This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3645565

Desynchronization-Resistant Anonymous Authentication Protocol for RFID Systems Utilizing Physically Unclonable Functions

Fazal Muhammad, Akhtar Badshah, Xin Ai, Muhammad Waqas, *Senior Member, IEEE*, Jalal Khan, Athanasios V. Vasilakos, *Senior Member, IEEE*, and Houbing Song, *Fellow, IEEE*

***Abstract*—Radio frequency identification (RFID) systems are an indispensable part of many critical Internet of Things (IoT) applications, including supply chain management and access control. Ensuring strong security in these systems is critical to safeguarding sensitive information and protecting user privacy. In recent years, in order to meet the diversified security needs of RFID systems, authentication and key protocols based on physical unclonable functions (PUFs) have received wide attention. Nevertheless, existing protocols typically require RFID tags to prestore an excessive number of secret credentials and impose considerable computational and communication overheads, which prove challenging for resource-constrained RFID tag. Additionally, certain lightweight protocols fall short of achieving their intended security and functional objectives, exhibiting insufficient anonymity and untraceability, and vulnerability to desynchronization attacks. To address these critical challenges, this paper first proposes a lightweight anonymous authentication and key agreement protocol designed for an ideal PUF environment. The proposed protocol integrates the arbiter PUF with cryptographic hash functions, providing robust resistance to potential attacks while minimizing system overhead. Subsequently, an enhanced protocol specifically tailored for noisy PUF scenarios is presented. This protocol employs a fuzzy extractor to reliably derive stable keys from noisy PUF responses, thereby mitigating the instability caused by inherent noise. Through comprehensive security analysis and formal verification, as well as performance evaluations compared with existing state-of-the-art protocols, both protocols are demonstrated to overcome the limitations of prior protocols and provide efficient and practically feasible solutions well suited for resource-constrained RFID environments.**

***Index Terms*—RFID systems, Internet of Things (IoT), physically unclonable functions (PUFs), authentication, key agreement.**

I. Introduction

A. Motivation

Radio Frequency Identification (RFID) technology has emerged as a cornerstone of the Internet of Things (IoT), enabling seamless automation and data exchange across diverse domains such as smart retail, access control, and electronic healthcare [1]–[3]. Its cost-effectiveness and energy efficiency have driven widespread adoption, making RFID systems integral to modern IoT applications [4], [5]. However, the dependence on wireless communication makes the RFID system face serious security vulnerabilities. Malicious attackers can exploit these vulnerabilities through eavesdropping, tag forgery, replay attacks and desynchronization attacks, thus compromising user privacy and system integrity [6]–[8]. These security threats will not only endanger sensitive data, but also weaken the effectiveness of RFID technology in practical applications.

The resource-constrained nature of RFID tags further exacerbates these challenges. The authentication and key agreement (AKA) protocol is considered one of the most promising candidates to mitigate these security threats. However, although many existing AKA protocols provide robust security, they bring a lot of storage, computing and communication overhead, making them unsuitable for deployment on low-cost and resource-limited RFID tags [6]–[8]. Moreover, some lightweight protocols that rely only on cryptographic hash functions lack comprehensive security features, making them vulnerable to physical capture and desynchronization attacks [9], [10]. In addition, in these protocols, basic privacy requirements such as anonymity and untraceability are often ignored. Addressing these issues is crucial to ensure the safe and efficient operation of RFID systems in IoT applications.

B. Problem Statement

Although some commendable protocols have been proposed to enhance the security of RFID systems, they are limited by excessive resource overhead, complex system architecture and imperfect security properties [11], [12]. These limitations make it difficult for them to deploy in RFID systems with limited resources [13], [14]. In addition, the lack of an effective mechanism to resist Desynchronization attacks remains a key concern. Desynchronization attacks will disrupt the synchronization between RFID tags and backend servers, causing the system to fail to run and endanger user privacy. Therefore, there is an urgent need for a lightweight, secure and efficient AKA protocol to address these challenges while maintaining low computing and storage overhead.

C. Main Contributions

This paper proposes two PUF-based authentication protocols specially designed for RFID systems. The main contributions of this paper are as follows:

1) Using the high efficiency of cryptographic hash function and the hardware security features of PUF, this paper first

Corresponding authors: (Akhtar Badshah and Muhammad Waqas).

