Securing the Internet of Medical Things using PUF-based SSI Authentication

Mario Barbareschi[1], Biagio Boi[2], Franco Cirillo[2,*], Marco De Santis[2] and Christian Esposito[2]

[1]*University of Naples Federico II, Via Claudio 21, Napoli, 80125, Italy*

[2]*University of Salerno, Via G. Paolo II 132, Fisciano, 84084, Italy*

Abstract

The proliferation of Internet of Things (IoT) technologies has revolutionized the medical domain, offering enhanced reliability and efficiency. However, the inherent vulnerability of these devices, often equipped with inadequate authentication mechanisms, poses significant threats to user privacy. To address these challenges, user-centric authentication leveraging asymmetric encryption has emerged as a valuable security measure across various domains. In this manuscript, we introduce a novel approach to authentication within the domain of the Internet of Medical Things (IoMT), capitalizing on the integration of Self-Sovereign Identity (SSI) and SRAM-based Physical Unclonable Function (PUF). Our approach stands out for its efficacy in resource-constrained environments, ensuring robust security without compromising operational efficiency. We present a concrete implementation under the SSI paradigm, showcasing a convenient time complexity, indicating its efficiency and suitability for practical use.

Keywords

Internet of Things, Device Authentication, Physically Unclonable Function, Self-Sovereign Identity

1. Introduction

In recent years, the expansion of connected devices has changed the entire healthcare context, introducing new opportunities in terms of remote patient monitoring, real-time diagnosis, and personalized treatment. The ensemble of devices used in the healthcare domain created the so-called Internet of Medical Things (IoMT), which range from wearable sensors to implantable devices, collectively aimed at improving patient care and healthcare outcomes [1]. However, these connections introduce new threats in terms of security and privacy for patients, such as data branches or attacks aimed at disrupting the medical device itself [2]. These devices are usually really small causing huge challenges when implementing traditional security mechanisms causing insufficient protection against attacks. Physical Unclonable Functions (PUFs) have emerged as a solution for enhancing the security of embedded systems. Within the field, there are multiple types of PUFs, with Silicon PUFs being the most relevant for the IoT context. These special kinds of PUFs leverage small hardware variations introduced by manufacturers [3]. PUFs

ITASEC '24: The Italian Conference on CyberSecurity, April 08–11, 2024, Salerno, IT

*Corresponding author.

mario.barbareschi@unina.it (M. Barbareschi); bboi@unisa.it (B. Boi); fracirillo@unisa.it (F. Cirillo); mdesantis@unisa.it (M. De Santis); esposito@unisa.it (C. Esposito)

0000-0002-1417-6328 (M. Barbareschi); 0000-0003-3044-5345 (B. Boi); 0009-0006-9599-5996 (F. Cirillo); 0009-0004-6514-4168 (M. De Santis); 0000-0002-0085-0748 (C. Esposito)

can be used in multiple security applications, such as key derivation, encryption, authentication, and so on. Their security relies on the uniqueness and unpredictability of the generated key. However, while PUF encryption addresses device-level security concerns, the broader landscape of IoT security necessitates a well-established authentication schema. User-centric authentication and Self-Sovereign Identity (SSI) have been demonstrated to be a valuable alternative to classical user authentication within various domains. This concept can be easily adapted for device authentication within the IoT context, decreasing the possible security attacks associated with centralized Identity Management (IdM) systems [4]. These approaches leverage asymmetric encryption in order to enhance security. Despite clear advantages introduced by device-centric authentication, the implementation of SSI within the IoT context poses multiple challenges due to the complexity of asymmetrical encryption algorithms taking into account the limitations of devices used [5]. In this manuscript, we want to exploit PUFs as a means of generating strong keys to use in device authentication. The key contributions of the current manuscript are listed below:

- Introduction of a novel authentication schema predicated upon the response characteristics of Physical Unclonable Functions (PUFs). This schema is uniquely capable of ensuring both identification and encryption functionalities.
- Application of the aforementioned schema towards the development of a lightweight and power-efficient solution tailored specifically for IoT-based Self-Sovereign Identity (SSI) wallets. This approach capitalizes on the inherent strengths of PUF-based authentication to address the constraints of resource-limited IoT environments.
- Comprehensive evaluation of the proposed user-centric authentication mechanism via empirical experimentation conducted on an experimental device. This evaluation encompasses a thorough analysis of performance metrics, thereby providing empirical validation of the feasibility and efficacy of the proposed approach in practical settings.

