Security Layers and Related Services within the Horizon Europe NEUROPULS Project

Security layers and related services within the Horizon Europe NEUROPULS project

Fabio Pavanello[1], Cedric Marchand[2], Paul Jimenez[2], Xavier Letartre[2], Ricardo Chaves[3], Niccolò Marastoni[4], Alberto Lovato[4], Mariano Ceccato[4], George Papadimitriou[5], Vasileios Karakostas[5], Dimitris Gizopoulos[5], Roberta Bardini[6], Tzamn Melendez Carmona[6], Stefano Di Carlo[6], Alessandro Savino[6], Laurence Lerch[7], Ulrich Ruhrmair[7], Sergio Vinagrero Gutiérrez[8], Giorgio Di Natale[8], Elena Ioana Vatajelu[8]

[1]Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, Grenoble INP, IMEP-LAHC, Grenoble, France
[2]Univ. Lyon, Ecole Centrale de Lyon, INSA Lyon, Université Claude Bernard Lyon 1, CPE Lyon, CNRS, INL, Ecully, France
[3]INESC-ID,IST, ULisboa, Lisbon, Portugal
[4]Department of Computer Science, University of Verona, Verona, Italy
[5]Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece
[6]Politecnico di Torino, Control and Computer Eng. Department, Italy
[7]Technical University of Berlin, Berlin, Germany
[8]Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France

***Abstract*—In the contemporary security landscape, the incorporation of photonics has emerged as a transformative force, unlocking a spectrum of possibilities to enhance the resilience and effectiveness of security primitives. This integration represents more than a mere technological augmentation; it signifies a paradigm shift towards innovative approaches capable of delivering security primitives with key properties for low-power systems. This not only augments the robustness of security frameworks, but also paves the way for novel strategies that adapt to the evolving challenges of the digital age.**

This paper discusses the security layers and related services that will be developed, modeled, and evaluated within the Horizon Europe NEUROPULS project. These layers will exploit novel implementations for security primitives based on physical unclonable functions (PUFs) using integrated photonics technology. Their objective is to provide a series of services to support the secure operation of a neuromorphic photonic accelerator for edge computing applications.

***Index Terms*—PUFs, low power, security, photonics.**

I. Introduction

The growing intelligence of the current environments is closely related to the deployment and use of neural networks (NN). However, many edge computing devices, such as those related to the Internet of Things (IoT), present serious challenges in terms of computing and security [1], [2]. On the one hand, edge computing devices cannot rely on the same level of resources as more sophisticated high-end computing solutions, e.g., located in data centers, for costs and weight reasons. On the other hand, edge computing devices present multiple access points, for example, through other computing nodes in the same network, which can be exploited to carry out various types of attacks [3]. Therefore, it is crucial to develop novel security layers compatible with edge computing device constraints in terms of lightweight character, low cost, low power, and robustness against attacks.

To address these requirements, hardware security primitives can be regarded as one of the best candidates to relax some constraints of classical systems, for example, where a secret key is required to be stored directly in a non-volatile memory [4]. Such security primitives can be used efficiently against attacks that aim to access specific memory sectors, as in the case of Spectre and Meltdown hardware vulnerabilities [5], [6].

A well-known class of hardware primitives that offers a solution to such attacks is one of physically unclonable functions (PUFs). Originally pioneered at MIT in 2002 by two independent groups in the optical and electronic domains, these primitives can be used in conjunction with well-known security protocols to establish low-power and robust security layers [7], [8]. Although various technologies have been investigated to enable such primitives, CMOS-based PUFs have become the most widely used solution to enable a series of security services such as cryptographic key generation, secure authentication, or root-of-trust features [9]. However, electronic solutions such as SRAM or Arbiter PUF have limitations in terms of reliability with respect to aging and fluctuations, as well as a limited number of degrees of freedom and the related achievable complexity, resulting in solutions prone to ML attacks [10].

In the Horizon Europe NEUROPULS project (which started in January 2023) [11], our objective was to develop security layers based on novel security primitives by leveraging the technology available in our platform. Specifically, our goal is to develop security layers based on a photonic integrated circuit (PIC). Such technology will be used to develop a neuromorphic photonic accelerator which we aim to protect using the very same photonic technology as the accelerator. A high-level

This work has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No. 101070238. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. It was also supported by the ANR within the PHASEPUF Project ANR-20-CE39-0004.

