2025 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium

Lightweight Framework for PUF based IoT Authentication without Fuzzy Extractor

Franco Cirillo
University of Salerno
Fisciano (SA), Italy
fracirillo@unisa.it

Christian Esposito
University of Salerno
Fisciano (SA), Italy
esposito@unisa.it

Francesco Palmieri
University of Salerno
Fisciano (SA), Italy
fpalmieri@unisa.it

***Abstract*—The widespread adoption of Internet of Things (IoT) devices has underscored the critical importance of robust authentication mechanisms to protect against misuse and security breaches. Traditional authentication systems that rely on stored keys are vulnerable to theft, prompting a shift towards keyless authentication methods. Among these, Physical Unclonable Functions (PUFs) have emerged as a promising solution by leveraging the inherent physical uniqueness of each device to generate cryptographic keys dynamically. To address the inherent variability and noise in PUF responses, modern systems often employ Fuzzy Extractors, cryptographic tools designed to stabilize these responses. However, the integration of Fuzzy Extractors introduces notable challenges, including increased computational overhead, potential vulnerabilities in implementation, and the risk of exposing sensitive information, making such solutions less suitable for resource-constrained IoT devices.**

This study proposes a novel authentication framework that utilizes PUFs without relying on Fuzzy Extractors, thereby addressing the limitations of existing approaches. The proposed method enhances security by exploiting the unpredictability of PUF challenge-response pairs, ensuring that attackers face significant uncertainty in guessing the correct responses. Additionally, an implementation based on SRAM-PUF is presented as a practical use case of the framework. By eliminating the need for Fuzzy Extractors, the proposed framework mitigates risks associated with data leakage and modeling attacks while reducing computational complexity.

***Index Terms*—Physical Unclonable Function (PUF), SRAM-PUF, Device Authentication, Key Generation, IoT Security.**

I. Introduction

The Internet of Things (IoT) has transformed the landscape of digital applications, introducing a decentralized network of smart devices that communicate autonomously through wireless technologies without requiring human intervention [1]. However, this interconnected ecosystem brings significant security challenges, including concerns about privacy, authorization, access control, system integrity, and secure data storage [2]. Addressing these challenges is crucial to ensure a reliable and secure IoT environment.

Authentication plays an important role in establishing trust within IoT networks [3]. While traditional password-based methods are prone to vulnerabilities and insufficient for device authentication, cryptographic solutions provide robust security. Nonetheless, these approaches often require processing and storage resources that exceed the capabilities of many IoT devices. Given the constraints of IoT nodes, such as limited memory, processing power, and energy efficiency [4], research has focused on lightweight cryptographic methods, particularly symmetric ciphers optimized for resource-constrained environments. Despite their advantages, cryptography is not impervious to attacks, as it depends on securely storing and managing cryptographic keys, which must remain confidential, unique, and unpredictable.

To enhance security against unauthorized access and physical tampering, cryptographic keys are typically stored in secure hardware environments designed to resist leakage and attacks. An alternative approach to key storage is the use of Physically Unclonable Functions (PUFs) [5]. PUFs exploit inherent manufacturing variations to dynamically generate cryptographic keys without requiring them to be stored. These devices operate using a challenge-response mechanism, where a challenge (an input) is provided to the PUF, and a unique response is generated based on the device's physical characteristics. This variability ensures that responses are unpredictable and unique, making PUF-based systems both secure and tamper-resistant.

PUFs are categorized into various types—such as silicon-based, optical, and magnetic PUFs—depending on the physical properties they exploit. Among these, SRAM-based PUFs are particularly noteworthy due to their compatibility with widely used integrated circuits (ICs) that already incorporate SRAM technology, making them cost-effective and practical for large-scale deployment.

Modern implementations of PUFs often rely on cryptographic tools called Fuzzy Extractors [6]–[8] to mitigate noise and variability in PUF responses, ensuring consistent outputs for authentication. However, Fuzzy Extractors introduce several challenges, including susceptibility to side-channel attacks, risks of sensitive data leakage, and implementation vulnerabilities. Additionally, their computational overhead can be prohibitive, particularly in resource-constrained IoT devices.

