PUF-based Authentication in IoT against Strong Physical Adversary using Zero-Knowledge Proofs

Lukas Petzi[1], Alexandra Dmitrienko[1], Ivan Visconti[2]
[1]University of Würzburg [2]University of Salerno

***Abstract*—This work focuses on utilising Physically Unclonable Functions (PUFs) for device authentication, exploiting a device's unique manufacturing-induced hardware variations. Traditional PUF-based authentication methods often rely on trusted third parties for validation or necessitate that Verifiers maintain large databases. Existing approaches that aim to reduce storage demands by reutilizing information typically address only network-level threats, leading to doubts about the necessity of PUFs, or they focus exclusively on adversaries aiming at non-volatile memory. This paper introduces a classification guideline that delineates the scenarios in which PUFs are necessary or advantageous. Additionally, we present a novel PUF-based authentication scheme that incorporates challenge concealment to safeguard against comprehensive invasive physical attacks. This method offers *perfect* hiding, an enhanced level of security compared to previous models that permitted the reusing of PUF challenges. Through this approach, we aim to provide a more secure yet efficient framework for PUF-based authentication, addressing the limitations of current methodologies and extending the protection against a broader spectrum of adversaries.**

I. Introduction

The Internet of Things (IoT) includes a wide array of communication-enabled devices. Considering the extensive interaction among these devices, alongside their capabilities to generate and share data, ensuring the authenticity of IoT devices emerges as a paramount concern. Authentication involves a party providing evidence that it possesses or knows a secret, such as a password. Securing credentials on an IoT platform poses significant challenges, especially from physical threats: Vulnerabilities due to an attacker's physical access to the device can result in credentials being compromised and subsequently used for impersonation. In practice, some manufacturers use non-volatile storage, like ROM or EPROM, to store keys. While this approach guarantees key integrity, it still necessitates additional measures to safeguard their confidentiality, e.g., in case of offline physical attacks [1], [2].

In 2002, it was discovered that devices exhibit unique hardware characteristics due to inherent manufacturing variations [3]. These characteristics are difficult to clone and distinct for every device. Consequently, the notion of Physically Unclonable Functions (PUFs) arose, and PUFs were treated akin to a human fingerprint and employed for device authentication. Since these device fingerprints are extracted from the underlying hardware at runtime, PUF-based authentication alleviates the burden of storing secret keys on the device.

In PUF-based authentication, the device possessing the PUF called Prover must convince another party, the Verifier, that they indeed have the authentic PUF. To conduct the verification process, the Verifier must possess a piece of certain PUF-related information, which needs to be securely stored. Such a Verifier is trustworthy from the point of view of a Prover, as that information can be used to impersonate Prover.

PUFs are categorized as either weak or strong, depending on their ability to produce unique Challenge-Response Pairs (CRPs). Weak PUFs can generate only a limited set of CRPs, typically proportional to the challenge size, while strong PUFs are capable of producing an exponential number of CRPs [4].

Most prior PUF-based authentication systems leverage strong PUFs to generate a new CRP for each authentication session. Typically, these CRPs are pre-generated and utilized in subsequent authentication processes. This necessitates the Verifier securely storing a substantial quantity of CRPs. The requirement for the Verifier to possess adequate storage capacity restricts the use of such systems in IoT networks, as resource-constrained devices common in these environments are often unsuitable to serve as Verifiers, thus limiting the applicability of these systems in IoT contexts.

Several methods have been developed to mitigate this issue by facilitating the reuse of a single CRP, thereby diminishing the storage demands on the Verifier. This also allows for compatibility with weak PUFs. Commonly, this is achieved by incorporating the PUF response as the confidential component in an asymmetric key pair, coupled with a signature scheme for authentication purposes [5]–[8]. However, these approaches only assume limited physical access only where the attacker can compromise non-volatile memory.

In this work, we present a comprehensive methodology designed to evaluate the necessity and potential benefits of incorporating a Physical Unclonable Function within a device. This evaluation is grounded in an analysis of the device's available hardware capabilities and the specific threats posed by the considered adversary model. Through this approach, we aim to not only determine the appropriateness of employing a PUF but also to pinpoint the most fitting type of PUF tailored to the scenario under consideration.

Additionally, we introduce a new PUF-based authentication scheme that allows the reuse of challenges and eliminates the need for the Verifier to store large quantities of confidential information. This allows the Verifier to be implemented on resource-constrained IoT devices. Moreover, our design provides perfect hiding of the used response, a stronger security guarantee than previous schemes. Additionally, our scheme provides challenge concealment to protect against invasive physical adversaries.

Contributions. We present the following contributions:

- We evaluate the suitability of PUFs for specific application scenarios and introduce a methodology to guide decision-making. This approach recommends if a PUF should be used and advises the optimal PUF type based on device capabilities and potential security threats.
- Our innovative PUF-based authentication approach combines a perfectly hiding commitment scheme with perfect zero-knowledge proofs, ensuring response correctness is verified without disclosure. It also hides the challenge during authentication, enabling challenge reuse even if the device is compromised between sessions. This strategy reduces the Verifier's storage demands and protects against a stronger adversarial model than previous schemes that reuse challenges.
- We provide a prototype implementation of the proposed authentication system using the LPC55S69-EVK evaluation board produced by NXP. This board is a readily accessible development platform, exemplifying the practicality and suitability of the proposed scheme for devices with constrained resources.
- Comprehensive analysis of the performance of both the authentication and verification processes, as illustrated through our reference implementation, demonstrates that execution time can be as low as 0.3 seconds even on a resource-constrained device. Additionally, an analysis of the storage requirements is presented.

