ ms$, being from 13 up to about 29 times faster than the SW-only solution.

Concerning the ECC functions, only the most significant operations were analyzed, i.e. the ECDSA one, whose results are summarized in Table 8. As shown in Table 8, in the case of ECDSA functions the hardware-accelerated version is about 200 to 400 times faster than the software-only counterpart that can take up to almost $1s$ to perform a signature verification over the 521-bit elliptic curve.

Finally, Table 9 reports, respectively, the throughput of AES and SHA functions. In this case, also the time required to configure the CO in the Crypto-Tile has been included, as this phase is always required to execute a cryptographic operation with such hardware accelerator, whereas the key installation time can be excluded as the key slots can be set at boot time and then kept unchanged while running multiple COs with the same key value.

In the case of RNG engine, instead of implementing speed tests, the software drivers have been employed to evaluate the most significant feature for the category of operations it can offer, i.e. the randomness of the generated bitstreams. The results are reported in [37]

SHA	Payload	Execution time [ms]			Ratio (T_S/T_H)	Ratio (T_S/T_{HD})
		SW-only (T_S)	HW (T_H)	HW DMA (T_{HD})		
SHA2-224	8.4 kb	0.92	0.07	0.05	13.14	18.40
SHA2-256	8.4 kb	0.86	0.07	0.05	12.29	17.20
SHA2-384	8.9 kb	0.78	0.07	0.05	11.14	15.60
SHA2-512	8.9 kb	0.67	0.07	0.05	9.57	13.40
SHA-3-224	8.1 kb	0.85	0.07	0.05	12.14	17.00
SHA-3-256	8.7 kb	0.88	0.07	0.05	12.57	17.60
SHA-3-384	8.3 kb	1.05	0.07	0.05	15.00	21.00
SHA-3-512	8.7 kb	1.49	0.07	0.05	21.29	29.80

TABLE 7
SW-only vs. HW-accelerated: SHA algorithms.

ECDSA		Execution time [ms]			Ratio (T_S/T_H)	Ratio (T_S/T_{HD})
		SW-only (T_S)	HW (T_H)	HW DMA (T_{HD})		
NIST P256, SHA2-224	Gen	166.39	0.4	0.43	415.98	386.95
	Ver	221.7	0.67	0.7	330.90	316.71
NIST P256, SHA2-256	Gen	166.02	0.4	0.43	415.05	386.09
	Ver	220.05	0.67	0.7	328.43	314.36
NIST P256, SHA2-384	Gen	168.21	0.4	0.43	420.53	391.19
	Ver	220.05	0.67	0.7	328.43	314.36
NIST P256, SHA2-512	Gen	165.72	0.4	0.43	414.30	385.40
	Ver	219.31	0.67	0.7	327.33	313.30
NIST P521, SHA2-224	Gen	694.22	2.72	2.75	255.23	252.44
	Ver	922.25	4.54	4.57	203.14	201.81
NIST P521, SHA2-256	Gen	697.04	2.72	2.75	256.26	253.47
	Ver	924.63	4.54	4.57	203.66	202.33
NIST P521, SHA2-384	Gen	694.58	2.72	2.75	255.36	252.57
	Ver	923.6	4.54	4.57	203.44	202.10
NIST P521, SHA2-512	Gen	697.47	2.72	2.75	256.42	253.63
	Ver	924.84	4.54	4.57	203.71	202.37

TABLE 8
SW-only vs. HW-accelerated: ECDSA functions. The keywords *NIST P256* and *NIST P521* indicate the execution of the ECDSA operation over the 256-bit or the 521-bit curve, whereas the associated SHA2 algorithm indicates the function used for the computation of the digest of the message to which the digital signature is applied.