This paper introduces a novel framework for lightweight authentication in IoT systems using PUFs that eliminates the need for Fuzzy Extractors. The main contributions are as follows:

- Framework Design: Development of an authentication mechanism based on PUFs for resource-constrained devices, designed to operate without Fuzzy Extractors.

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

- Implementation: Practical realization of the proposed framework using SRAM-PUF on real hardware to demonstrate feasibility and performance.
- Threat Model: Comprehensive analysis of potential risks, threats, and vulnerabilities affecting the proposed scheme.

The remainder of the paper is structured as follows: in Section II, we discuss PUF principles and implementations. Section V provides an overview of the related works in PUF technology, outlining recent advancements. In Section VI, we introduce our Proposed Framework, detailing its components including the Enrollment and Authentication phases, and analyzing its effectiveness. Section VII focuses on the implementation of SRAM-PUF, exploring technical aspects and practical considerations. Finally, in Section VIII, we conclude by resuming key findings and highlighting future research directions.

II. Physical Unclonable Functions

A. Introduction

In modern electronic systems, digital keys are typically stored in non-volatile memory (NVM). Ensuring the security of these keys throughout the lifespan of an embedded system is a significant challenge, particularly when keys are stored digitally in NVM without dedicated protection, as adding such protection increases costs. Moreover, keys stored in NVM are assigned externally and are not inherently tied to the physical characteristics of a device, making them susceptible to duplication.

An alternative approach is the use of PUFs, which are resistant to modification and cloning, cost-effective, and resilient against various physical attacks [5]. PUFs exploit the inherent random variations introduced during the manufacturing process of electronic devices to generate unique secret keys dynamically. Unlike externally assigned keys, PUF-generated keys are internal to the device and are intrinsically tied to its physical characteristics. These random variations act as a unique fingerprint for each device and do not affect the digital functionality of integrated circuits, making PUFs a highly cost-effective solution for secure key generation.

In this work, PUFs are employed as secure key generators to design an authentication framework aimed at enhancing the security of IoT devices and fortifying perimeter systems. Before delving into the details of the proposed approach, the primary features and operation of PUFs are summarized.

III. Features and Operation of PUFs

PUFs operate on the challenge-response paradigm, where a specific input, referred to as a challenge, is applied to the PUF, and the corresponding output, referred to as a response, is generated. The response is a bit string of fixed length and depends on the unique physical characteristics of the device. Unlike traditional mathematical functions, PUFs are influenced by nanoscale manufacturing imperfections, which make their behavior unpredictable and resistant to manipulation. This mapping between challenges and responses is unique to each PUF and is commonly referred to as the challenge-response pair (CRP) behavior.

The operation of PUFs can be divided into two distinct phases:

1) Enrollment Phase: A set of CRPs is collected from the PUF during this phase and stored securely in a CRP database.
2) Verification Phase: During authentication, a randomly selected challenge is applied to the PUF, and the generated response is compared with the corresponding response stored in the database.

PUFs have been implemented using a variety of technologies and materials, such as glass, plastic, paper, electronic components, and silicon ICs. The most common PUF designs include:

- Delay-based Silicon PUFs: These measure random variations in the delay of digital circuits, such as Arbiter-PUFs or Anderson-PUFs.
- Memory-based Silicon PUFs: These leverage random variations in bistable memory elements due to mismatches in device parameters.

IV. SRAM PUFs

A prominent example of memory-based PUFs is the SRAM PUF [9], which leverages the bistable nature of SRAM cells. Each SRAM cell has two stable states, representing binary digits (0 and 1). When powered on, an SRAM cell initializes into one of its stable states based on random variations in the cell's manufacturing process. These variations, introduced during production, determine the preferred state of the cell, making it unique to each device.

The underlying behavior of an SRAM cell can be explained using its voltage transfer characteristics. The CMOS (Complementary Metal-Oxide Semiconductor) inverter structure within the SRAM has three potential operating points: two stable states and one metastable state. Due to the influence of random noise, an SRAM cell rarely remains in its metastable state and quickly transitions to one of the two stable states. This behavior is driven by positive feedback, which amplifies any small deviations from the metastable state, forcing the circuit toward stability.