In summary, this paper introduces a new methodology that takes into account various adversary capabilities to ascertain the utility of a PUF and, if beneficial, to identify the most appropriate type. Furthermore, we present the first PUF-based authentication scheme that addresses the potential temporal compromise of the Prover device and eliminates the need for the Verifier to hold large volumes of sensitive information.

Outline: The remainder of the paper is organized as follows: Section II provides key background details. Section III outlines our methodology for selecting the appropriate PUF. Section IV details our system model and proposes our new PUF-based authentication scheme. Section V discusses our implementation and evaluation. Section VI examines existing related work on PUF-based authentication, and Section VII ends with conclusions.

II. Background

In this section, we present the necessary background information on PUF-based authentication, cryptographic commitment schemes, and zero-knowledge proofs.

A. Authentication based on Physical Unclonable Functions

The standard PUF-based authentication protocol comprises two phases: *Enrollment* and *Authentication*.

1) **Enrollment Phase:** In this phase, the Verifier initiates a random set of challenges $\{C_1, C_2, ..., C_n\}$. Next, the Verifier requests the corresponding responses $\{R_1, R_2, ..., R_n\}$ from the Prover's PUF. The resulting Challenge-Response Pairs (CRPs) $\{R_1C_1, R_2C_2, ..., R_nC_n\}$ are stored in a database DB maintained by the Verifier. This one-time process must be conducted in a secure environment to protect the confidentiality and integrity of the responses.
2) **Authentication Phase:** The Verifier initiates the authentication by randomly selecting a Challenge-Response Pair (C_i, R_i) from its database and transmits that challenge C_i to the Prover, who inputs it into its PUF to generate a response $R = PUF(C_i)$. The Prover sends this response back to the Verifier who compares the received response with the one stored in its database and accepts the authentication if they match and rejects it otherwise. Afterward, the used CRP gets deleted from DB (to prevent reuse).

B. Commitment Schemes and Zero-Knowledge Proofs

A cryptographic commitment scheme is a two-party protocol that occurs over two rounds and involves a committer C and a receiver R. In the first round, the committer commits to a chosen message m from a message space M while ensuring that the message remains concealed from all other parties involved. The committing party can later reveal the committed value at a chosen time by opening the commitment [9]. Every commitment scheme is underpinned by two pivotal security properties: *hiding* and *binding* [9]. Very roughly, the *hiding* property guarantees that the committed values remain concealed, ensuring they cannot be deduced merely by examining the commitment. In contrast, the *binding* property assures that once a commitment has been created it cannot be opened to an m' with $m' \neq m$.

Zero-knowledge proofs [10], are sophisticated cryptographic protocols enabling a party, termed the Prover, to validate the truth of a statement or confirm possession of specific information to another party, the Verifier, without disclosing the actual information or any extra knowledge beyond the inherent implications of the statement. Essentially, these protocols permit the Prover to assure the Verifier of its knowledge of confidential information without revealing the information itself.

III. When does a PUF make sense?

The decision to employ either a weak or strong PUF is primarily motivated by the need to protect sensitive information, such as cryptographic keys, from potential adversaries. To determine whether a platform or IoT device should incorporate a PUF, and if so, which type of PUF is most suitable, it is crucial to evaluate the adversary model alongside the device's other hardware capabilities. Figure 1 systematically outlines the decision-making process for ascertaining the necessity and type of PUF to be used. This discussion is particularly focused on the authentication use-case, given the frequent application of PUFs in this area; however, the same reasoning applies to other scenarios, such as the protection of intellectual property using PUF [11], [12].

In scenarios where the threat model only includes network-level adversaries, the use of a PUF may be deemed unnecessary. In such cases, cryptographic information, including secret