2. Background

2.1. Physical Unclonable Functions

PUFs are digital circuits that operate on the challenge-response paradigm, where challenges and responses are bitstrings of a given length. The uniqueness of the mapping function is inherent, and its resistance to manipulation arises from exploiting nanoscale imperfections introduced during the circuit manufacturing process. Furthermore, the unpredictability of responses is maintained, as neither physical imperfections can be controlled nor successful cloning of devices is feasible. PUFs can be categorized as memory-based PUFs or delay-based PUFs. Delay-based PUFs determine responses by measuring differences in signal propagation time over symmetric paths, exemplified by Arbiter-PUF or Anderson-PUF. In contrast, Memory-based PUFs, such as those utilizing SRAM (as implemented in our proposed approach), generate responses by exploiting nanoscale imperfections in symmetric memory cells.

IoT devices designed for medical applications often utilize SRAM as a common memory component. SRAM are a low cost memory technology and its inherent properties, such as variability and low-latency access, make it well-suited for medical IoT devices. One innovative application of SRAM in these devices is its utilization in conjunction with PUFs.

As SRAM-PUF cannot directly serve as a seed generator due to unstable memory cell responses, the Fuzzy Extractor (FE) technique [6, 7] is employed to reconstruct PUF responses, ensuring uniformity and randomness. This mechanism comprises two phases: an enrollment phase, where helper data is calculated for later use in the reconstruction of noisy PUF material and key generation; and a subsequent phase where the key is generated as needed. Implementing the FE scheme involves utilizing Error Correction Codes (ECC) and a hash function to derive the seed.

While PUF implementations often rely on custom circuits integrated into specialized hardware platforms like Field Programmable Gate Arrays (FPGA) [8], our work focuses on exploiting PUFs in non-custom circuits. Among low-cost Integrated Circuit (IC) that utilize SRAM, ATMEGA328P is particularly popular. Developing a mechanism to generate asymmetric keys based on ATMEGA328P PUF and utilizing them via SSI presents a promising prototype solution for enhancing the security of the Internet of Medical Things.

2.2. Self Sovereign Identity (SSI)

The concept of SSI represents a significant evolution in the area of digital identity management, which can be applied in the context of IoT. This approach puts individuals in complete and autonomous control of their digital identities at the center, transforming IoT devices into entities capable of interacting directly with end users in a secure and private manner.

In the SSI ecosystem, one of the key elements are decentralized identifiers (DIDs), which provide a unique and persistent way to identify digital entities such as people, organizations and devices. Each DID is composed of three parts: the DID URI scheme identifier, the identifier for the DID method, and the DID method-specific identifier. DIDs allow agents to maintain complete control over their digital identities without depending on central authorities.

Another crucial aspect of SSI are wallet credentials, which allow users to securely store their identity information and selectively and securely share it with third parties. Using encryption and digital signatures, wallet credentials ensure the authenticity and integrity of information, allowing agents to present only information relevant to a given interaction. So, the application of SSI in the IoT introduces the concept of *device user-centric* turning devices into entities that can manage their own digital identities and interact with other devices and services in a transparent and secure manner. This approach fosters greater privacy, security and interoperability in the IoT context, allowing agents to retain complete control over their personal information and decide how and when to share it.