F. Muhammad is with the Department of Electrical Engineering, University of Engineering and Technology, Mardan 23200, Pakistan (e-mail: fazal.muhammad@uetmardan.edu.pk).

A. Badshah is with the Department of Software Engineering, University of Malakand, Dir Lower, Pakistan. (e-mail: akhtarbadshah@uom.edu.pk).

X. Ai is with the College of Computer Science, Beijing University of Technology, Beijing 100124, China, (email: aixin1022@gmail.com).

M. Waqas is with the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, United Kingdom, and also with the School of Engineering, Edith Cowan University, Perth, 6027 WA, Australia (e-mail: engr.waqas2079@gmail.com).

J. Khan is with the Department of Telecommunication Engineering, University of Engineering and Technology, Mardan 23200, Pakistan (e-mail: jalal@uetmardan.edu.pk).

Athanasios V. Vasilakos is with the Center for AI Research (CAIR), University of Agder (UiA), 4879 Grimstad, Norway (e-mail: thanos.vasilakos@uia.no).

H. Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD, USA (e-mail: h.song@ieee.org; songh@umbc.edu).

TABLE I: Existing authentication protocols: A comparative summary

Protocol	Year	Employed techniques	Descriptions	Limitations
Meher *et al.* [18]	2022	• ECC • Symmetric cryptography	• Storage overhead optimization • Low computational overhead	• Vulnerable to session key leakage attacks • Requires strict clock synchronization • No physical security guarantees
Tanveer *et al.* [19]	2022	• Authenticated encryption with associated data • Fuzzy extractor • Cryptographic hash function • PUF	• Multi-factor authentication • Dynamic device enrollment support • Implementation of PUF-based physical security	• Lacks robust untraceability • Requires strict clock synchronization
Masud *et al.* [21]	2022	• Cryptographic hash function	• Lightweight	• No physical security guarantees • Lacks anonymity • Vulnerable to privileged insider attacks
Zhang *et al.* [22]	2022	• Cryptographic hash function • PUF	• Implementation of PUF-based physical security • Lightweight	• Requires strict clock synchronization • Vulnerable to secret leakage attacks
Kumar *et al.* [24]	2023	• Blockchain • Cryptographic hash function • Hamming weight rotation operation	• Lightweight	• Vulnerable to server spoofing attacks • No physical security guarantees
Wu *et al.* [9]	2023	• Cryptographic hash function	• Lightweight	• Requires strict clock synchronization • Vulnerable to stolen sensor attack • Vulnerable to physical capture attack • Vulnerable to insider threats from compromised users or sensors
Chaudhary *et al.* [10]	2023	• Cryptographic hash function	• Lightweight	• No physical security guarantees • Fails to provide unlinkability and untraceability
Egide and Li [25]	2023	• Cryptographic hash function • ECC	• Supports revocation functionality	• Fails to achieve forward security • Vulnerable to ephemeral key leakage attack • Vulnerable to key compromise impersonation attack
Zhou *et al.* [27]	2024	• Fuzzy extractor • Cryptographic hash function • PUF	• Implementation of PUF-based physical security • Multi-factor authentication	• Vulnerable to privileged insider attacks • Vulnerable to stolen verifier attacks • Vulnerable to ephemeral key leakage attacks
Subramani *et al.* [29]	2024	• Blockchain • Cryptographic hash function • Symmetric cryptography • Bilinear pairing	• Robust security properties • User revocation support	• Exorbitant resource consumption • No physical security guarantees
Mahmood *et al.* [30]	2024	• Chaotic map • Cryptographic hash function • Fuzzy extractor • Symmetric cryptography	• Multi-factor authentication	• No physical security guarantees • Exorbitant resource consumption • Vulnerable to man-in-the-middle attacks

proposes a lightweight authentication protocol suitable for the ideal PUF environment. Subsequently, by integrating the fuzzy extractor, an enhanced protocol was proposed for noisy PUF scenarios. This dual strategy maintains low computational overhead while ensuring strong security, making it suitable for RFID devices with limited resources.