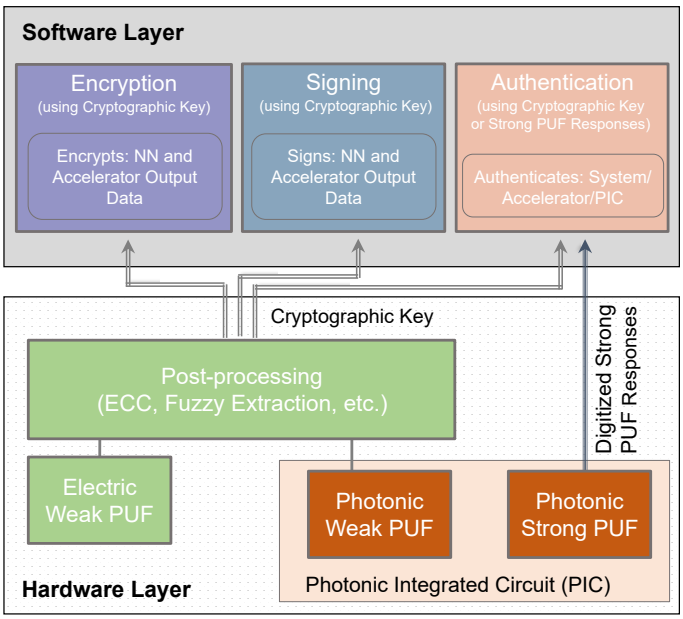

Fig. 1. Hardware-Software communication flow for security services in NEUROPULS. Weak and strong PUFs target different security services and supporting circuitry.

description of the security workflow in NEUROPULS is represented in Fig. 1 where primitives based on PUFs, implemented in a PIC alongside the accelerator and in an ASIC (based on SRAM) to guarantee unique binding between the chips, provide their output to the software layer. The software layer will implement specific protocols, discussed in the following, which will provide a series of services for the secure operation of the accelerator.

In the remainder of this document, we will first discuss security primitives based on photonic technologies and their operation in Section II. Then, in Section III, we will introduce the security services provided to the accelerator based on such primitives. Finally, we will qualitatively analyze the strength against attacks of the proposed security layers in Section IV and their modeling requirements in Section V to precisely predict their impact on the overall performance of the system.

II. Security Primitives and Related Metrics

A. PUFs

Although the majority of PUF implementations are based on CMOS technology because of its native compatibility with CMOS interfaces, they present a series of limitations, as mentioned above, which stem from their digital nature, but also technology. Photonics can allow the building of PUFs that present a much larger degree of freedom (e.g., phase, polarization, amplitude) and where the information is manipulated completely differently from a classical CMOS-based PUF. Furthermore, signals propagate in the analog domain and, therefore, can carry a much higher entropy than digital PUFs [10]. The strongly confined optical signals severely reduce the leakage of portions of information across the chip, thus providing enhanced security against side-channel attacks.

In NEUROPULS, we investigate various types of photonic architectures for weak and strong PUFs, which can be schematically summarized in Fig. 2. Here, we have a telecom laser source that is modulated by means of an optical modulator (OM) driven by an ASIC. The light beam enters a passive architecture (no active devices are present) featuring many photonic components. In particular, they affect the electric field of the optical beam not only in amplitude, by splitting it into multiple optical waveguides and introducing losses, but also in phase. Memory effects, e.g., for resonant devices, will also be used to mix up incoming signals in time with previous ones, therefore having past bits interacting with present ones, similarly to what happens in reservoir computing. At the output of this class of architectures, we aim to use non-linear devices such as photodiodes (PDs) that are sensitive not only to the amplitude but also to the phase of the light field due to the coherence of the approach. The ASIC then processes the responses through transimpedance amplifiers (TIAs) and analog-to-digital converters (ADCs). The collected signals are then corrected by various means, for example, using error correction codes (ECCs) to account for potential deviations in the case of weak PUFs (see Fig. 1). Finally, the post-processed responses are sent to the software layer by means of a RISC-V interface. We recently proposed and demonstrated a PUF architecture based on microring resonator arrays according to the scheme of Fig. 2 capable of achieving very good statistical performance (fractional Hamming distance close to 50% intra and inter-device and good score for various NIST tests) [12]. This architecture worked at 25 Gbit/s (based on a Mach-Zehnder modulator) and was considered on a Silicon-On-Insulator (SOI) platform. This mode of operation is in net contrast with, e.g., the operation of an arbiter PUF where each bit of the challenge affects a single switch block which has only 2 possible outcomes aside from the introduced delay which is additive. Future work in NEUROPULS will further investigate these architectures and how to achieve not only good statistical properties, but also improve their robustness against fluctuations and ML attacks with respect to the full platform. In particular, the next section will discuss some techniques that we are considering in NEUROPULS to improve the reliability of these primitives.