The random process variations in manufacturing determine the initial operating state of each SRAM cell, creating a unique and reproducible bit pattern when the SRAM is powered on. This property makes SRAM PUFs an ideal choice for secure key generation and authentication, as they do not require additional hardware and are naturally present in many integrated circuits.

V. State of the Art

The field of PUF-based authentication mechanisms has seen numerous advancements in recent years, with proposals aimed at overcoming existing challenges. However, significant gaps remain, and a universally robust implementation has yet to be realized.

The work in [10] introduces a novel two-factor authentication protocol for IoT devices that combines PUFs and radio frequency fingerprints to uniquely identify devices. While the authors employ Bose–Chaudhuri–Hocquenghem (BCH) codes for error correction, their approach is hindered by challenges such as high complexity and significant overhead. Paper [11] focuses on secure broadcast authentication schemes for smart meters in smart grid networks, utilizing a ring oscillator PUF. While the protocol demonstrates resilience to physical attacks, it imposes increased memory and computational requirements on resource-constrained smart meters, which limits its applicability in large-scale deployments. Similarly, [12] proposes a privacy-aware authentication and key agreement scheme for secure smart grid communication. This scheme incorporates PUFs, one-way hash functions, and BCH-based fuzzy extractors for error correction. However, storing public helper data and other information in device memory makes the protocol vulnerable to physical attacks and helper data manipulation.

In [13], a PUF-based mutual authentication protocol is proposed for unmanned aerial vehicles (UAVs) and ground stations. Although innovative, the protocol does not include error correction mechanisms, making it unsuitable for noisy environments. A similar limitation exists in [14], which proposes a PUF-based authentication protocol for IoT devices. To counter threats like modeling and man-in-the-middle attacks, [15] proposes a lightweight PUF-based protocol for mutual authentication between IoT devices and a trusted server. Despite its computational efficiency, the protocol fails to address practical PUF reliability issues and device anonymity, leaving room for improvement. A unique two-way PUF-based authentication scheme for smart grid communications is presented in [16]. This protocol eliminates the need for CRP storage on a trusted server by retaining only a single CRP. However, the reliance on smart meters to store secret information contradicts fundamental PUF principles and exposes the devices to physical attacks. The work in [17] presents a PUF-based authentication protocol for the Internet of Medical Things (IoMT). The protocol employs a secure database to store CRPs for IoMT devices, with both devices and servers equipped with PUF circuits. However, the scheme lacks error correction mechanisms and reliability guarantees, making it vulnerable to modeling attacks.

Limitations such as high computational and memory overhead, vulnerabilities to attack and lack of error correction mechanisms for noisy environments, underscore the need for a robust, lightweight, and secure PUF-based authentication scheme that can address these challenges while ensuring practical applicability.

VI. Proposed Framework

This paper introduces a flexible framework for device authentication that serves as a high-level methodology to secure interactions between devices. Furthermore, the framework allows for the integration of different PUF mechanisms that operate on the challenge-response paradigm, fostering innovation and scalability in the development of authentication solutions.

TABLE I
Enrollment and authentication symbols notation

N	CRPs database size
c	PUF challenge
C	PUF challenges set
r	PUF response
n	The number of challenges for each execution
b	Variable for selection
fHD	Fractional hamming distance
k	Key derived from PUF responses
Hash	Hash function
nonce	Random generated nonce
Enc	Encryption algorithm
$k_{session}$	Pseudo Random Key for session
m	Ciphered message

A distinguishing feature of the proposed framework is the elimination of fuzzy extractors in the PUF-based authentication process. The absence of a fuzzy extractor offers several key advantages. First, it simplifies the design and implementation of the system by reducing the overall complexity of the cryptographic operations involved. This simplicity makes the framework easier to integrate into resource-constrained environments. Second, the exclusion of the fuzzy extractor enhances efficiency by eliminating additional processing steps, thereby reducing computational overhead and conserving power. Third, removing this component mitigates vulnerabilities introduced by fuzzy extractors, such as susceptibility to side-channel attacks or other exploits targeting helper data. Lastly, dispensing with the fuzzy extractor reduces hardware and software development costs, contributing to a more cost-effective solution for PUF-based systems.