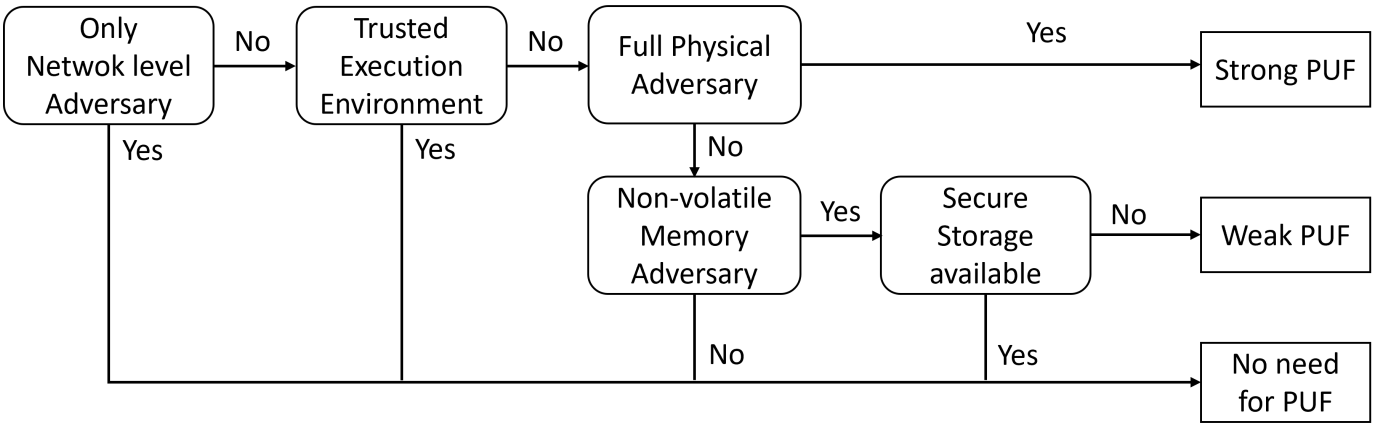

Fig. 1. Flow chart to determine whether and which Physical Unclonable Function is the appropriate technical solution considering the adversary model as well as the hardware capabilities of the platform.

keys for authentication, can be securely stored in non-volatile memory. Given that the adversary's capabilities are confined to network access without the ability to interact with the device physically, the protection of confidential data on the device itself is not a concern, negating the need for a PUF.

However, in situations extending beyond mere network-level threats, a more detailed examination is warranted. If the device is already equipped with a Trusted Execution Environment (TEE), such as ARM TrustZone [13] or Intel Trusted Execution Technology (TXT) [14], the deployment of a PUF might be redundant. The TEE can safeguard cryptographic keys and ensure that sensitive information remains inaccessible or unusable outside of the TEE. Thus, cryptographic keys can be securely stored within a functioning TEE, and operations requiring these keys, such as authentication, are confined to this isolated environment. Even in the presence of a platform-level adversary, the TEE's isolation ensures that access to secret information is effectively blocked.

But since TEEs represent complex and costly hardware security enhancements they are often unavailable on resource-constraint IoT devices. In the absence of a TEE, the adversary model requires deeper analysis to select the appropriate PUF type. In scenarios involving a physical adversary capable of interacting with an operating device, the threat model includes the adversary's ability to present challenges to the PUF and observe the resultant responses. Given this, a weak PUF is deemed inadequate for defence against such an adversary, primarily because of its constrained ability to produce a large set of CRPs. The limited CRP generation capacity of a weak PUF exposes it to the risk of being entirely deciphered through brute-force attacks by the adversary. Consequently, the employment of a strong PUF is advocated. Unlike its weaker counterpart, a strong PUF is characterized by its capacity to generate an exponentially large pool of potential CRPs. This vast number of possible CRPs acts as a deterrent, making it impractical for the adversary to exhaustively brute-force all combinations, thereby offering a robust layer of security against physical tampering and unauthorized access.

For adversaries targeting non-volatile memory [15], a weak PUF suffices. Such adversaries typically execute storage extraction attacks on unpowered devices, aiming to access and download flash memory content. An example is stealing a publicly mounted IoT device to later impersonate it by extracting secrets stored in non-volatile memory. While secure storage solutions like the Trusted Platform Module (TPM) provide strong protection for secret keys, making PUFs potentially unnecessary for devices with robust secure storage, PUFs become essential in their absence. PUFs negate the need to store sensitive data in easily accessible non-volatile memory by generating necessary information on demand. This approach ensures data remains secure against physical access. Since the adversary can only access non-volatile memory content and not the PUF, a weak PUF is sufficient.

IV. Design

This section first outlines our system model, followed by the considered adversary model. After this analysis, the design details of our scheme are presented.

A. System Model

We consider an IoT network consisting of multiple heterogeneous devices deployed in a controlled environment maintained and operated by a stakeholder.

A Smart Factory Line [16], serves as a prime illustration of such an interconnected network ecosystem. In Smart Factories, numerous devices are strategically placed throughout production lines to monitor and automate manufacturing steps. These devices are interconnected to share vital information, often within a command-and-control framework that delegates tasks along the production line. To access a service, a device is required to prove its identity and is thus termed the Prover. The authentication process involves the Prover submitting proof of identity, which is subsequently validated by the service host, known as the Verifier. Typically, in PUF-based authentication protocols, the Prover generates this proof by responding to a specific challenge using its integrated PUF.

B. Adversary Model

Our adversary possesses a dual set of capabilities. Firstly, it is assumed to have frequent full physical access to a device.

This includes feeding challenges to the PUF and reading the corresponding responses. This scenario could involve a malicious employee working within a smart factory, who gains access to a device when unobserved or during maintenance tasks, but not during authentication. Additionally, the adversary can passively intercept information transmitted via the network. This may be achieved by deploying a device discreetly within the factory to monitor wireless traffic or by accessing unsecured log files containing network information. However, it's important to note that the adversary is unable to directly connect their device to the internal factory network and thus is limited to passive network attacks. This means it cannot actively transmit any messages.We assume that the underlying PUF structure is resilient against attacks, such as Machine Learning-based modeling. Resilience against such attacks is more a characteristic of the PUF implementation and, hence, orthogonal to the protocol. Also, while some PUF implementations have been identified as vulnerable [17], others are still considered secure [18], [19].