2) We conducted formal security analysis of the proposed protocol under the Real-or-Random (ROR) model and performed automated security verification using the AVISPA tool set. In addition, informal security analysis also confirms the resistance of the proposed protocol to critical attack vectors such as replay attacks, physical tampering, and desynchronization attacks.
3) The comparative evaluation with the existing protocols shows that the proposed protocols excel in security, functional characteristics and efficiency (including time, storage and communication costs), marking significant progress in this field.

D. Novelty

The proposed protocols introduce several key innovations that collectively address critical gaps in RFID system authentication. First, it adopts a hybrid security architecture that uniquely integrates cryptographic hash functions with PUF-based device authentication, thereby achieving algorithmic robustness and hardware-rooted trust while maintaining computational efficiency for resource-constrained devices. In addition, unlike existing protocols that rely solely on lightweight cryptographic primitives or PUFs [15], [16], the enhanced protocol further incorporates a fuzzy extractor to derive stable secrets from arbiter-PUF responses, ensuring reliable protocol operation even in systems affected by environmental noise. Second, the proposed protocols establish an independent session key between the RFID tag and the reader during each authentication round while simultaneously safeguarding device privacy-an aspect frequently overlooked in prior work [17]. Additionally, this paper employs a combined design methodology that integrates ROR-model-based proofs with AVISPA verification, effectively bridging formal analysis and practical validation. This dual-layer verification ensures both theoretical soundness and real-world applicability. Finally, the proposed protocols employ dynamic pseudonyms together with a dual maintenance mechanism for new and old credentials to provide anonymity and untraceability while preventing desynchronization attacks-an integrated set of capabilities unmatched by existing protocols. These innovations position the proposed protocols as a comprehensive authentication framework tailored to the unique security and performance requirements of RFID systems.

E. Paper Organization

The rest of the paper is structured as follows. In Section II, we review existing AKA protocols designed for resource-constrained IoT devices. Section III introduces our proposed PUF-based protocols in detail. Section IV covers the security analysis to show how the protocols resists common attacks. In Section V, we evaluate its performance in terms of computation, communication, and storage costs and compare it against recent state-of-the-art schemes. Finally, Section VI concludes the paper.

II. Related Work

In recent years, several AKA protocols have been proposed to address security challenges in resource-constrained IoT environments. Table I summarizes the relevant protocols, highlighting the cryptographic primitives used and their limitations. Meher *et al.* [18] utilized elliptic curve cryptography and

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3645565

TABLE II: List of notations

Notation	Description
TBS	Trusted backend server
RT_j	j-th RFID tag
RU_i	i-th RFID reader unit
TID_j	Temporary identity of entity RT_j
(C_j, RS_j)	Challenge-response pair of RT_j
r_k	k-th random number
AM_t	t-th authentication message
SK_{ij}	The session key established between RU_i and RT_j
N^o, N^n	Old and new status of notation N
$PUF(\cdot)$	Physical unclonable function
$\mathcal{G}en(\cdot)$	Generation function
$\mathcal{R}ep(\cdot)$	Reconstruction function
$h(\cdot)$	Hash function
$\oplus, \|$	XOR and concatenation operations
$\mathcal{A}$	Adversary

symmetric encryption to design a key-less RFID authentication framework. Although it effectively reduces storage overhead. For smart grid applications, Tanveer *et al.* [19] introduced an authentication protocol based on authenticated encryption with associated data (AEAD), incorporating multi-factor verification to enhance robustness. However, Wu *et al.* [20] identified significant shortcomings in its traceability guarantees. In the context of smart healthcare, Masud *et al.* [21] proposed a three-party lightweight authentication protocol, which lacks full protection of user anonymity and remains vulnerable to insider privilege escalation. To enhance physical device security, Zhang *et al.* [22] proposed a PUF-based protocol, which was later shown by Tian *et al.* [23] to be susceptible to secret leakage attacks. In 2023, Kumar *et al.* [24] developed an RFID authentication protocol that integrates blockchain, cryptographic hashing, and Hamming-weight rotation. While the protocol ensures anonymity, forward secrecy, and resistance to several known attacks, it remains prone to server impersonation and physical tampering. Wu *et al.* [9] proposed a hash-based lightweight authentication protocol for smart healthcare; however, it does not support mutual authentication and lacks protection against sensor theft, physical capture, and insider threats. Chaudhary *et al.* [10] presented a lightweight authentication protocol for UAV networks, but its applicability is limited due to the absence of traceability and unlinkability features. Egide and Li [25] introduced HAP-SG, a heterogeneous device authentication framework for smart grids. Nonetheless, Cheng *et al.* [26] reported that it lacks forward security and is vulnerable to ephemeral secret leakage and key compromise impersonation. In 2024, Zhou *et al.* [27] proposed a PUF-enhanced lightweight protocol verified using BAN logic and ProVerif. However, Lee *et al.* [28] identified weaknesses including insufficient protection against privileged insider attacks, verifier compromise, and ephemeral secret leakage. Subramani *et al.* [29] designed an anonymous authentication protocol for fog-enabled smart grids, but its reliance on bilinear pairings introduces computational overhead unsuitable for constrained IoT devices. Lastly, Mahmood *et al.* [30] employed chaotic maps and hashing for mutual authentication between users and UAVs; however, Sharma and Mehra [31] showed that it is vulnerable to man-in-the-middle attacks and imposes high computational overhead.