4.2 Comparison With Existing Solutions

While each of the single Crypto-Tile engines has been compared separately with other available solutions in the state of the art in [28], [29], [30], [31], demonstrating to outperform them, it is also necessary to compare the features and performances at the system level of the entire Crypto-Tile. The solutions presented in Section 2.1 are already available on the market; however, they have differences and drawbacks from our proposed system: all the presented solutions are provided by a US-based company, which may be a limitation for strategic circuits such as Cryptographic processors. Moreover, all the presented systems are developed to be integrated with their proprietary processor architectures, with the exclusion of [38]. All of them however share the lack of detailed information on throughput and hardware complexity, which makes it impossible to fairly compare them with our proposed solutions. Other attempts in the literature to develop systems similar to our Crypto-tile are: [39], where a System-on-Chip (SoC) implementation of a TEE is carried out and [40], where a AES-SHA Optimised Crypto Processor is presented. Both these solutions however include far fewer features: they miss the ECC engines and the available engines do not have all the operating modes that we included. Our Crypto-tile moreover includes innovative hardware support for key management, clock randomisation and access privilege mechanism.

AES-128	Throughput [Mbps]			
	HW		HW DMA	
	Enc	Dec	Enc	Dec
ECB	44.41	44.74	357.54	358.54
CBC	44.24	44.28	357.42	357.99
CFB	44.27	45.37	358.14	358.04
OFB	42.94	45.37	358.14	358.03
CTR	44.26	45.21	358.05	358.07
XTS	44.89	45.22	358.05	358.06

SHA	Throughput [Mbps]	
	HW	HW DMA
SHA2-224	15.69	21.97
SHA2-256	16.82	23.55
SHA2-384	18.62	26.07
SHA2-512	15.69	21.97
SHA-3-224	19.63	27.48
SHA-3-256	19.81	27.73
SHA-3-384	18.96	26.54
SHA-3-512	19.63	27.48

TABLE 9
Throughput of HW-accelerated AES and SHA algorithms

5 Conclusion

Based on how much expressed so far, the implemented cybersecurity module (i.e. the Crypto-Tile) meets all the highly-qualified security requirements emanated by the most significant institutions in the matter of cybersecurity, guaranteeing long-term protection also for PQC criteria. Only the public-key cryptography functions are currently able to offer appropriate security strength levels with respect only to classical cryptography criteria, as the standardization process of a public-key PQC resistant algorithm is ongoing. Anyway, thanks to the modular architecture of Crypto-Tile IP, once the draft version of the new standard will be available, a public-key engine and cryptoprocessor with PQC features will be easily integrated, by substituting actual ECC cryptoprocessor or by instancing it in parallel to other already present cryptoprocessors, replicating and adapting the same logic resources and hardware mechanisms for the interface security policies and without altering the functionalities already implemented and verified. As expressed in [28], [29], [30], [31], all the hardware acceleration engines are competitive with the state of the art of the corresponding implementations. On top of that, with our contribution, we present a Crypto-Tile with unprecedented hardware-supported capabilities, both in terms of system-level features (e.g. key management) and supported algorithms and operating modes.

Finally, the activities carried out exploiting the FPGA demoboard on the VCU128 device confirmed the need for a hardware acceleration module for high-performance applications, showing numerically that the usage of Crypto-Tile as a coprocessing peripheral of the (secure) processor allows achieving lower execution time of routines concerning cryptographic algorithms, especially in case of the public-key ones with an overall speed-up in the order of hundreds. In addition, also the support to the HMAC scheme has been validated, and a similar analysis in terms of speed of execution could be conducted also using a Linux kernel, rather than bare-metal applications because the boot of Linux image has already been tested and the corresponding kernel drivers for the integration of Crypto-Tile already developed.

Moreover, the software code implemented with the demoboard allowed also us to evaluate the main features of the random bitstreams generated by the DRBG unit and the entropy source module of RNG engine. On one hand, the correct implementation of the RNG engine as a CSPRNG was validated, because the results confirmed that, once seeds with appropriate entropy levels are used, it can generate sequences whose randomness cannot be distinguished from the one of an ideal random generator, with the confidence of 99%. Being such characteristic depending only on the deterministic part of RNG engine, it will be maintained also for the chip on the 7 nm standard-cell technology. On the other hand, the measured level of entropy contributed to proving the portability of the digital entropy source module design, in addition to how much illustrated in the work presented in [31], showing also characteristics in terms of entropy rate that outperform the other main solutions in the field of digital TRNGs. Once the EPI chip will be available, during the Specific Grant Agreement (SGA)2 phase which started in the last month of 2021 and will last to the end of 2023, the same analysis will be conducted to evaluate the entropy generated on the 7 nm standard-cell technology. Moreover, we plan also to extend the support to more Quantum-resistant algorithms.