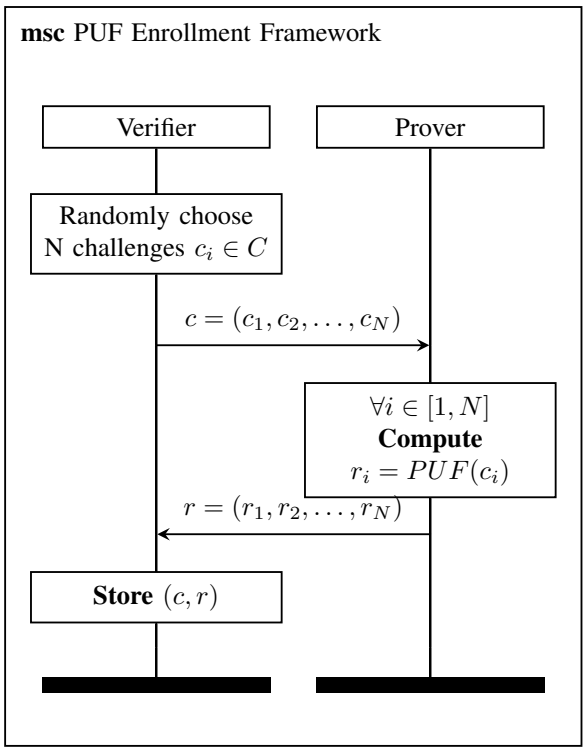
The authentication process within the framework is structured into two distinct operations. The first is the enrollment phase, during which essential key information is stored securely. The second is the authentication phase, where the stored data is utilized to verify the authenticity of a device.

Table I provides an overview of the symbols used in the diagrams.

A. Enrollment phase

The following Message Sequence Chart illustrated depicts the enrollment phase of the proposed framework. The enrollment protocol begins with the verifier, which selects N challenges from the challenge space C, where N represents the size of the CRPs database. These challenges are denoted as c_i for $i = 1, 2, \ldots, N$. The verifier transmits these challenges to the prover for processing. Upon receiving the challenges, the prover utilizes the PUF at its disposal to compute a response r_i for each challenge c_i. This results in a sequence of responses $r = (r_1, r_2, \ldots, r_N)$. Once the responses are generated, the prover sends the computed responses back to the verifier. It then securely stores the N challenge-response pairs (c_i, r_i) in the CRPs database. These pairs will be used later in the authentication phase to verify the identity of the prover. It is crucial to emphasize that the enrollment phase

must be conducted in a secure environment to maintain the confidentiality of the transmitted data, particularly the CRPs.



B. Authentication phase

The authentication phase of the proposed PUF-based framework is depicted in the following Message Sequence Chart. This phase assumes that the device has already completed the enrollment phase, and the verifier has stored a secure database of CRPs. A key aspect of the authentication process is the use of the fractional Hamming distance (fHD), a metric designed to measure the level of dissimilarity between two binary strings or vectors. In this scheme, the fHD is employed to calculate the distance between two PUF responses. Let r_k denote the k-th bit of a response, and let K represent the total bit-length of the PUF responses. The fractional Hamming distance, denoted as fHD, is mathematically defined as follows:

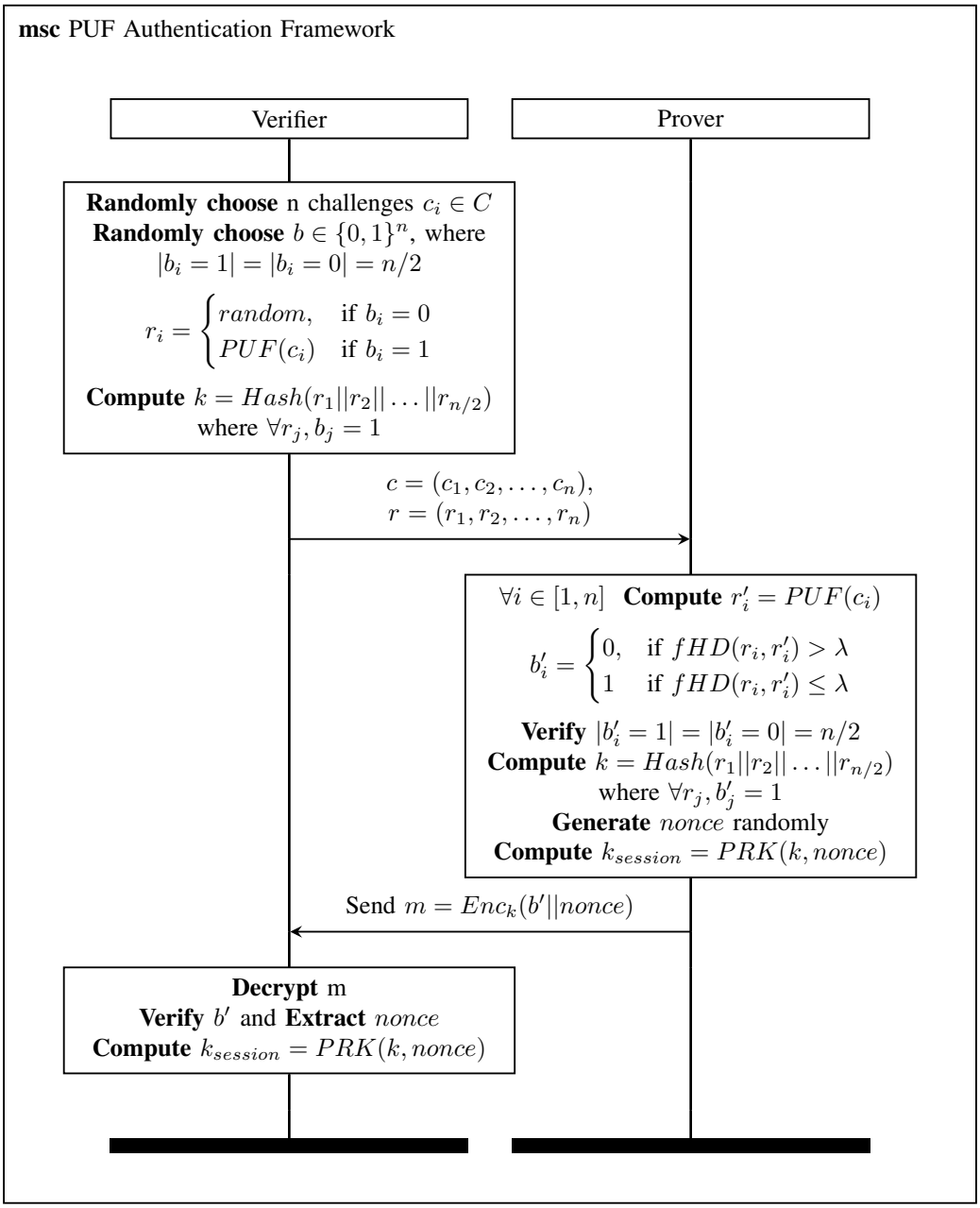
$$\text{fHD}\,(r_k,\, r_h) = \frac{1}{K} \sum_{n=0}^{K-1} r_k \oplus r_h \tag{1}$$

The authentication protocol begins with the verifier randomly selecting n challenges, denoted as c_i, from the challenge space C. A binary vector b of length n is then generated, with half of its elements set to 0 and the other half set to 1. For each challenge c_i, if the corresponding bit in b is 0, a random response r_i is assigned. If the bit is 1, the response is retrieved from the Challenge-Response Pair (CRP) database, corresponding to the PUF's response to the specific challenge. The verifier then constructs a key by hashing the concatenation of only those responses where the corresponding bit in b is 1. These challenges c and their corresponding responses r are transmitted to the prover. Upon receiving the challenges and responses, the prover generates noisy responses r_i' for each challenge c_i using the PUF function and compares them with the received responses r_i to form a new binary vector b'. This comparison is made using the fHD (fractional Hamming distance) formula, which measures whether the distance between responses exceeds a predefined error threshold λ. If the distance exceeds the threshold, the response is considered random, and the corresponding bit in b' is set to 0. If the response matches the CRP database, the bit is set to 1. The prover ensures that the number of 0s and 1s in b' are equal; if not, the process is halted. Next, the prover constructs a key by hashing the concatenation of responses where the corresponding bit in b' is 1. These responses are assumed to be from the CRP database, while the others are random. The prover then generates a session key k_{session} using a Pseudo Random Key (PRK) generation method, based on the primary key and a newly generated nonce. The original key is used to encrypt the binary vector b' along with the nonce, and this encrypted message is sent back to the verifier. The verifier decrypts the message, verifies the binary vector b', and extracts the nonce. Finally, the verifier computes the session key k_{session} using the same method as the prover. This ensures that both the verifier and prover share the same session key, establishing authentication and securing the communication channel by utilizing the properties of PUFs to recover matching responses, aside from noise-induced errors.