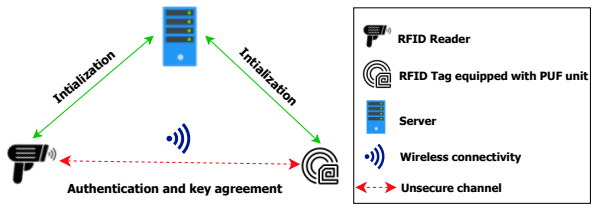


Fig. 1: System model for the proposed protocol.

III. Proposed Protocol

This section presents two anonymous AKA protocols for RFID systems. The first assumes ideal PUFs, suitable for settings where flawless PUF responses can be expected, while the second addresses real-world deployments, explicitly mitigating the stochastic noise inherent in PUF measurements. Table II lists the symbols used along with their definitions. To provide a clear understanding of the protocols, we first describe the system and adversary models, followed by a detailed exposition of each protocol phase.

A. Adversary Model

In this study, we used two classic adversary models, the Dolev-Yao (DY) model [32] and the Canetti-Krawczyk (CK) model [33] to describe potential security threats and evaluate the robustness of the proposed protocols. According to the assumption of the DY model, the adversary $\mathcal{A}$ can fully control the public communication channel and has the ability to intercept, change, delete and replay any transmitted message. On this basis, the CK model further enhances the ability of $\mathcal{A}$, allowing it to obtain short-term or long-term secrets to compromise the confidentiality and integrity of the communication process. In addition, $\mathcal{A}$ can also use the inherent physical exposure of RFID tags and readers to achieve physical capture or tampering. This comprehensive adversary model ensures that the proposed protocols are rigorously evaluated in a wide range of complex attack vectors, thus verifying its effectiveness in protecting the security of the RFID system.

B. System Model

Fig. 1 illustrates the RFID system model tailored to the proposed protocols, including three core components: RFID tag $\boldsymbol{RT_j}$, trusted backend server $\boldsymbol{TBS}$ and RFID reader RU_i. In this framework, TBS registers RT_j and RU_i through the secure channel and provides the necessary secret credentials, while all data exchanges between RU_i and RT_j are carried out on an insecure public channel. In addition, each RFID tag is embedded with a PUF unit to produce the unique challenge-response pair. In particular, any invasive attempt to tamper with PUF will irreversibly destroy the function of the tag and disable it. Ultimately, the purpose of the proposed protocols is to authenticate each other between RT_j and RU_i and establish a secure session key, so as to ensure the security of subsequent communication.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3645565

C. Proposed Anonymous Authentication Protocol for Ideal PUFs

This subsection presents a detailed description of each phase of the proposed protocol.

1) Setup Phase

In this phase, the trusted backend server TBS registers the RFID Reader RU_i and RFID Tag RT_j within the system through a secure channel and disseminates the necessary secret credentials to facilitate the subsequent authentication process.

Step 1. TBS initially generates a unique and unpredictable challenge value $\boldsymbol{C_j}$ for RT_j, and safely distributes this challenge to RT_j through a secure channel.

Step 2. After receiving the registration message from TBS, RT_j first calculates $\boldsymbol{RS_j = PUF(C_j)}$, and then forwards this response value back to TBS through the secure channel.