C. System Design

Our scheme introduces a novel strategy that employs a single challenge across multiple authentication sessions, crucially ensuring the perfect confidentiality of the PUF response as well as its corresponding challenge. This approach stands in contrast to previous methods, where PUF challenges are transmitted in plaintext and PUF responses are used as secret keys in signature schemes [5]–[8]. While these existing schemes allow for the reuse of PUF responses, they fall short in achieving perfect hiding of the used response. This security property is necessary as despite the theoretical idealizations of PUF-based systems, practical implementations of PUFs have been shown to leave potential vulnerabilities open to exploitation [20]–[22]. Given the inherent security risks associated with PUFs, it is paramount that no additional information is leaked during the authentication process. Therefore, our scheme's emphasis on the perfect hiding of the used PUF response is a critical measure to ensure that adversaries do not gain any supplementary information, supporting the overall security of the system.

Furthermore, our scheme conceals the challenge during the authentication process by employing ephemeral Diffie-Hellman for forward secrecy [23]. Therefore, it remains robust against adversaries who can eavesdrop and temporarily access the device physically, overcoming limitations of previous PUF-based authentication schemes, allowing CRP reuse, that only addressed non-volatile memory or network-level threats, without accounting for full physical adversary access.

In our approach, a Prover uses a perfectly hiding commitment scheme to commit to a PUF-response during enrollment, and the commitment is then published. Since commitments are designed to conceal their contents, the Prover, during the authentication phase, must demonstrate knowledge of the secret for authentication purposes without actually disclosing it. This approach is crucial for maintaining the reusability of the PUF-response. This is achieved through zero-knowledge proofs, where the Prover demonstrates knowledge of the secret within the commitment without disclosing it, ensuring the Verifier can confirm this knowledge while maintaining the response's confidentiality for future authentication.

Similar to other PUF-based authentication systems, ours involves a one-time *Enrollment Phase* and an *Authentication Phase*. We will first outline the necessary system parameters before delving into the specifics of our scheme.

Protocol Parameters: These values include the description of a cyclic prime order group G in which the discrete logarithm problem is computationally infeasible. Furthermore, there are two generators, g and h, with the requirement that the discrete logarithm of h relative to the base g (i.e., finding an α such that $h = g^{\alpha}$) remains unknown. Additionally, we require a public cryptographic hash function H. Once the system parameters are established, they remain consistent for every device utilizing the scheme. These parameters are selected during the initialization of the network and can be for instance chosen from standardized parameters as delineated in TLS [24]. In obtaining the second generator h we leverage public randomness [25]. Alternatively, h can be derived from g with $h = H(g)$ with H modeled as a random oracle.

Enrollment Phase: The Enrollment Phase is a one-time procedure conducted in a secure environment, by the device's stakeholder. To enrol the device, the PUF of the Prover device is fed with a random challenge C_1 creating a response.

Further, a commitment containing the responses is created. To instantiate our system, we opted for Pedersen commitments [26] because of its simplistic and well-defined structure, it guarantees perfect hiding and supports efficient zero-knowledge proofs. However, every other perfectly hiding commitment scheme with existing zero-knowledge-proof systems could be used. The Pedersen commitment scheme consists of the following algorithms:

- $Com(m, r)$: Creates the commitment $COM = g^m \cdot h^r$ using our established protocol parameters where m represents committed value and r is a random blinding factor.
- $Ver(m, r, COM)$: Runs the verification of the commitment by checking if $Com(m, r) = COM$.

The above commitment scheme guarantees perfect hiding and computational binding and represents an implementation of the generic commitment scheme outlined in Section II.

To enrol a device, the stakeholder runs the following steps:

1) Select random enrollment challenge C_1 and $C_2 = H(C_1)$
2) Feed these challenges into the PUF of the device generating the resulting confidential responses R_1 and R_2.
3) Construct a Pedersen Commitment COM_P for Prover P, expressed as $Com(R_1, R_2) = g^{R_1} \cdot h^{R_2}$, where R_1 signifies the committed value, and R_2 serves as a random blinding factor.
4) The stakeholder then publicly records the commitment, possibly on a public bulletin board (e.g., a distributed ledger). The public storage address of the commitment COM_P further functions as the identifier ID_P of the Prover P.