3. Security Challenges of the Internet of Things

IoT devices usually suffer from memory and power constraints, making security a critical concern and posing limitations to the development of SSI within this context. Widely known asymmetric key-based solutions such as RSA cannot be considered as part of lightweight cryptography [9] due to its large key size, resulting from the use of two large prime numbers and modulo operations, which enhances security and preserves user privacy. In contrast, Elliptic Curve Cryptography (ECC) [10] necessitates a smaller key size compared to RSA, facilitating

Table 1
Related Works implementing IoT Device decentralized authentication

Approach	PUF	DLT	Authentication	Identification	Encryption
Luecking et al. [16]	✘	✓	✘	✓	✘
BlockPro [17]	✓	✓	✘	✓	✘
KYoT [18]	✓	✓	✘	✓	✘
Our Approach	✓	✓	✓	✓	✓

faster processing and requiring less memory. This characteristic makes ECC well-suited for hardware implementations, enabling quicker real-time computations. However, encrypting all data with an asymmetric algorithm could be very expensive. A useful method, as described in [9], is to use a hybrid algorithm, a combination of lightweight symmetric and asymmetric encryption, providing confidentiality, and integrity with small key size and less computation power as well as requiring less memory space. However, physical attacks in the IoT domain include node capture, side-channel attacks, node tampering, physical damage, and others, making the storage of cryptographic keys a serious concern [11]. PUFs have emerged as a secure alternative technique for generating keys without storing them, making them ideal for secure key management [12]. In addition, PUF can also be exploited to obtain randomness [13]. The authors of [14] proposed a symmetric and asymmetric encryption scheme based on ElGamal encryption scheme utilizing a PUF module. This approach aims to minimize implementation requirements and operational resource consumption while simplifying the overall key management process. Despite providing an intuitive approach to key generation and management, no analysis of resource consumption was carried out to assess the actual lightweight of the algorithms used. A fully decentralized approach has been presented in [15] which discusses a privacy-assuring authentication protocol, utilizing blockchain, PUF, and Ethereum-powered smart contracts to ensure security and prevent various attacks, utilizing lightweight cryptography primitives instead of traditional public-key cryptography. Authors do not considered time needed for the generation of key pair, by focusing only on the security of the proposed approach. Implementing decentralized identity solutions within the IoT context tries to solve multiple challenges, such as interoperability, scalability, and trust. On the same concept of decentralized identity management, the usage of SSI-based solutions can be leveraged to enhance the quality of authentication, as well as credentials management in general. In [16] a novel approach based on Distributed Ledger Technology (DLT) has been discussed in conjunction with SSI. This approach define the trust triangle and identification of IoT devices, also in this case without a clear evaluation of performance needed for the key generation. Moreover, the manuscript only propose an identification mechanism, instead of a complete solution, highlighting limitations of the IoT domain. Although benefits are introduced by SSI, concrete implementation typically suffers from various challenges typical of the IoT context [5]. Among these, asymmetric encryption is the major concern; limiting SSI expansion in this context. In [17], authors investigated the role of PUF in identity management systems, by leveraging DLT as trust anchor. Despite providing good results and efficacy, the manuscript lack a clear evaluation of time and performance. Research in [18] continued the discussion of decentralized identity with a concrete implementation of SSI and all the related concepts such as DID. The manuscript proposes a novel approach in identity management without providing a complete solution for both encryption and authentication. As outlined in prior work, according to Table 1, there is a gap in existing literature regarding a complete framework that leverages PUFs for establishing the necessary operations in SSI-based identity management systems. Our objective is to address these limitations by presenting a tangible implementation of these operations and conducting an evaluation within the context of the Internet of Medical Things (IoMT).

4. Proposed Authentication Scheme

SSI leverages the cryptographic wallet to store and retrieve VCs. Each wallet is uniquely associated with a DID which is an identifier for the proposed context. When turning this concept into constrained environments and when considering an IoT device as a platform for holding this wallet, multiple challenges arise. In this paper, an alternative solution is proposed for managing wallet in SSI ecosystem, which is based on asymmetric keys by leveraging the intrinsic characteristics of IoT devices. Specifically, PUFs are utilized, which exploit inherent variations in production to generate a unique and stable seed for each device, enabling the generation of a private key and subsequently the derivation of a public key. Furthermore, the instability of the SRAM is leveraged to generate derived temporary keys for each session. Considering the vulnerability of IoT nodes to capture and hacking, which may result in the potential exposure of stored secrets, the utilization of a PUF becomes essential in addressing this risk. This system is developed using the SRAM PUF in microcontrollers with limited resources, focusing on the ATMEGA328P chip as a case study for generating reliable PUF responses.