Step 3. Furthermore, TBS assigns a unique temporary identity $\boldsymbol{TID_j}$ to RT_j as a dynamic identifier for the subsequent authentication process. Finally, TBS stores the credentials $\{(\boldsymbol{TID_j^o, RS_j^o}), (\boldsymbol{TID_j^n, RS_j^n})\}$ in the memory of RU_i, in which coloredTID_j^o and TID_j^n are both configured as TID_j, while RS_j^o and RS_j^n are set to RS_j. Then, TBS safely transmits $\{TID_j, C_j\}$ through the secure channel and stores it in the memory of RT_j.

2) Authentication Phase

During this stage, RU_i and RT_j perform mutual authentication over a public channel, resulting in a secure session key that ensures the confidentiality and integrity of subsequent data transmissions.

Step 1. RU_i generates a random number r_1 and forwards $Msg_1 = \{r_1\}$ to RT_j over an insecure public channel.

Step 2. Upon receiving Msg_1 from RU_i, RT_j retrieves C_j from its memory and derives $RS_j = PUF(C_j)$. It then computes $AM_1 = h((r_1 + 1) \parallel RS_j)$, splits AM_1 into two equal parts AM_1^a and AM_1^b, and calculates $C_j^n = AM_1^a \oplus AM_1^b$, $RS_j^n = PUF(C_j^n)$, $AM_2 = RS_j \oplus RS_j^n$, and $AM_3 = h(AM_2 \parallel RS_j \parallel (r_1 + 1))$. Finally, RT_j sends $Msg_2 = \{TID_j, AM_2, AM_3\}$ back to RU_i.

Step 3. Upon the reception of Msg_2, RU_i extracts TID_j for verification. If TID_j matches either TID_j^o or TID_j^n, RU_i proceeds to retrieve the corresponding RS_j for further authentication. Otherwise, the connection is terminated.

Step 4. Furthermore, RU_i computes $RS_j^{n'} = RS_j \oplus AM_2$ and $AM_3' = h(AM_2 \parallel RS_j \parallel (r_1 + 1))$, and verifies whether $AM_3' = AM_3$. If the verification succeeds, RU_i calculates $AM_4 = h((r_1 + 2) \parallel RS_j)$, splits it into AM_4^a and AM_4^b, and updates its records such that $(TID_j^o = TID_j, RS_j^o = RS_j)$ and $(TID_j^n = AM_4^a \oplus AM_4^b, RS_j^n = RS_j^{n'})$. The temporary credential update is triggered only upon successful verification of AM_2, ensuring the authenticity of Msg_2, while the previous credential is retained as a backup to maintain synchronization with RT_j and provide resilience against message loss. Next, RU_i generates a new random number r_2 and computes $AM_5 = r_2 \oplus RS_j^{n'}$, the session key $SK_{ij} = h(RS_j \parallel r_2)$, and $AM_6 = h(RS_j^{n'} \parallel SK_{ij} \parallel TID_j^n)$. Finally, RU_i stores SK_{ij} and sends $Msg_3 = \{AM_5, AM_6\}$ to RT_j over an insecure public channel.

Step 5. Upon receiving $Msg_3 = \{AM_5, AM_6\}$, RT_j computes $r_2' = AM_5 \oplus RS_j^n$, $AM_7 = h((r_1 + 2)||RS_j)$, $TID_j^n = AM_7^a \oplus AM_7^b$, $SK_{ij}' = h(RS_j||r_2')$, and $AM_6' = h(RS_j^n||SK_{ij}'||TID_j^n)$. Subsequently, RT_j verifies whether $AM_6' = AM_6$. Upon successful verification, RT_j updates TID_j to TID_j^n and C_j to C_j^n. Ultimately, RT_j stores the session key SK_{ij}' in its memory to facilitate secure data transmission in the future.

The authentication phase between RT_j and RU_i described above is outlined in Fig. 2.