C. Analysis of the proposed framework

To fully appreciate the effectiveness of the proposed framework, it is important to clarify several key points. The primary goal of the scheme is to eliminate the need for a Fuzzy Extractor, thereby avoiding the use of ECCs that require storing helper data. Storing such data could introduce a security risk, as an attacker could potentially exploit a theft of this information to gain insight into the PUF responses. In the framework described, the prover does not retain any data after the enrollment phase and only accesses the PUF during authentication queries. Moreover, the security of the protocol relies on the assumption that an attacker, even if they intercept CRPs, cannot distinguish between authentic and random responses. This is because half of the responses are randomly generated, making it difficult to deduce the authenticity of any given response. It is crucial that the PUF used exhibits high quality, including characteristics such as entropy, uniqueness, instability, bit-aliasing, and uniformity, to ensure the protocol's security. Additionally, it is assumed that each challenge is used only once; otherwise, an attacker could capture the response for each challenge and later exploit this data to identify authentic responses.

The prover only needs to determine whether each response is authentic or random. After making this determination for all responses, and verifying that the number of random responses is approximately 50%, the prover can securely derive a key by hashing the valid responses. Since both the verifier and the prover apply the same hash function to the same responses, they will arrive at the same key, eliminating the need for



error correction for noisy responses. Furthermore, the hash function ensures that the responses are independent of the resulting key. To protect against replay attacks, a session key is created using a random nonce, which the prover sends to the verifier along with the vector b' in an encrypted message. Verifying b' allows the verifier to confirm that the authentication was successful before engaging in any further communication using the session key. An adversary might try to guess the correct combination of responses in order to derive the key by hashing their guesses. However, this would require testing all possible combinations of valid responses, which is computationally infeasible. The number of such combinations is given by the binomial coefficient: $|K| = \binom{n}{n/2} = \frac{n!}{\frac{n}{2}!\frac{n}{2}!}$ For an appropriately chosen parameter n, the operation becomes computationally impractical. Nonetheless, it is important to note that there is still a nonzero probability of error, even if the prover is authentic. A randomly generated response may accidentally resemble the response to a particular challenge. The probability of this occurring is given by: $p_{error} = \frac{1}{2^K}\lambda n$ where λ is the probability of similarity between a random response and a valid PUF response, and n is the number of challenges used in the authentication process. Naturally, as the length of the responses increases, this error probability decreases, making the system more secure.

VII. SRAM-PUF Implementation

In this section, we demonstrate the application of the proposed framework through an implementation in a real-world

scenario using an SRAM-PUF. The case study is conducted on the STM32 Nucleo F401RE, which is equipped with 96 KB of SRAM memory.

In the system setup, challenges are represented by the addresses of consecutive 10-bit segments within the SRAM, and the corresponding responses are the values read from these 10-bit segments. We define $L = 7000$ bits as the SRAM memory area that remains uninitialized and is utilized as the PUF. While the memory of the STM32 device is somewhat limited, using a more memory-capable device could yield better performance. During the enrollment phase, the verifier selects $L/10$ challenges, each corresponding to a tuple of 10 bits. For example, r_0 corresponds to the first 10 bits of the uninitialized SRAM area, and similarly for subsequent responses. The verifier, which acts as a simple server in this case study (implemented on a basic computer), stores the CRPs in its database. In the authentication phase, the verifier randomly selects $n = 140$ challenges and computes 70 PUF responses, which corresponds to 700 bits. The hash function used to derive the session key is SHA256, the symmetric encryption algorithm is AES, and the key derivation process uses HMAC-based Key Derivation Function. The parameters selected for the implementation allow for the generation of 2^{700} possible response sequences. Given that the responses can either be authentic or random, and maintaining the 50% condition, the total number of possible combinations is approximately $\binom{140}{70} \approx 2^{129}$. This suggests that the difficulty for an attacker to guess the correct key using brute-force methods is roughly equivalent to the complexity of guessing a 128-bit key. The parameter for response similarity is set to $\lambda = 0.2$, meaning that up to 2 bits of error are allowed for each response.