4.1. Overall Architecture

The architecture of the proposed schema is composed of an ATMEGA328P microcontroller, connected with an ESP-32 representing the IoT node. The ATMEGA328P is used to generate a PUF to create asymmetric keys. Such keys can be used for multiple purpose, in the next section we will see how it will be leveraged for the generation of a cryptographic wallet and associated DID. In this section we limit the discussion to the operation which it is possible to perform on this key in order to create a secure authentication schema. Considering the constrained environment, we will first exploit a lightweight technique for generating key pairs starting from key material offered by PUF; and then we leverage key exchange to produce a symmetric session key able to secure the communication between the device and external parties. In what follows we start the discussion from the key generation approach, then we discuss of session key algorithm and data encryption methods.

4.2. Key Generation

The key generation process of the proposed system is composed of a phase where a key is extracted from the PUF, and a second phase where this key is used for the generation of an asymmetric key pair.

The mechanism is depicted in Figure 1. In the primary phase it is necessary for the private key (K_{priv}) to remain the same, so the challenge must be unique to obtain the same response.

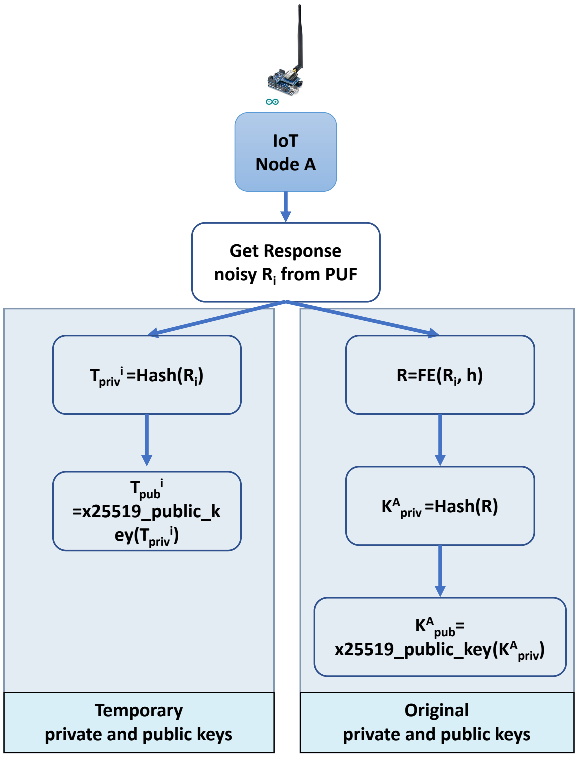

Figure 1: Generation of original private and public keys from reconstructed response and temporary keys from unstable responses.

This becomes a special case of PUF, known as single-challenge PUF, which can also be called Physically Obfuscated Key (POK) and represents a method for storing keys within an IC without actually storing them, thereby achieving greater resistance to device hacking. The challenge is represented by the need of reading a block of SRAM, which must be large enough to correctly perform the FE to reconstruct the same response every time and ensure security, but at the same time, it must be reduced as much as possible to decrease the resources used by the microcontroller. Based on various tests conducted, it was decided to use a key material of 304 bytes on 2 KB of ATMEGA328P SRAM, to be kept intact during the execution of the extractor algorithm. The response is represented by the effective reading of this block, which is then reconstructed by the FE. The reconstructed response is then hashed to create a K_{priv} of a fixed length 32 byte to be used in the asymmetric keys generator. Moreover, for the generation of a session key, we also use the same PUF to create a temporary private key (T_{priv}), which is inherently random due to the instability of SRAM. In this case, the PUF responses and SRAM block readings are not reconstructed; rather, they are only hashed to obtain a uniformly temporary secret key for use in the asymmetric key generator.