D. Proposed Enhanced Anonymous Authentication Protocol for Noisy PUFs

While differential PUF architectures improve reliability through structural optimization, process variations and environmental noise still significantly affect raw PUF responses in practical deployments [17]. To meet this challenge, we have proposed a lightweight and noise-resistant authentication protocol, which uses a fuzzy extractor to export a stable cryptographic key from the PUF output. The protocol is based on the coordinated operation of two core algorithms:

- $(\mathcal{K}, \mathcal{HD}) = \mathcal{G}en(\mathcal{D}ata)$: The generation function $\mathcal{G}en(\cdot)$ accepts $\mathcal{D}ata$ as input, and generates a unique cryptographic key $\mathcal{K}$ and its corresponding helper data $\mathcal{HD}$.
- $(\mathcal{K}) = \mathcal{R}ep(\mathcal{D}ata', \mathcal{HD})$: The reconstruction function $\mathcal{R}ep(\cdot)$ receives the noise-affected biased input $\mathcal{D}ata'$ and helper data $\mathcal{HD}$; if the Hamming distance between $\mathcal{D}ata'$ and the original $\mathcal{D}ata$ is kept below the predefined threshold, then $\mathcal{R}ep(\cdot)$ can tolerate the deviation between the two and accurately reconstruct the original cryptographic key $\mathcal{K}$.

1) Setup Phase

Step 1. Initially, TBS generates a unique random challenge $\boldsymbol{C_j}$ for RT_j and transmits it to RT_j over a secure channel.

Step 2. Upon receiving the TBS-issued challenge C_j, RT_j computes the response $\boldsymbol{RS_j = PUF(C_j)}$ and returns this value to TBS through the secure channel.

Step 3. Subsequently, TBS assigns a unique temporary identity $\boldsymbol{TID_j}$ to RT_j and invokes a fuzzy extractor to compute $\boldsymbol{(K_j, HD_j) = \mathcal{G}en(RS_j)}$. It then stores $\boldsymbol{\{(TID_j^o, K_j^o), (TID_j^n, K_j^n)\}}$ within RU_i's memory, where $TID_j^o = TID_j^n = TID_j$ and $K_j^o = K_j^n = K_j$. Finally, TBS transmits $< TID_j, HD_j >$ to RT_j over a secure channel and records the same in the RT_j's local storage.

RFID Reader Unit RU_i	**RFID Tag** RT_j
$\{TID_j^o, RS_j^o > \mid < TID_j^n, RS_j^n >\}$	$\{TID_j, (C_j, PUF(\cdot))\}$
Generate random number r_1; $\xrightarrow{Msg_1=\{r_1\}}$	Extract C_j; Compute $RS_j = PUF(C_j)$; $AM_1 = h((r_1+1)\|\|RS_j)$ $C_j^n = AM_1^a \oplus AM_1^b$ $RS_j^n = PUF(C_j^n)$; $AM_2 = RS_j \oplus RS_j^n$; $AM_3 = h(AM_2\|\|RS_j\|\|(r_1+1))$; $\xleftarrow{Msg_2=\{TID_j, AM_2, AM_3\}}$.
Search TID_j in its memory. if no match is found, connection is terminated. Compute $RS_j^{n'} = RS_j \oplus AM_2$, $AM_3' = h(AM_2\|\|RS_j\|\|(r_1+1))$; Check $AM_3' \stackrel{?}{=} AM_3$, if holds: Compute $AM_4 = h((r_1+2)\|\|RS_j)$; Update $(TID_j^o = TID_j, RS_j^o = RS_j)$ $(TID_j^n = AM_4^a \oplus AM_4^b, RS_j^n = RS_j^{n'})$; Generate r_2; Compute $AM_5 = r_2 \oplus RS_j^{n'}$; $SK_{ij} = h(RS_j\|\|r_2)$; $AM_6 = h(RS_j^{n'}\|\|SK_{ij}\|\|TID_j^n)$; Store SK_{ij}. $\xrightarrow{Msg_3=\{AM_5, AM_6\}}$	Compute $r_2' = AM_5 \oplus RS_j^n$; $AM_7 = h((r_1+2)\|\|RS_j)$ $TID_j^n = AM_7^a \oplus AM_7^b$ $SK_{ij}' = h(RS_j\|\|r_2')$; $AM_6' = h(RS_j^n\|\|SK_{ij}'\|\|TID_j^n)$; Check $AM_6' \stackrel{?}{=} AM_6$, if holds; Update $TID_j = TID_j^n$ $C_j = C_j^n$ Store: SK_{ij}'.
Both Tag and Reader-Server Unit store $SK_{ij} = h(RS_j \parallel r_2)$.	