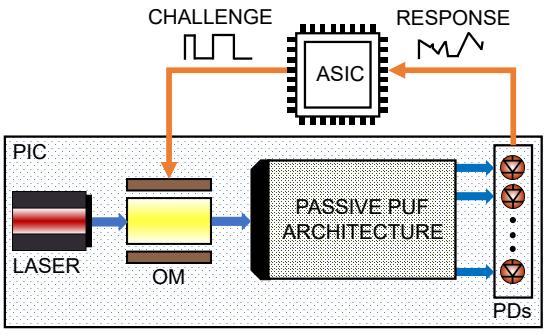

Fig. 2. Schematic of the PUF operation considered in NEUROPULS. OM: optical modulator, PDs: photodiodes. The passive PUF architecture section separates the initial light beam in several different paths and scrambles them before the output. No active devices are present. In the final demonstrator, all the PIC components shown and ASIC-related PUF circuitry will constitute the actual PUF.

B. Techniques for PUF quality improvement

Vinagrero et al. in [13] developed a simulation-based filtering algorithm that computes the probability of bit aliasing, exploiting the inherent relationship between reliability and bit aliasing. For RO-based PUFs, pairs of ROs with low frequency differences are known to be unreliable. However, pairs of ROs with the highest frequency difference cannot be chosen either, as the influence of process variability will be weakened, thus making the responses very similar among different devices.

Indeed, frequency differences close to the response selection boundary tend to provide the maximum amount of entropy due to the Gaussian nature of the random source, but can be deemed unreliable since they are more prone to bit flips due to noise or environmental conditions. However, frequency differences that lie further from the selection boundary could be deemed biased (aliased). Extreme values of frequency difference could be present in multiple devices because of the lower effect of process variability and, consequently, present aliasing.

Fig. 3 reports an example of this phenomenon [13]. Bit-aliasing is represented as Shannon entropy; values close to 1 represent no bit-aliasing, while values close to 0 represent aliasing. The shaded area represents a good trade-off between bit aliasing, reliability, and the number of challenge-response pairs (CRPs). This technique can be tailored to different PUF architectures by adapting the threshold selection to the key generation mechanism of the PUF. For delay-based PUFs, this filtering technique is based on a threshold on the count or frequency difference of the RO pair used. In NEUROPULS, we will use a similar approach, where instead of considering a counting threshold, we will consider a threshold dependent on the amplitude of the photocurrent read at the PD. Other techniques will also be considered to reduce the effect of fluctuations, such as introducing a photonic sensor for temperature measurement and considering this additional parameter when evaluating the genuinity of the responses. Hardware approaches based on the temperature controller will also be used to reduce reliability concerns.

III. Proposed Services and Related Security Protocols

A. Mutual authentication

Authentication is the first step in secure communication, which consists of verifying the identity of a participant (e.g., a device) before exchanging sensitive data with the participant. The context of NEUROPULS includes two main roles: the device to be verified and an external entity acting as the verifier. This context imposes two requirements: First, the authentication shall be *mutual*, i.e., considering that sensitive data travel in both directions, the device and the verifier should verify each other's identity. Second, the authentication process shall be lightweight because the resources on the device are constrained.