For the creation of asymmetric keys, the Monocypher library [19] has been used. Monocypher is a lightweight cryptography library designed to be portable, highly suitable for constrained environments and low-power microcontrollers, thanks to its minimal footprint and low processing demands. Deployment is straightforward; users simply need to add two files to their

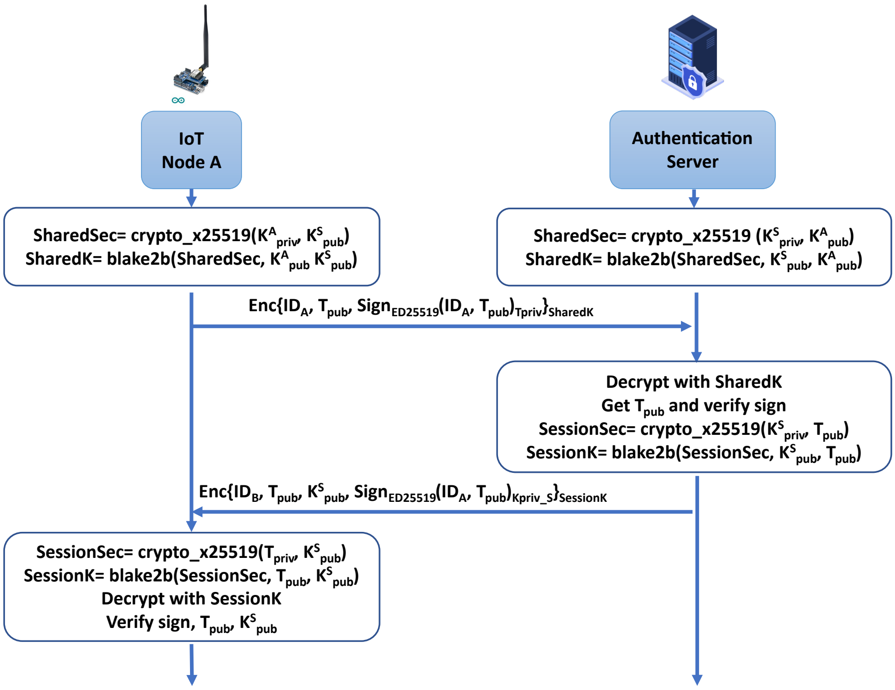
IoT Node A
Authentication Server
SharedSec= crypto_x25519(K^A_{priv}, K^S_{pub})
SharedK= blake2b(SharedSec, K^A_{pub} K^S_{pub})
SharedSec= crypto_x25519 (K^S_{priv}, K^A_{pub})
SharedK= blake2b(SharedSec, K^S_{pub}, K^A_{pub})
Enc{ID_A, T_{pub}, $Sign_{ED25519}$(ID_A, T_{pub})$_{Tpriv}$}$_{SharedK}$
Decrypt with SharedK
Get T_{pub} and verify sign
SessionSec= crypto_x25519(K^S_{priv}, T_{pub})
SessionK= blake2b(SessionSec, K^S_{pub}, T_{pub})
Enc{ID_B, T_{pub}, K^S_{pub}, $Sign_{ED25519}$(ID_A, T_{pub})$_{Kpriv\_S}$}$_{SessionK}$
SessionSec= crypto_x25519(T_{priv}, K^S_{pub})
SessionK= blake2b(SessionSec, T_{pub}, K^S_{pub})
Decrypt with SessionK
Verify sign, T_{pub}, K^S_{pub}

Figure 2: Exchange of shared key and temporary session shared key for encryption

project, which compile in both C99 and C++. Among its various functions, this library enables the generation of a public key from a given secret using the x25519 protocol. As depicted in both branches of Figure 1, the private key is given as input to the asymmetric key generator and is used to generate two key pairs. The first one is the long-live key pair (K_{priv},K_{pub}), that can be used for the identification, while the second one is the temporary key pair (T_{priv},T_{pub}), that can be leverage for the creation of a session key. The key difference between the two key pairs is the way in which the PUF response is used.