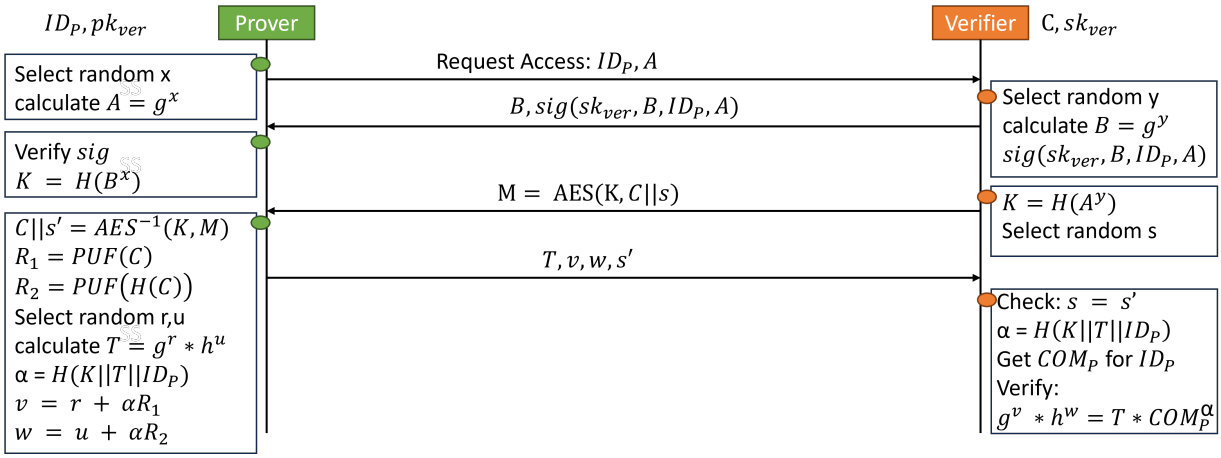

Fig. 2. Flow-diagram outlining the Authentication Phase of our PUF-based authentication scheme

This completes the enrollment phase, establishing a secure setting for device authentication. The device can now be deployed and transitioned into the authentication phase.

Authentication Phase: The authentication process displayed in Fig. 2 goes as follows.

1) The process is initiated by the Prover who selects a random number x and calculates $A = g^x$. Afterwards, it engages with the Verifier (e.g., to access a service) announcing its identity ID and the value A.
2) In response, the Verifier selects a random y, computes $B = g^y$, and signs B, A, and ID_P with sk_{ver} before sending it back to the Prover.
3) The Prover verifies the received signature to confirm that it indeed engaged with the correct Verifier.
4) Exchanging A and B enables Prover and Verifier to establish a shared secret K. Which is then used to confidentially transmit the challenge C together with a random string s from the Verifier to the Prover. The random string s is used to ensure the freshness of the authentication process.
5) The Prover decrypts the received challenge C and feeds it into its PUF generating responses R_1 and R_2. Afterwards, it picks two random values r and u and generates the Pedersen Commitment $T = Com(r, u)$. This is then fed into the public hash function H together with the ID of the Prover and the session key K to generate α.
6) The Prover runs Okamoto's Identification scheme [27] adopted for Pedersen Commitment to generate two zero-knowledge proofs computing v and w, where $v = r + cR_1$ and $w = u + cR_2$. These proofs enable the Prover to demonstrate knowledge of R_1 and R_2 to the Verifier, without disclosing the actual values of R_1 and R_2.
7) Finally, the value T together with the two zero-knowledge proofs v and w and the string s are replied to the Verifier.
8) The Verifier now checks the received values by applying the following steps:
   a) Verifier checks if the received s actually matches the initially sent s, ensuring freshness of the protocol run.
   b) Next, it calculates α by applying the public hash function H on T and K and the ID of the Prover, resulting in $H(K, T, ID_P) = \alpha$.
   c) It retrieves the public commitment COM_P from public storage (e.g., distributed ledger) using the ID_P of the Prover, as well as g and h, and verifies whether $g^v \cdot h^w = T \cdot COM_P^{\alpha}$.
9) If the equation holds, the Prover is authenticated; otherwise, the authentication process fails. After successful authentication, the authenticated session is terminated when the established connection between the Prover and Verifier is intentionally closed or inadvertently disrupted. Once the session is terminated, the Prover is required to rerun the authentication process.

Whenever the Prover correctly computes the proof, the honest Verifier always successfully verifies it, as demonstrated by:

$$\begin{aligned} g^v \cdot h^w &= g^{r+\alpha R_1} \cdot h^{u+\alpha R_2} = g^r \cdot g^{\alpha R_1} \cdot h^u \cdot h^{\alpha R_2} \\ &= g^r \cdot h^u \cdot g^{\alpha R_1} \cdot h^{\alpha R_2} = \\ &= T \cdot (g^{R_1} \cdot h^{R_2})^{\alpha} = T \cdot COM^{\alpha} \end{aligned}$$

Upon reviewing the submitted proof, the Verifier can ascertain whether the Prover genuinely holds the PUF response sealed in the commitment from the enrollment phase. This method verifies the prover's claim without disclosing the response, courtesy of the proof's zero-knowledge attribute. Thus, the PUF response's secrecy is preserved during and after authentication. Additionally, the challenge is kept concealed from potential eavesdroppers on the communication channel.

V. Implementation & Evaluation

To demonstrate the practicality of our proposed scheme, we carried out a reference implementation using the LPCXpresso55S69 Development Board, a resource-constrained IoT development platform running a 150MHz Arm Cortex-M with 320KB of RAM. Our implementation, utilizing MbedTLS [28], comprises roughly 500 lines of C code and supports both generating and verifying our authentication proof. Additionally, the enrollment process, typically executed on the stakeholder's side, was developed in Python with around 30 lines of code. The enrollment process was executed on a laptop powered by an AMD Ryzen 7-5825U processor.

Performance Evaluation. The evaluation used the SECP256R1 elliptic curve and groups of 1024-bit, 1536-bit, and 2048-bit sizes, with results shown in Table I.