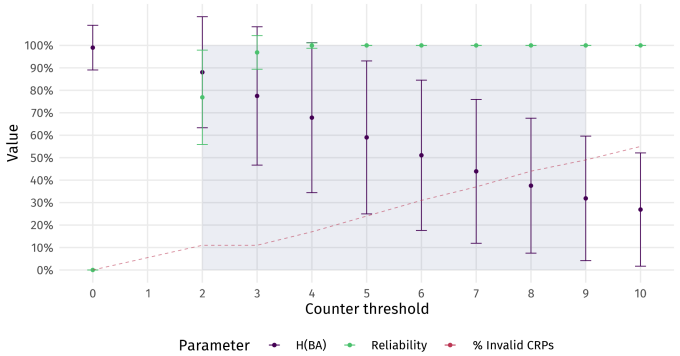

Fig. 3. Relationship between bit aliasing and reliability taking into account the counter threshold

Existing authentication strategies based on PUFs require the verifier to store a large database of CRPs for each device, as first described in seminal works on PUFs [14], [15] and detailed by Suh et al. [16], or to exploit heavier protocols [17], [18]. However, to meet the lightweight requirement, we decided to adopt a different strategy, that is, *HSC-IoT*, the authentication procedure proposed by Hossain et al. [19]. Their idea is to use only a single CRP as a shared secret between the device and the verifier to support mutual authentication and to update it after each use with a fresh CRP.

Practically, the first CRP is shared at manufacturing time and is meant to support the first actual authentication session. At each actual authentication session, the device uses a fresh CRP that is based on the response of the previously used CRP. The new response is sent to the verifier in encrypted form and, if mutual authentication succeeds, the current CRP is updated on both the device and the verifier.

Fig. 4 contains a UML sequence diagram showing messages exchanged between the device and the verifier according to the mutual authentication protocol. The protocol starts with the authentication request from the verifier. The device derives the new challenge c_{i+1} from the current response r_i, using it as a seed for a pseudo-random number generation (RNG) function known to both participants, $c_{i+1} = RNG(r_i)$. The device computes a message m containing the new response r_{i+1}, XORed with the current response r_i. In the figure, the operator ^ denotes the XOR operation and || denotes concatenation. The message may also contain proof of the integrity of the software, such as the hash of the memory H and a clock count CC

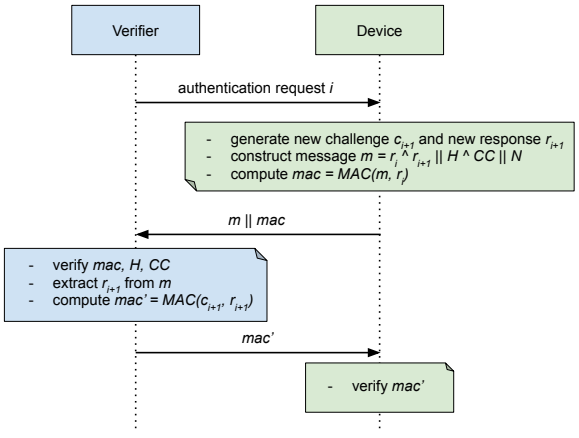

Fig. 4. Session i of the mutual authentication protocol.

that represents the time needed to perform a given task. The message can contain a nonce N for freshness. This message m is then sent to the verifier along with its MAC signature, calculated using the $MAC(data, key)$ function, whose first argument is the data to sign and the second is the key, r_i in this case. Now, the verifier can authenticate the device by checking the message signature using the secret r_i. The new response r_{i+1} is derived from m and stored in the verifier to be used as a shared secret in the next authentication session. Finally, the verifier authenticates to the device by demonstrating that it knows the new secret r_{i+1}, which is used to sign c_{i+1} (generated using r_{i+1}) through the MAC function again.

This protocol only needs one CRP to be known by the verifier at any point, which is more scalable than other solutions that require a large database of CRPs. In addition, CRPs are kept confidential because they are never exchanged in clear text. Although this protocol is very lightweight, it is designed to resist many attacks, as discussed by Hossain et al. [19].