Acknowledgments

This research was partially funded by the European Union's Horizon 2020 research and innovation programme "European Processor Initiative" under grant agreement No. 826647 (EPI SGA1) and No. 101036168 (EPI SGA2), by the Italian Ministry of University and Research (MUR) through the project CN4 - CN00000023 of the Recovery and Resilience Plan (PNRR) program, grant agreement no. I53C22000720001 and in the framework of the FoReLab project (Departments of Excellence).

References

[1] European Processor Initiative, "European Processor Initiative," https://www.european-processor-initiative.eu/, Accessed on 15 February 2022.

[2] M. Nabeel, M. Ashraf, S. Patnaik, V. Soteriou, O. Sinanoglu, and J. Knechtel, "2.5d root of trust: Secure system-level integration of untrusted chiplets," *IEEE Transactions on Computers*, vol. 69, no. 11, pp. 1611–1625, 2020.

[3] P. Nannipieri, S. Di Matteo, L. Zulberti, F. Albicocchi, S. Saponara, and L. Fanucci, "A risc-v post quantum cryptography instruction set extension for number theoretic transform to speed-up crystals algorithms," *IEEE Access*, 2021.

[4] M. D. Hill and V. J. Reddi, "Accelerator-level parallelism," *Communications of the ACM*, vol. 64, no. 12, pp. 36–38, 2021.

[5] A. Ehret, E. Del Rosario, K. Gettings, and M. A. Kinsy, "A Hardware Root-of-Trust Design for Low-Power SoC Edge Devices," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, September 2020, pp. 1–6.

[6] R. Perez, "Silicon Systems Security and Building a Root of Trust," in *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, November 2015, pp. 1–4.

[7] Z. Yu, H. Dai, X. Xi, and M. Qiu, "A Trust Verification Architecture with Hardware Root for Secure Clouds," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 3, pp. 353–364, July 2020.

[8] L. Baldanzi, L. Crocetti, S. Di Matteo, L. Fanucci, S. Saponara, and P. Hameau, "Crypto accelerators for power-efficient and real-time on-chip implementation of secure algorithms," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, November 2019, pp. 775–778.

[9] OpenSSL, "OpenSSL," https://www.openssl.org/, Accessed on 15 February 2022.

[10] Rambus, "Introducing the Rambus CryptoManager Root of Trust (CMRT)," https://www.rambus.com/blogs/introducing-the-rambus-cryptomanager-root-of-trust-cmrt, Accessed on 15 February 2022.

[11] ARM, "ARM security technology – Building a secure system using TrustZone technology," ARM, Tech. Rep., 2005-2009, https://developer.arm.com/documentation/PRD29-GENC-009492/c, Accessed on 15 February 2022.

[12] ——, "TrustZone for Cortex-A," https://developer.arm.com/ip-products/security-ip/trustzone/trustzone-for-cortex-a, Accessed on 15 February 2022.

[13] ——, "TrustZone for Cortex-M," https://developer.arm.com/ip-products/security-ip/trustzone/trustzone-for-cortex-m, Accessed on 15 February 2022.

[14] Intel, "Intel Software Guard Extensions (Intel SGX) – Developer Guide," Intel, Tech. Rep., May 2018, https://www.intel.com/content/dam/develop/public/us/en/documents/intel-sgx-developer-guide.pdf, Accessed on 15 February 2022.

[15] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, "Using innovative instructions to create trustworthy software solutions," *2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, vol. 11, pp. 2 487 726–2 488 370, June 2013.

[16] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative Instructions and Software Model for Isolated Execution," *2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, vol. 10, June 2013.