VIII. Conclusion and Future Work

The growing prevalence of IoT devices in critical sectors highlights the need for robust authentication mechanisms to safeguard against misuse and security breaches. Traditional authentication methods, which rely on stored keys, are inherently vulnerable to theft, making it essential to explore alternative approaches, such as those based on PUFs. While current PUF implementations often use Fuzzy Extractors to mitigate noise and variability, these solutions introduce challenges related to potential data leakage and increased computational overhead. This study introduces an innovative framework for developing authentication mechanisms based on PUFs, specifically designed for resource-constrained devices, which eliminates the need for Fuzzy Extractors. Furthermore, the proposed implementation utilizes SRAM-PUF, achieving a level of attacker complexity comparable to guessing a 128-bit key. Future research could focus on the formal analysis of these protocols and a deeper examination of potential attacks.

Acknowledgment

This work was supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] A. Khanna and S. Kaur, "Internet of things (iot), applications and challenges: a comprehensive review," *Wireless Personal Communications*, vol. 114, pp. 1687–1762, 2020.

[2] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, "Cyber threats to industrial iot: a survey on attacks and countermeasures," *IoT*, vol. 2, no. 1, pp. 163–186, 2021.

[3] N. H. Kamarudin, N. H. S. Suhaimi, F. A. Nor Rashid, M. N. A. Khalid, and F. Mohd Ali, "Exploring authentication paradigms in the internet of things: A comprehensive scoping review," *Symmetry*, vol. 16, no. 2, p. 171, 2024.

[4] A. Haroon, M. A. Shah, Y. Asim, W. Naeem, M. Kamran, and Q. Javaid, "Constraints in the iot: the world in 2020 and beyond," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

[5] A. Al-Meer and S. Al-Kuwari, "Physical unclonable functions (puf) for iot devices," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–31, 2023.

[6] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M.-D. Yu, "Efficient fuzzy extraction of puf-induced secrets: Theory and applications," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 412–431.

[7] M. Taniguchi, M. Shiozaki, H. Kubo, and T. Fujino, "A stable key generation from puf responses with a fuzzy extractor for cryptographic authentications," in *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*. IEEE, 2013, pp. 525–527.

[8] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, and K. Iwamura, "Cryptographie key generation from puf data using efficient fuzzy extractors," in *16th International conference on advanced communication technology*. IEEE, 2014, pp. 23–26.

[9] G.-J. Schrijen and V. Van Der Leest, "Comparative analysis of sram memories used as puf primitives," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2012, pp. 1319–1324.

[10] M. Alkanhal, M. Younis, A. Alali, and S. S. Mehjabin, "Ferha: Fuzzy-extractor-based rf and hardware fingerprinting two-factor authentication," *Applied Sciences*, vol. 14, no. 8, p. 3363, 2024.

[11] M. H. Ameri, M. Delavar, and J. Mohajeri, "Provably secure and efficient puf-based broadcast authentication schemes for smart grid applications," *International Journal of Communication Systems*, vol. 32, no. 8, p. e3935, 2019.

[12] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2018.

[13] C. Pu and Y. Li, "Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system," in *2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN*. IEEE, 2020, pp. 1–6.

[14] B. Kim, S. Yoon, Y. Kang, and D. Choi, "Puf based iot device authentication scheme," in *2019 international conference on information and communication technology convergence (ICTC)*. IEEE, 2019, pp. 1460–1462.

[15] T. A. Idriss, H. A. Idriss, and M. A. Bayoumi, "A lightweight puf-based authentication protocol using secret pattern recognition for constrained iot devices," *IEEE Access*, vol. 9, pp. 80 546–80 558, 2021.

[16] M. Kaveh, S. Aghapour, D. Martin, and M. R. Mosavi, "A secure lightweight signcryption scheme for smart grid communications using reliable physically unclonable function," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2020, pp. 1–6.

[17] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, "Pmsec: Physical unclonable function-based robust and lightweight authentication in the internet of medical things," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 388–397, 2019.