4.3. Key Exchange

Original keys and temporary keys are exploited to securely obtain a session key to encrypt exchange data, as depicted in Figure 2. Specifically, crypto_x25519() performs an X25519 key exchange between NodeA and Server, both nodes have to use its secret key and the counterpart's public key to create a Shared Secret (SharedSec). The shared secret, known only to those who know a relevant secret key (NodeA and Server), is not cryptographically random. It can not be used directly as a key. It has to be hashed in both parts, concatenated with its public key and counterpart's public key using blake2b(), a cryptographically secure hash based on the ideas of ChaCha20, to obtain a shared key (SharedK). Node A sends the concatenation of its identifier (ID), (T_{pub}) (constructed as described in Section 4.2) and a sign of both, all encrypted with the SharedK. Encryption is performed using the AES algorithm, which in IoT devices provides a combination of security, efficiency, and adaptability, making it a popular choice for safeguarding sensitive data exchanged and stored on these resource-limited devices. On the other hand, signature generation is carried out using ED25519, which is an appealing choice due to its combination of resource efficiency, performance, security, and ease of implementation. The Server, after decrypting the message and verifying the sign, calculates the Session Key (SessionK) with the same method used for the SharedK, but using the T_{pub} of Node A. It sends back the concatenation of its ID, T_{pub} obtained from node A and a sign of both, all encrypted with the SessionK. Node A computes the SessionK in same way, decrypt the message and very the sign, T_{pub} and the server K_{pub}, which have to be the same of the first message to avoid replay attack. The obtained SessionK can be used to encrypt the communication between server and node, in order to allow the secure exchange of information for all the functions related to the management of identity.

5. Medical use-case

In this section, we present a possible integration of the proposed system within a classical IoMT context by considering the authentication use case. Taking into account the advantages introduced by SSI, which can be seen as a complete framework for identification, authentication, and data encryption; we will leverage it as a possible implementation of our system. Typical IoMT scenarios are composed of multiple devices exchanging data with a gateway or, in more advanced systems, directly with an external server. For our scenario, we envision an IoT device comprising a secure element, such as the ATMEGA328P, and an ESP-32-based board for transmitting data to a server. This device continuously sends data to the server or at specified intervals. To ensure secure authentication and robust session key management, it is needed to implement secure identification mechanisms and flexible, renewable session key management protocols. These mechanisms safeguard user data during exchanges and mitigate identity spoofing risks. As evidenced by existing literature, the adoption of certificates or SSI-based solutions significantly enhances both the reliability and privacy of end-users. Therefore, integrating such solutions into IoMT environments promises to bolster security while preserving user privacy. The system proposed within this manuscript offers all the cryptographic operations needed for the implementation of SSI, which will be discussed below.

5.1. DID Document

According to DIDs Recommendation [20]: *A DID document can express verification methods, such as cryptographic public keys, which can be used to authenticate or authorize interactions with the DID subject or associated parties.*

In our implementation, we demonstrated the capability of IoT devices to extract a private key K^{A}_{priv} directly from SRAM material, endowing them with unique features. This facilitates the association of each IoT device with a long-lived public key K^{A}_{pub}, thereby enabling their identification. According to the SSI definition, each IoT device must possess a unique DID, which can be derived from the K^{A}_{pub} by applying a hash function to the public key. While previous solutions utilized the base58 function for this purpose, in our approach, the final DID used for IoT device identification is formulated as follows: $DID^{A} = did : exp : base58(x25519\_public\_key(K_{priv}))$.

Through the utilization of this DID, it becomes feasible to uniquely identify an IoT device and validate its identity by verifying the signature applied to a message. Consequently, the issuer gains the capability to directly issue arbitrary data to the IoT device, which can subsequently be utilized for authentication purposes. The IoT device possesses the capability to consistently derive its associated private key K_{priv}, even in the absence of explicit storage within the device. While effective for identification purposes, it is apparent that this key cannot serve communication needs.

5.2. Session Key

Once the long-live identifier has been defined, establishing a session key becomes imperative prior to initiating the authentication process. As demonstrated in the preceding section, we presented a temporary key pair extractor (Fig. 2) to address this requirement. To mitigate potential vulnerabilities, we propose a strategy wherein a distinct key is defined for each communication session. In our proposal, we assume that the server does not implement key derivation, a measure sufficient to thwart replay attacks, given that the message exchanged from the IoT node A includes a form of nonce represented by its derived temporary key.

5.3. Signature

Upon completion of the exchange procedure by the server, it gains the ability to verify the identity of the IoT device by examining the signature applied to the ID_A message, which contains the DID of the party. Generating a signature in this scenario is relatively straightforward, as we already have an extracted private key available for credential signature. Once two parties have established a common symmetric secret, which dynamically changes with each new session, data and VCs can be encrypted for communication. The IoT Node typically retains a VC necessary for demonstrating its identity and the identity of the issuer. This enables authentication within the IoT node and facilitates the creation of a shared secret that can be utilized in lightweight algorithms.