B. Software attestation

Remote software attestation validates the integrity status of remote computing devices without relying on secure hardware components such as Trusted Platform Modules (TPMs). Such specialized components are often unsuitable for devices with limited resources [20], [21]. This approach enables remote systems to detect malicious or unintended changes in the firmware, software, or hardware that operates on these devices.

Attestation mechanisms generally send a hash of the device's memory to the verifier to prove that the device is not compromised [22]. An example of this is given in the previous section; during mutual authentication, the algorithm can include a hash of the memory, which provides some proof of the device's integrity. In this section, we describe a more powerful approach that, however, imposes stronger assumptions, i.e., it leverages an ideally reliable strong PUF and on a PUF model available to the verifier. To avoid attacks where a device hides its compromised memory regions from verification (e.g., by moving these regions around while the algorithm hashes the uncompromised memory), attestation protocols often employ temporal constraints that guarantee the unfeasibility of these attacks [23]. An important part of these protocols is the root of trust, a part of the system proven not to be compromised, on which the protocol can base its operations. Without being able to use secure hardware modules (i.e. TPMs) as a root of trust, many modern protocols have started adopting PUFs as an alternative. In our framework, we leverage the photonic PUF (pPUF) embedded in the neuromorphic accelerator to generate many CRPs that can then be used to hash different areas of the device's memory.

The verifier starts the attestation by crafting a message, the attestation request, that contains a timestamp t and a challenge c_1. The message is then sent to the device, which then promptly starts the attestation process by using the issued challenge to compute a response r_1 in the pPUF. The response is combined with the timestamp as the seed for an RNG that generates the random walk in memory: $m_1, \ldots, m_n = RNG(r_1 + t)$. A secure hash algorithm is then used with the initial chunk of memory m_1 on the device and r_1, generating the first hash $h_1 = HASH(m_1, r_1)$, while r_1 is used simultaneously as the next challenge for pPUF, as $r_1 = pPUF(r_1)$. All subsequent hashes depend on the previously calculated ones: $h_{i+1} = HASH(m_{i+1}, r_{i+1}, h_i)$. The inherent speed of the pPUF (at least 5 Gb/s) guarantees that the constant challenge-and-response generation never slows down the protocol, so the temporal constraints of our approach can be stricter than those found in previous work. After exhausting all memory regions, the final hash, h_n, is sent to the verifier. This protocol minimizes the network load by having a very small footprint in both the attestation initiation and its finalization, which allows our temporal constraints to be mostly focused on the speed of the iterative hash function. The verifier has a copy of the uncompromised device memory and a model of the pPUF, so it can start to calculate h_n right after generating the attestation request. After receiving h_n from the device, the verifier checks that its value is correct and that the attestation did not exceed its temporal constraints; if both of these requirements are satisfied, the attestation is successful. Otherwise, a new request is issued and the protocol restarts with a new timestamp and challenge.

TABLE I
FUNCTIONS TO LOAD AND EXECUTE A NEURAL NETWORK

Function name	Parameters	Results
`load_network`	`ciphered_network`	
`execute_network`	`ciphered_input`	`ciphered_output`

C. Neural network configuration and data encryption

Another important security requirement is the confidentiality of the actual data processed by the accelerator and the confidentiality of the neural network configuration. To meet this confidentiality requirement, encryption encodes the data in all communications with external parties and all the software running on the device. The NN configuration, the input data to the device, and the computation output are encrypted using the secret keys described in Section II. This key is never exposed to the software layer, but encryption and decryption occur on the hardware in the implementation of the security primitives shown in Table I. `load_network` receives the neural network configuration in encrypted form. The configuration is decrypted in hardware and loaded in the accelerator. `execute_network` takes the input as a decrypted parameter and fed to the accelerator. Then, the computation result is encrypted and returned by this function. As specified by the function signatures, data are never exposed in plaintext to the software. Data are decrypted internally by the device using primitives that never leave plaintext in the memory after execution, thus preserving confidentiality even against an internal attacker capable of reading the RAM.