[17] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for cpu based attestation and sealing," *2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, vol. 13, June 2013.

[18] Intel, "Product Brief - 3rd Gen Intel Xeon Scalable processors," https://www.intel.in/content/dam/www/central-libraries/us/en/documents/3rd-gen-intel-xeon-scalable-processors-product-brief.pdf, Accessed on 15 February 2022.

[19] ——, "Intel Integrated Performance Primitives Cryptography Acceleration on 3rd Generation Intel Xeon Processor Scalable and 10th Gen Intel Core Processors," https://www.intel.com/content/www/us/en/developer/articles/technical/intel-ipp-crypto-multi-buffer-acceleration.html, Accessed on 15 February 2022, November 2020.

[20] AMD, "BIOS and Kernel Developer's Guide (BKDG) for AMD Family 16h Models 30h-3Fh Processors," AMD, Tech. Rep., 2013-2016, https://www.amd.com/system/files/TechDocs/52740_16h_Models_30h-3Fh_BKDG.pdf, Accessed on 15 February 2022.

[21] L. Piccolboni, G. Di Guglielmo, S. Sethumadhavan, and L. P. Carloni, "Hardroid: Transparent integration of crypto accelerators in android," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, 2021, pp. 1–8.

[22] P. Choi, W. Kong, J.-H. Kim, M.-K. Lee, and D. K. Kim, "Architectural supports for block ciphers in a risc cpu core by instruction overloading," *IEEE Transactions on Computers*, pp. 1–1, 2021.

[23] T. Lu, "A survey on risc-v security: Hardware and architecture," 2021.

[24] M. Deutschmann, L. Iriskic, S.-L. Lattacher, M. Münzer, and O. Tomshchuk, "A puf based hardware authentication scheme for embedded devices," Technical report, Technikon Forschungs-und Planungsgesellschaft mbH . . . , Tech. Rep., 2018.

[25] NXP, "Building a Secure System using NXP Secure MCU LPC54S0xx," NXP, Tech. Rep., March 2019, https://www.nxp.com/docs/en/application-note/AN12385.pdf, Accessed on 15 February 2022.

[26] A. S. Siddiqui, Y. Gui, and F. Saqib, "Secure Boot for Reconfigurable Architectures," *Cryptography*, vol. 4, no. 4, September 2020.

[27] A. Kumar, R. S. Mishra, and K. R. Kashwan, "PUF based challenge response pair for secured authentication," *International Journal of Circuit Theory and Applications (IJCTA)*, vol. 9, no. 115-121, pp. 1–36, February 2017.

[28] P. Nannipieri, S. D. Matteo, L. Baldanzi, L. Crocetti, L. Zulberti, S. Saponara, and L. Fanucci, "Vlsi design of advanced-features aes cryptoprocessor in the framework of the european processor initiative," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, p. 177 – 186, 2022.

[29] S. Di Matteo, L. Baldanzi, L. Crocetti, P. Nannipieri, L. Fanucci, and S. Saponara, "Secure elliptic curve crypto-processor for real-time IoT applications," *Energies*, vol. 14, no. 15, August 2021.

[30] P. Nannipieri, M. Bertolucci, L. Baldanzi, L. Crocetti, S. Di Matteo, F. Falaschi, L. Fanucci, and S. Saponara, "Sha2 and sha-3 accelerator design in a 7 nm technology within the european processor initiative," *Microprocessors and Microsystems*, vol. 87, 2021.

[31] P. Nannipieri, S. Di Matteo, L. Baldanzi, L. Crocetti, J. Belli, L. Fanucci, and S. Saponara, "True Random Number Generator Based on Fibonacci-Galois Ring Oscillators for FPGA," *Applied Sciences*, vol. 11, no. 8, April 2021.

[32] L. Zulberti, S. Di Matteo, P. Nannipieri, S. Saponara, and L. Fanucci, "A script-based cycle-true verification framework to speed-up hardware and software co-design: Performance evaluation on ecc accelerator use-case," *Electronics*, vol. 11, no. 22, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/22/3704

[33] O. Group, "CVA6: A Linux-Capable RISC-V CPU," https://www.hackster.io/news/cva6-a-linux-capable-risc-v-cpu-299a40a5f871, Accessed on 15 February 2022.