6. Results

In this section, we will focus on the performance of the proposed system in terms of time needed to execute the operations described in the section 4. For the evaluation of the performance we used an Arduino Uno equipped with ATMEGA328P microcontroller. Such a microcontroller holds 16MHz for the frequency clock, a 32KB flash memory, 2KB SRAM, and 1KB EEPROM memory. It is used only for the extraction of private key, which is used by an ESP32 for implementing the remaining functions needed to run SSI protocol. We decided to use this device to demonstrate the feasibility of the proposed system also within a really constrained environment. Regarding the work performed on Arduino, the time required for computing both original keys and temporary keys was measured, as explained in Section 4.2. The results obtained indicate an average time of 2.4 seconds for the former and 0.2 seconds for the latter. This disparity arises from the fact that the computational workload of the FE in reconstructing the key is more substantial, even though manageable as it only needs to be executed once

Table 2
Maximum, minimum and average time in milliseconds of original and temporary keys over Arduino Uno using PUFs.

Metric	Minimum	Maximum	Average
Original key	2110	2790	2444
Temporary key	193	211	203

Table 3
Maximum, minimum and average time in microsecond of steps depicted in Figure 2.

Metric	Minimum	Average	Maximum
Public key gen	153220	15339	15402
Shared Sec	15035	15105	15198
Shared Key	860	871	882
Ed25519 keys	59825	59946	60023

Table 4
Average time in microsecond of steps depicted in Figure 2 varying block size in bytes.

Metric	128	512	1024	2048	4096
Create signature	60079	60644	61424	62943	65987
Verifying signature	97149	96469	95413	97947	98138
AES encrypt	148	389	701	1357	2646
AES decrypt	86	326	655	1296	2594

upon restart. Measurements were also conducted for the various primary steps of the key exchange scheme outlined in Section 4.3. Specifically, the times for generating the public key according to the x25519 protocol, generating the shared secret and shared key, and generating the keys according to the Ed25519 protocol were evaluated. The results indicate very low microsecond-level times, on the order of 10^-2 seconds. Additionally, times were assessed for encryption, decryption, signing, and verification according to the AES and Ed25519 protocols, gradually varying the input size while remaining within the data exchange size ranges required for the operation of the SSI mechanism. The results demonstrate nearly constant times for signing and verification, namely 60 milliseconds for signing and nearly 100 milliseconds for verification. However, encryption and decryption times increase with increasing input size, yet remain relatively low for the size of the data utilized.

7. Conclusion

The rapid advancement of IoT technologies, particularly within the medical domain, has greatly enhanced reliability and efficiency in healthcare systems. However, the increasing vulnerability of these devices due to inadequate authentication mechanisms presents significant privacy and security risks. To mitigate these issues, user-centric authentication leveraging asymmetric encryption has emerged as a crucial security measure across various domains. This work introduced a novel approach to authentication within the IoMT by integrating SSI and SRAM-

PUF. Our approach tries to offer robust security in resource-constrained environments without compromising operational efficiency. Through a concrete implementation exploiting the PUF mechanism, the solution demonstrates convenient time complexity, indicating that it is efficient and suitable for practical use. Future work will focus on testing the formal security of the proposed authentication mechanism. This includes employing formal verification, threat modeling, and conducting real-world deployment studies to ensure robustness and practicality in real-world scenarios.

References

[1] R. Dwivedi, D. Mehrotra, S. Chandra, Potential of internet of medical things (iomt) applications in building a smart healthcare system: A systematic review, Journal of oral biology and craniofacial research 12 (2022) 302–318.

[2] M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty, Y. Park, Iomt malware detection approaches: Analysis and research challenges, IEEE Access 7 (2019) 182459–182476. doi:10.1109/ACCESS.2019.2960412.

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

[4] P. D. More, S. R. Sakhare, P. Mahalle, Identity management in the internet of things: A survey of the state of the art, IEEE Systems, Man, and Cybernetics Magazine 9 (2023) 13–19. doi:10.1109/MSMC.2022.3230215.