IV. STRENGTHS AGAINST ATTACKS

The core of the security services to be provided in NEUROPULS are supported by the use of PUFs intrinsically bound at both the PIC and the ASIC levels. This protects our NN accelerator from tampering attacks where one malicious chip could replace the genuine PIC or control ASIC. The robustness of this security mechanism is also possible thanks to the effectiveness achieved by the physical connection between chips

that are highly dependent on the packaging (e.g., using high-frequency wire-bonding). Components such as transimpedance amplifiers (TIAs) and ADC modules further modify the photonic PUF response, and thus it is possible to generate a composite response from the 2 chips, which can be used to assess the genuine character of the accelerator as a whole.

In NEUROPULS, we will investigate the robustness of the security layers against various types of attacks, with a particular focus on attacks targeting the PUF, ranging from machine learning (ML) modeling to side-channel attacks. We will specifically look at attacks that tamper with the responses received on the PD array or at how laser power levels can be altered to produce responses that can provide insights into the inner working mechanisms of the PUFs. In particular, effects such as bit flips will be addressed from both a theoretical and an experimental point of view in the strings sent from the driving ASIC to the PIC. Photonic PUFs do not suffer from RF leakage, contrary to many electronic PUF solutions. In the latter case, RF signals can be detected, for example, from the Si substrate, as was pointed out in various works. Therefore, by performing a power analysis, it was possible to extract key information about PUF behavior and carry out modeling attacks [9], [24].

The capability of transferring information in photonic waveguides where signals leak out only a few hundred nanometers hinders side-channel attacks. Although these attacks can still potentially occur at the interface between the PIC and the ASIC (see Fig. 2), the analog and high-speed nature of such signals and the fact that the signals will be processed further at the ASIC level bring several additional layers of complexity that require a very expensive way to break such solutions. Furthermore, although in our prototype the ASIC and the PIC will be connected by wire bonding, approaches in which the electronics are integrated with the photonics in a monolithic way will render this potential attack point useless, as the information will be electronically manipulated locally where it was processed optically [25], [26].

Furthermore, the photonic PUF discussed in Fig. 2 operates in a time domain; therefore, its response is present only during the interrogation time and then disappears. Therefore, attacks based on the remanence decay time cannot be used, as in SRAM PUFs that share memory with other functionalities [27]. Another strength comes from the fact that the response is present in the PUF for a very short period of time (below 100 ns), making its potential extraction very complex. ML modeling of PUFs is another common way to attack PUFs. In such a case, by acquiring a sufficiently large number of CRPs (for strong PUFs), the adversary can build a model to predict the response to the next challenge. This approach could then be used to impersonate the system containing the PUF, thus breaking the security of the layer and the system as a whole. These attacks have been particularly successful against common types of PUF, such as PUFs with ring oscillators (ROs) or arbitrators [28]. The main weakness of this type of PUF lies in the relatively small number of components and variables that participate in processing the challenge to generate the response. Photonic PUFs are expected to provide a greater gain with respect to ML attacks because of the much larger number of components and, especially, variables that participate. Various examples of optical/photonic PUFs resistant to this type of attack have been previously shown in the literature [29]. In NEUROPULS, we will investigate strong PUFs that exploit non-linearities at different levels, as well as architectural solutions that rely on the combination of a strong and a weak PUF to encrypt challenges before entering the photonic PUF as we previously proposed for purely electronic PUFs [30]. To further enhance the robustness and security of the PUF use in the authentication process, while also generating session keys for the data encryption, an Authentication and Key Agreement (AKA) protocol can also be considered. AKA can protect the PUF responses in such a way that an attacker cannot guess or brute-force the protocol to find the CRP. One approach to achieve this is to see the CRP as a low-entropy shared secret. With this, we can consider the use of the well-established and secure EKE protocol to achieve mutual authentication and key exchange, which will be used in the implementation of the secure channel. This approach protects against most possible attacks on the CRP while providing perfect forward security to the key established for data encryption. Note that this approach is computationally more expensive. However, lighter AKA protocols can be considered [31] depending on the target security and robustness of the achieved PUF.