Operation	ECC	ECC*	Bit-size		
			1024	1536	2048
Enrollment Phase	0.001	0.001	0.002	0.003	0.004
Authentication Proof	2.15	0.3	2.81	6.22	10.93
Authentication Verification	3.24	0.39	6.34	14.57	20.76

TABLE I
PERFORMANCE ANALYSIS IN SECONDS BASED ON OUR CONDUCTED PROTOTYPE IMPLEMENTATION (*UTILIZING HARDWARE ACCELERATION)

The performance evaluation reveals that enrollment procedures are remarkably fast, thus posing no bottleneck even when enrolling multiple devices. Our development board's ECC hardware acceleration significantly boosts speed, achieving nearly 10x faster performance compared to non-accelerated setups. Furthermore, our ECC implementation outpaces the 2048-bit group by 5 to 7 times, maintaining comparable security levels.

Storage Evaluation. The proposed scheme greatly reduces the Verifier's storage load, shifting from a confidential CRP database to just one challenge per Prover. This is possible because the Verifier only retains a single challenge per Prover, with commitments stored on public platforms like distributed ledgers. This approach is viable as confidentiality concerns are effectively addressed, negating the need for privately securing CRP pairs. This leads to a storage requirement of 68 bytes for the Prover, split into 35 bytes for ID_P and 33 bytes for pk_{ver}, and 48-64 bytes for the Verifier, comprised of 16-32 bytes for C (varying with the PUF) and 32 bytes for sk_{ver}.

Security Analysis. We provide informal security analysis by discussing potential attack vectors and their mitigation.

Eavesdropping: An adversary monitoring the communication channel may attempt to intercept confidential information, specifically the challenge or the response of the PUF. However, by encrypting the challenge, we prevent the adversary from obtaining any information about it. Additionally, the use of a zero-knowledge proof for authentication ensures that no information about the response is leaked, thereby preventing any attempts by the adversary to gain further insights.

Physical Attack: By keeping the challenge confidential, it prevents attackers from presenting the correct challenge to the PUF. Contrary to previous schemes [5]–[8], [29], where adversaries could intercept the challenge and use it to generate the reused response, our approach conceals the challenge. As a result, adversaries are compelled to resort to brute force methods to deduce the correct challenge — a task rendered infeasible due to the inherent complexity of typical PUF implementations. Furthermore, it's important to highlight that our adversary model does not encompass scenarios involving physical access by the attacker during the authentication process. This means that PUF is queried with the challenge and generates a response in the absence of the attacker.

Deauthentication / DoS: The adversary is restricted to passive network attacks, thereby incapable of conducting active attacks, like Deauthentication and Denial of Service.

VI. RELATED WORK

Research in PUF-based authentication can be divided into two primary categories: (i) the traditional approach that pre-generates a large quantity of CRPs, each for a single authentication session, and (ii) schemes that attempt to reuse challenges across multiple sessions. Strategies highlighted in works such as [30]–[36] adopt the traditional approach generating a comprehensive CRP database during the enrollment phase, using and then discarding a single challenge per authentication. Some schemes use the CRPs to train machine learning models for PUF response authentication [37], [38]. Nonetheless, both approaches require the Verifier to store a substantial amount of data, either as a CRP database or a machine learning model.

To overcome this, several proposed schemes such as [39]–[42] introduced a trusted third party that maintains confidential PUF information on behalf of the Verifier. Further schemes opted to introduce a trusted intermediary that mediates communication between parties [43]–[51]. These schemes rely on a trusted third party to authenticate a Prover on behalf of the Verifier or authenticate both parties and support them by establishing an authenticated channel between the devices. We classify them in the first category, as they adopt a similar approach but store large information via a trusted third party instead of directly on the Verifier.

In schemes like [5]–[8], [29], challenges are reused across sessions with PUFs generating asymmetric keys for signature-based authentication. These approaches do not protect against adversaries gaining temporary physical access. Further, they rely on witness hiding for PUF responses, presupposing security from perfectly random PUF outputs. However, analyses in [52]–[55] reveal that PUF responses lack full randomness, indicating a need for perfect hiding to ensure security. Conversely, schemes using public PUF-simulators for response verification [56], [57] are computationally demanding and make unrealistic assumptions about execution, making them impractical for IoT devices.

In summary, our solution enhances current methods with a secure, efficient PUF-based authentication system, enabling IoT devices to act as both Prover and Verifier independently of external trusted parties. It avoids unrealistic assumptions about storage and execution time. Additionally, we ensure perfect hiding of PUF responses and maintain security despite full physical access by adversaries.

VII. CONCLUSION

In conclusion, this paper presents a two-fold contribution. Firstly, we introduced a comprehensive methodology designed to analyze the decision-making process in selecting the most suitable type of PUF for various security contexts. This methodology provides a structured approach to determine the appropriateness of weak and strong PUFs based on the specific attacker model. Secondly, we proposed a novel PUF-based authentication scheme that protects against stronger adversaries than previous schemes that reuse challenges.

ACKNOWLEDGEMENTS

The research team from Würzburg has been funded by the European Union under Horizon Europe Programme - Grant Agreement 101070537 — CrossCon. The work of the third author is partially supported by the project "PARTHENON" - Research Programs of National Interest (PRIN) 2022 funded by the EU - Next Generation EU - CUP D53D23008610006.

REFERENCES

[1] N. Huynh, H. Cherian, and E. C. Ahn, "Hardware security of emerging non-volatile memory devices under imaging attacks," in *2021 International Conference on Applied Electronics (AE)*, pp. 1–4, IEEE, 2021.