[5] G. Fedrecheski, J. M. Rabaey, L. C. P. Costa, P. C. Calcina Ccori, W. T. Pereira, M. K. Zuffo, Self-sovereign identity for iot environments: A perspective, in: 2020 Global Internet of Things Summit (GIoTS), 2020, pp. 1–6. doi:10.1109/GIOTS49054.2020.9119664.

[6] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, M.-D. M. Yu, Efficient fuzzy extraction of puf-induced secrets: Theory and applications, volume 9813, 2016, pp. 412–431. doi:10.1007/978-3-662-53140-2_20.

[7] M. Barbareschi, P. Bagnasco, D. Amelino, A. Mazzeo, Designing an sram puf-based secret extractor for resource-constrained devices, International Journal of Embedded Systems 9 (2017) 353–364.

[8] K. Lata, L. R. Cenkeramaddi, Fpga-based puf designs: A comprehensive review and comparative analysis, Cryptography 7 (2023). URL: https://www.mdpi.com/2410-387X/7/4/55. doi:10.3390/cryptography7040055.

[9] S. Singh, P. K. Sharma, S. Y. Moon, J. H. Park, Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions, Journal of Ambient Intelligence and Humanized Computing (2017) 1–18.

[10] S. Ullah, J. Zheng, N. Din, M. T. Hussain, F. Ullah, M. Yousaf, Elliptic curve cryptography; applications, challenges, recent advances, and future trends: A comprehensive survey, Computer Science Review 47 (2023) 100530. URL: https://www.sciencedirect.com/science/article/pii/S1574013722000648. doi:https://doi.org/10.1016/j.cosrev.2022.100530.

[11] R. R. Krishna, A. Priyadarshini, A. V. Jha, B. Appasani, A. Srinivasulu, N. Bizon, State-of-the-

art review on iot threats and attacks: Taxonomy, challenges and solutions, Sustainability 13 (2021) 9463.

[12] D.-Z. Sun, Y.-N. Gao, Y. Tian, On the security of a puf-based authentication and key exchange protocol for iot devices, Sensors 23 (2023). URL: https://www.mdpi.com/1424-8220/23/14/6559.

[13] A. K. Boke, S. Nakhate, A. Rajawat, Fpga implementation of puf based key generator for secure communication in iot, Integration 89 (2023) 241–247. URL: https://www.sciencedirect.com/science/article/pii/S016792602200178X. doi:https://doi.org/10.1016/j.vlsi.2022.12.006.

[14] S. Buchovecká, R. Lórencz, J. Buček, F. Kodýtek, Symmetric and asymmetric schemes for lightweight secure communication, in: S. Furnell, P. Mori, E. Weippl, O. Camp (Eds.), Information Systems Security and Privacy, Springer International Publishing, Cham, 2022, pp. 97–114.

[15] M. Masud, G. S. Gaba, P. Kumar, A. Gurtov, A user-centric privacy-preserving authentication protocol for iot-ami environments, Computer Communications 196 (2022) 45–54. URL: https://www.sciencedirect.com/science/article/pii/S014036642200370X. doi:https://doi.org/10.1016/j.comcom.2022.09.021.

[16] M. Luecking, C. Fries, R. Lamberti, W. Stork, Decentralized identity and trust management framework for internet of things, in: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2020, pp. 1–9. doi:10.1109/ICBC48266.2020.9169411.

[17] U. Javaid, M. N. Aman, B. Sikdar, Blockpro: Blockchain based data provenance and integrity for secure iot environments, in: Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems, 2018, pp. 13–18.

[18] S. R. Niya, B. Jeffrey, B. Stiller, Kyot: Self-sovereign iot identification with a physically unclonable function, in: 2020 IEEE 45th Conference on Local Computer Networks (LCN), IEEE, 2020, pp. 485–490.

[19] L. Vaillant, Monocypher: Boring crypto that simply works, 2024. URL: https://monocypher.org.

[20] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, M. Sabadello, J. Holt, Decentralized identifiers (dids) v1. 0, Draft Community Group Report (2020).