V. System-Level Modeling and Benchmarking

Building a simulator capable of modeling the behavior of security primitives, such as PUFs, requires modeling all system components (CPU, memory, accelerators) and implementing suitable benchmarks to evaluate the performance and security characteristics. Regarding system-level modeling, it is essential to define the PUF architecture and specify its fundamental components, such as CRPs and memory elements, together with the interface for programming them. Environmental factors, including temperature, voltage, and variations in the manufacturing process, must also be simulated to account for their potential impact on the behavior of the PUF. Furthermore, noise and other sources of variability should be modeled to fully assess the behavior of the PUF. The simulator should also account for the interactions between the PUF and other system components, such as cryptographic modules and key management systems. Parameters for configuration and tuning should be included to simulate the effects of different settings on PUF performance and security. The gem5 [32] simulation environment allows one to define a peripheral module connected to the RISC-V microprocessor, providing the essential infrastructure for the delivery of the programming API. Furthermore, the simulator should incorporate models for potential attacks on the PUF, as described in Section IV. A holistic approach to modeling and simulating a heterogeneous system is required, including RISC-V (or other ISA) CPUs and electronic or photonic accelerators [11], [33], [34].

As defined in Section II, reliability metrics are crucial for benchmarking, including evaluating the reliability of PUF responses under different conditions, such as environmental variations and the effects of aging. The gem5-provided log

facility allows data collection to assess entropy, uniqueness, and response uniformity to ensure that the modeled PUF provides a probability of random and unique identifiers. In addition, error rates, including false-positive and false-negative rates, should be analyzed to gauge the PUF's reliability. The PUF model (with its drivers, detectors, and sources) will initially be generated from PIC simulations and then abstracted for appropriate integration into the gem5-based simulation framework using statistical approaches. The communication link between the PUF and the processor, including the packaging aspects, will also be modeled to accommodate realistic implementation. Parameters such as throughput, latency, and power consumption that can be obtained from the abstract PUF model are essential to understand the performance of PUFs in real-world applications. Lastly, the simulator should evaluate the PUF's robustness against various attacks, encompassing ML-based, side-channel, and invasive attacks. By incorporating these considerations into the simulator, a comprehensive tool can be created to analyze and optimize the performance and security characteristics of PUFs in various scenarios.

REFERENCES

[1] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[2] Y. Xiao *et al.*, "Edge computing security: State of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[3] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on multi-access edge computing security and privacy," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1078–1124, 2021.

[4] I. Butun, A. Sari, and P. Österberg, "Hardware security of fog end-devices for the internet of things," *Sensors*, vol. 20, no. 20, p. 5729, 2020.

[5] P. Kocher *et al.*, "Spectre Attacks: Exploiting Speculative Execution," in *2019 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2019, pp. 1–19. [Online]. Available: https://ieeexplore.ieee.org/document/8835233/

[6] M. Lipp *et al.*, "Meltdown," *arXiv preprint arXiv:1801.01207*, 2018.

[7] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical One-Way Functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002. [Online]. Available: https://www.science.org/doi/10.1126/science.1074376

[8] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon Physical Random Functions," p. 13.

[9] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (puf)-based security solutions for internet of things," *Computer Networks*, vol. 183, p. 107593, 2020.

[10] F. Pavanello, I. O'Connor, U. Ruhrmair, A. C. Foster, and D. Syvridis, "Recent Advances in Photonic Physical Unclonable Functions," in *2021 IEEE European Test Symposium (ETS)*. Bruges, Belgium: IEEE, May 2021, pp. 1–10. [Online]. Available: https://ieeexplore.ieee.org/document/9465434/

[11] F. Pavanello *et al.*, "Neuropuls: Neuromorphic energy-efficient secure accelerators based on phase change materials augmented silicon photonics," in *2023 IEEE European Test Symposium (ETS)*, 2023, pp. 1–6.

[12] P. Jimenez *et al.*, "Photonic physical unclonable function based on symmetric microring resonator arrays," in *Frontiers in Optics*, 2023.

[13] S. V. Gutierrez, G. Di Natale, and E.-I. Vatajelu, "On-line method to limit unreliability and bit-aliasing in ro-puf," in *2023 IEEE 29th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2023, pp. 1–6.