[2] K. Shamsi and Y. Jin, "Security of emerging non-volatile memories: Attacks and defenses," in *2016 IEEE 34th VLSI Test Symposium (VTS)*, pp. 1–4, 2016.

[3] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 148–160, 2002.

[4] W. Che, V. K. Kajuluri, M. Martin, F. Saqib, and J. Plusquellic, "Analysis of entropy in a hardware-embedded delay puf," *Cryptography*, vol. 1, no. 1, p. 8, 2017.

[5] C. Felicetti, M. Lanuzza, A. Rullo, D. Saccà, and F. Crupi, "Exploiting silicon fingerprint for device authentication using cmos-puf and ecc," in *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 229–236, 2021.

[6] P. Tuyls and L. Batina, "Rfid-tags for anti-counterfeiting," in *Cryptographers' track at the RSA conference*, pp. 115–131, Springer, 2006.

[7] C. Felicetti, A. Guzzo, G. Manco, F. Pasqua, E. Ritacco, A. Rullo, and D. Saccà, "Deep learning/puf-based item identification for supply chain management in a distributed ledger framework," in *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*, pp. 28–35, IEEE, 2023.

[8] M. Shariq, K. Singh, M. Y. Bajuri, A. A. Pantelous, A. Ahmadian, and M. Salimi, "A secure and reliable rfid authentication protocol using digital schnorr cryptosystem for iot-enabled healthcare in covid-19 scenario," *Sustainable Cities and Society*, vol. 75, p. 103354, 2021.

[9] I. Damgård, "Commitment schemes and zero-knowledge protocols," in *Lectures on data security: modern Cryptology in Theory and Practice*, pp. 63–86, Springer, 2003.

[10] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pp. 203–225, 2019.

[11] D. Li, Y. Ren, D. Liu, Z. Guan, Q. Zhang, Y. Wang, and J. Liu, "Puf-based intellectual property protection for cnn model," in *International Conference on Knowledge Science, Engineering and Management*, pp. 722–733, Springer, 2022.

[12] D. Li, Y. Ren, D. Liu, Y. Guo, Z. Guan, and J. Liu, "Pipp: A practical puf-based intellectual property protection scheme for dnn model on fpga," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[13] A. ARM, "Security technology building a secure system using trustzone technology (white paper)," *ARM Limited*, 2009.

[14] J. Greene, "Intel® trusted execution technology: White paper. 2017," *URL: http://www. intel. com/content/www/us/en/architecture-and-technology/trusted-execution-technology/trusted-execution-technologysecurity-paper. html (visited on May 29, 2017)(cit. on p. 30)*.

[15] F. Armknecht, R. Maes, A.-R. Sadeghi, B. Sunar, and P. Tuyls, "Memory leakage-resilient encryption based on physically unclonable functions," *Towards Hardware-Intrinsic Security: Foundations and Practice*, pp. 135–164, 2010.

[16] E. Hozdić, "Smart factory for industry 4.0: A review," *International Journal of Modern Manufacturing Technologies*, vol. 7, no. 1, pp. 28–35, 2015.

[17] J. Delvaux, "Machine-learning attacks on polypufs, ob-pufs, rpufs, lhs-pufs, and puf–fsms," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2043–2058, 2019.

[18] L. Wu, Y. Hu, K. Zhang, W. Li, X. Xu, and W. Chang, "Flam-puf: A response–feedback-based lightweight anti-machine-learning-attack puf," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4433–4444, 2022.

[19] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose puf: Secure puf design against state-of-the-art machine learning attacks," *Cryptology ePrint Archive*, 2018.

[20] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "Puf modeling attacks on simulated and silicon data," *IEEE transactions on information forensics and security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[21] T. Kroeger, W. Cheng, J.-L. Danger, S. Guilley, and N. Karimi, "Cross-puf attacks: Targeting fpga implementation of arbiter-pufs," *Journal of Electronic Testing*, vol. 38, no. 3, pp. 261–277, 2022.

[22] U. Ruhrmair and J. Solter, "Puf modeling attacks: An introduction and overview," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2014.

[23] E. Rescorla, "Diffie-hellman key agreement method," tech. rep., 1999.

[24] M. Lochter and J. Merkle, "Elliptic curve cryptography (ecc) brainpool standard curves and curve generation," tech. rep., 2010.

[25] J. Bonneau, J. Clark, and S. Goldfeder, "On bitcoin as a public randomness source," *Cryptology ePrint Archive*, 2015.

[26] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*, pp. 129–140, Springer, 1991.

[27] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes," in *Annual international cryptology conference*, pp. 31–53, Springer, 1992.

[28] Arm, "Mbed tls."

[29] M. Delavar, S. Mirzakuchaki, M. H. Ameri, and J. Mohajeri, "Puf-based solutions for secure communications in advanced metering infrastructure (ami)," *International Journal of Communication Systems*, vol. 30, no. 9, p. e3195, 2017.

[30] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual design automation conference*, pp. 9–14, 2007.

[31] A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "Reverse fuzzy extractors: Enabling lightweight mutual authentication for puf-enabled rfids," in *Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers 16*, pp. 374–389, Springer, 2012.