[34] ——, "Ariane 4.2," https://github.com/openhwgroup/cva6, Accessed on 15 February 2022.

[35] wolfSSL Inc., "wolfCrypt," https://www.wolfssl.com/products/wolfcrypt-2/, Accessed on 15 February 2022.

[36] wolfSSL Inc., *wolfSSL User Manual, version 4.1.0*, wolfSSL Inc., August 2019, https://www.wolfssl.com/documentation/wolfSSL-Manual.pdf, Accessed on 15 February 2022.

[37] L. Crocetti, S. Di Matteo, P. Nannipieri, L. Fanucci, and S. Saponara, "Design and test of an integrated random number generator with all-digital entropy source," *Entropy*, vol. 24, no. 2, 2022.

[38] Rambus, "Hardware Root of Trust: Everything you need to know," https://www.rambus.com/blogs/hardware-root-of-trust/, Accessed on 15 February 2022.

[39] T.-T. Hoang, C. Duran, R. Serrano, M. Sarmiento, K.-D. Nguyen, A. Tsukamoto, K. Suzaki, and C.-K. Pham, "System-on-chip implementation of trusted execution environment with heterogeneous architecture," in *2021 IEEE Hot Chips 33 Symposium (HCS)*, 2021, pp. 1–16.

[40] D.-e.-S. Kundi, A. Khalid, A. Aziz, C. Wang, M. O'Neill, and W. Liu, "Resource-shared crypto-coprocessor of aes enc/dec with sha-3," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4869–4882, 2020.

Pietro Nannipieri Pietro Nannipieri got his PhD in Information Engineering in 2020 cum laude from the University of Pisa, where he is currently an Assistant Professor. His interests are digital and VLSI design, electronics for space applications, and cryptography. Pietro in 2019 was Visiting Researcher in the TEC-EDP Section in ESTEC (ESA), where he carried out different qualification tests on the SpaceFibre technology. His research focuses on hardware IPs for satellite onboard data handling, signal processing, and hardware cryptography. Indeed, he is currently working on the European Processor Initiative (EPI) project.

Luca Crocetti received a PhD degree (cum laude) in Information Engineering from the University of Pisa in 2022. He is currently an Assistant Professor in the Department of Information Engineering at the University of Pisa, and his research activities include design of VLSI and embedded systems for cybersecurity and cryptography in automotive, IoT, HPC, and space applications. He co-authored several scientific publications and holds 1 patent.

Stefano Di Matteo Stefano Di Matteo got his PhD in Information Engineering in 2023 cum laude from the University of Pisa, where he is currently a post-doc researcher. His research activities include digital and VLSI design of cryptographic primitives with a particular focus on asymmetric cryptography and post-quantum cryptography. He is currently involved in the European Processor Initiative (EPI) project.

Luca Fanucci Luca Fanucci got a PhD degree in Electronic Engineering from the University of Pisa in 1996. From 1992 to 1996, he was a research fellow with the European Space Agency. From 1996 to 2004 he was a senior researcher at the Italian National Research Council in Pisa. He is Professor of Microelectronics at the University of Pisa. His research interests include design technologies for integrated circuits and electronic systems. He is a co-author of more than 400 journal and conference papers and co-inventor of more than 40 patents. He served in several technical programme committees of international conferences.

Sergio Saponara Sergio Saponara received a PhD degree in electronic engineering from the University of Pisa, Italy. He was a Marie Curie Fellow with imec, Belgium. He is currently a Full Professor of electronics at the University of Pisa. He co-authored about 300 scientific articles and holds 18 patents. He is Founding Member of the IoT CASS SiG. He has been a TPC Member of over 100 international IEEE and SPIE conferences. He is an Associate Editor of several IEEE and IET journals. Since 2017, he has been an IEEE IMS Distinguished Lecturer.