[14] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1074376

[15] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, V. Atluri, Ed. ACM, 2002, pp. 148–160. [Online]. Available: https://doi.org/10.1145/586110.586132

[16] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007*. IEEE, 2007, pp. 9–14. [Online]. Available: https://doi.org/10.1145/1278480.1278484

[17] S. W. Jung and S. Jung, "HRP: A hmac-based RFID mutual authentication protocol using PUF," in *The International Conference on Information Networking 2013, ICOIN 2013, Bangkok, Thailand, January 28-30, 2013*. IEEE Computer Society, 2013, pp. 578–582. [Online]. Available: https://doi.org/10.1109/ICOIN.2013.6496690

[18] Y. Lee, H. Lee, and E. Alasaarela, "Mutual authentication in wireless body sensor networks (WBSN) based on physical unclonable function (PUF)," in *2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013, Sardinia, Italy, July 1-5, 2013*, R. Saracco, K. B. Letaief, M. Gerla, S. Palazzo, and L. Atzori, Eds. IEEE, 2013, pp. 1314–1318. [Online]. Available: https://doi.org/10.1109/IWCMC.2013.6583746

[19] M. M. Hossain, S. A. Noor, and R. Hasan, "Hsc-iot: A hardware and software co-verification based authentication scheme for internet of things," in *5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2017, San Francisco, CA, USA, April 6-8, 2017*. IEEE Computer Society, 2017, pp. 109–116. [Online]. Available: https://doi.org/10.1109/MobileCloud.2017.35

[20] C. Basile, S. Di Carlo, and A. Scionti, "Fpga-based remote-code integrity verification of programs in distributed embedded systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 187–200, 2012.

[21] M. N. Aman *et al.*, "Hatt: Hybrid remote attestation for the internet of things with high availability," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7220–7233, 2020.

[22] W. Feng, Y. Qin, S. Zhao, and D. Feng, "Aaot: Lightweight attestation and authentication of low-resource things in iot and cps," *Computer Networks*, vol. 134, pp. 167–182, 2018.

[23] M. N. Aman and B. Sikdar, "Att-auth: A hybrid protocol for industrial iot attestation with authentication," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5119–5131, 2018.

[24] U. Rührmair *et al.*, "Efficient power and timing side channels for physical unclonable functions," in *Cryptographic Hardware and Embedded Systems–CHES 2014: 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings 16*. Springer, 2014, pp. 476–492.

[25] A. H. Atabaki *et al.*, "Integrating photonics with silicon nanoelectronics for the next generation of systems on a chip," *Nature*, vol. 556, no. 7701, pp. 349–354, 2018.

[26] C. Sun *et al.*, "Single-chip microprocessor that communicates directly using light," *Nature*, vol. 528, no. 7583, pp. 534–538, 2015.

[27] S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl, and A.-R. Sadeghi, "Remanence decay side-channel: The puf case," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1106–1116, 2015.

[28] U. Rührmair *et al.*, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM conference on Computer and communications security*, 2010, pp. 237–249.

[29] B. T. Bosworth *et al.*, "Unclonable photonic keys hardened against machine learning attacks," *APL Photonics*, vol. 5, no. 1, p. 010803, Jan. 2020. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.5100178

[30] E. I. Vatajelu, G. Di Natale, M. S. Mispan, and B. Halak, "On the encryption of the challenge in physically unclonable functions," in *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2019, pp. 115–120.

[31] P. Mall, R. Amin, A. K. Das, M. T. Leung, and K.-K. R. Choo, "Puf-based authentication and key agreement protocols for iot, wsns, and smart grids: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8205–8228, 2022.

[32] J. Lowe-Power *et al.*, "The gem5 simulator: Version 20.0+," 2020. [Online]. Available: https://arxiv.org/abs/2007.03152

[33] O. Chatzopoulos, G. Papadimitriou, V. Karakostas, and D. Gizopoulos, "Enabling design space exploration of risc-v accelerator-rich computing systems on gem5," in *RISC-V Summit Europe (RISC-V Europe 2023)*, 2023, pp. 1–2.

[34] ——, "gem5-marvel: Microarchitecture-level resilience analysis of heterogeneous soc architectures," in *IEEE International Symposium on High-Performance Computer Architecture (HPCA 2024)*, 2024.