[32] W. Che, F. Saqib, and J. Plusquellic, "Puf-based authentication," in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 337–344, IEEE, 2015.

[33] D. Liu, X. Liu, H. Zhang, H. Yu, W. Wang, L. Ma, J. Chen, and D. Li, "Research on end-to-end security authentication protocol of nb-iot for smart grid based on physical unclonable function," in *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, pp. 239–244, IEEE, 2019.

[34] U. Chatterjee, R. Sadhukhan, V. Govindan, D. Mukhopadhyay, R. S. Chakraborty, S. Pati, D. Mahata, and M. M. Prabhu, "Pufssl: an openssl extension for puf based authentication," in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, 2018.

[35] J. W. Byun, "An efficient multi-factor authenticated key exchange with physically unclonable function," in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4, IEEE, 2019.

[36] S. Suganthi, R. Anitha, V. Sureshkumar, S. Harish, and S. Agalya, "End to end light weight mutual authentication scheme in iot-based healthcare environment," *Journal of Reliable Intelligent Environments*, vol. 6, pp. 3–13, 2020.

[37] R. Pugliese, S. Regondi, and R. Marini, "Machine learning-based approach: Global trends, research directions, and regulatory standpoints," *Data Science and Management*, vol. 4, pp. 19–29, 2021.

[38] W. Liang, S. Xie, D. Zhang, X. Li, and K.-c. Li, "A mutual security authentication method for rfid-puf circuit based on deep learning," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 2, pp. 1–20, 2021.

[39] V. Clupek and V. Zeman, "Robust mutual authentication and secure transmission of information on low-cost devices using physical unclonable functions and hash functions," in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 100–103, IEEE, 2016.

[40] Y.-H. Chuang and C.-L. Lei, "Puf based authenticated key exchange protocol for iot without verifiers and explicit crps," *IEEE Access*, vol. 9, pp. 112733–112743, 2021.
[41] M. Barbareschi, A. De Benedictis, E. La Montagna, A. Mazzeo, and N. Mazzocca, "Puf-enabled authentication-as-a-service in fog-iot systems," in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 58–63, IEEE, 2019.
[42] Z. Huang and Q. Wang, "A puf-based unified identity verification framework for secure iot hardware via device authentication," *World Wide Web*, vol. 23, no. 2, pp. 1057–1088, 2020.
[43] S. Yoon, B. Kim, Y. Kang, and D. Choi, "Puf-based authentication scheme for iot devices," in *2020 international conference on information and communication technology convergence (ICTC)*, pp. 1792–1794, IEEE, 2020.
[44] K. Lounis and M. Zulkernine, "T2t-map: A puf-based thing-to-thing mutual authentication protocol for iot," *IEEE Access*, vol. 9, pp. 137384–137405, 2021.
[45] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A puf-based secure communication protocol for iot," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–25, 2017.
[46] A. Braeken, "Puf based authentication protocol for iot," *Symmetry*, vol. 10, no. 8, p. 352, 2018.
[47] M. A. Muhal, X. Luo, Z. Mahmood, and A. Ullah, "Physical unclonable function based authentication scheme for smart devices in internet of things," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 160–165, IEEE, 2018.
[48] J. Lee, J. Oh, D. Kwon, M. Kim, S. Yu, N.-S. Jho, and Y. Park, "Puftap-iot: Puf-based three-factor authentication protocol in iot environment focused on sensing devices," *Sensors*, vol. 22, no. 18, p. 7075, 2022.
[49] Q. Jiang, X. Zhang, N. Zhang, Y. Tian, X. Ma, and J. Ma, "Two-factor authentication protocol using physical unclonable function for iov," in *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 195–200, IEEE, 2019.
[50] M. N. Aman, K. C. Chua, and B. Sikdar, "Physically secure mutual authentication for iot," in *2017 IEEE Conference on Dependable and Secure Computing*, pp. 310–317, IEEE, 2017.
[51] G. S. Gaba, M. Hedabou, P. Kumar, A. Braeken, M. Liyanage, and M. Alazab, "Zero knowledge proofs based authenticated key agreement protocol for sustainable healthcare," *Sustainable Cities and Society*, vol. 80, p. 103766, 2022.
[52] W.-C. Wang, Y. Yona, S. N. Diggavi, and P. Gupta, "Design and analysis of stability-guaranteed pufs," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 978–992, 2018.
[53] B. B. Talukder, F. Ferdaus, and M. T. Rahman, "Memory-based pufs are vulnerable as well: A non-invasive attack against sram pufs," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4035–4049, 2021.
[54] S. Duan and G. Sai, "Bti aging-based physical cloning attack on sram puf and the countermeasure," *Analog Integrated Circuits and Signal Processing*, vol. 117, no. 1, pp. 45–55, 2023.
[55] A. Roelke and M. R. Stan, "Attacking an sram-based puf through wearout," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 206–211, IEEE, 2016.
[56] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding: 11th International Workshop, IH 2009, Darmstadt, Germany, June 8-10, 2009, Revised Selected Papers 11*, pp. 206–220, Springer, 2009.
[57] H. Hamadeh and A. Tyagi, "Privacy preserving data provenance model based on puf for secure internet of things," in *2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*, pp. 189–194, IEEE, 2019.