Fig. 2: Authentication and key agreement protocol for RFID systems.

2) Authentication Phase

Analogously to the ideal PUF-based anonymous authentication protocol presented earlier, the enhanced protocol executes its AKA phase through the following steps:

Step 1. RU_i samples a nonce r_1 and conveys $Msg_1 = \{r_1\}$ to RT_j over an unsecured public channel.

Step 2. Upon receiving Msg_1, RT_j retrieves (C_j, HD_j) from its memory, computes $RS_j = PUF(C_j)$ and $K_j = \mathcal{R}ep(RS_j, HD_j)$. Subsequently, it derives $AM_1 = h((r_1+1)||K_j)$, $C_j^n = AM_1^a \oplus AM_1^b$, $RS_j^n = PUF(C_j^n)$, and obtains $(K_j^n, HD_j^n) = \mathcal{G}en(RS_j^n)$. With these values, RT_j constructs $AM_2 = K_j \oplus K_j^n$ and $AM_3 = h(AM_2||K_j||(r_1+1))$. Finally, RT_j returns $Msg_2 = \{TID_j, AM_2, AM_3\}$ over the public channel.

Step 3. Upon receipt of Msg_2, RU_i isolates TID_j for validation. If TID_j matches either TID_j^o or TID_j^n, RU_i retrieves the corresponding K_j to advance the authentication process; otherwise, the session is immediately aborted.

Step 4. Upon successful verification of the temporary identity, RU_i derives $K_j^{n'} = K_j \oplus AM_2$ and $AM_3' = h(AM_2\|K_j\|(r_1+1))$, then checks whether $AM_3' = AM_3$. With the equality established, RU_i computes $AM_4 = h((r_1+2)||K_j)$, updates its stored credentials to $(TID_j^o, K_j^o) = (TID_j, K_j)$ and $(TID_j^n = AM_4^a \oplus AM_4^b, K_j^n = K_j^{n'})$. Subsequently, it samples a fresh nonce r_2 and computes $AM_5 = r_2 \oplus K_j^{n'}$, $SK_{ij} = h(K_j||r_2)$ and $AM_6 = h(K_j^{n'}||SK_{ij}||TID_j^n)$. Finally, RU_i archives SK_{ij} for secure communications with RT_j and dispatches $Msg_3 = \{AM_5, AM_6\}$ to RT_j over the public channel.

Step 5. Upon successfully receiving Msg_3, RT_j evaluates $r_2' = AM_5 \oplus K_j^n$, $AM_7 = h((r_1+2)||K_j)$, $TID_j^n = AM_7^a \oplus AM_7^b$, $SK_{ij}' = h(K_j||r_2')$ and subsequently derives $AM_6' = h(K_j^n||SK_{ij}'||TID_j^n)$. It then verifies whether $AM_6' \stackrel{?}{=} AM_6$. If the check succeeds, RT_j updates $TID_j = TID_j^n$ and replaces $(C_j = C_j^n, HD_j = HD_j^n)$. Finally, RT_j stores SK_{ij}' for future secure communications.

IV. Security Analysis

This section presents a comprehensive security analysis of the proposed protocol, covering both formal and informal approaches.

A. Formal Security Analysis

We perform a rigorous formal security analysis of the proposed protocol using the ROR model [34], focusing on the following key security properties.

1) Formal Security Model

Participants. In the ROR model, participants represent the communicating entities in the protocol. In our proposed protocol, we define two principal entities: the RFID reader unit RU_i and the RFID tag RT_j, denoted by $\mathcal{P}^u$ and $\mathcal{P}^j$, respectively, corresponding to instance u of RU_i and instance j of RT_j. The notation $\mathcal{P}^e$ is used to generically refer to any participant instance.

Accept State. An instance $\mathcal{P}^e$ is said to be in the *accept* state if it has successfully completed all protocol steps without termination and has established a valid session key with its intended peer. Furthermore, the session identifier SID uniquely distinguishes each established session involving $\mathcal{P}^e$, thereby serving as a unique label for that session.

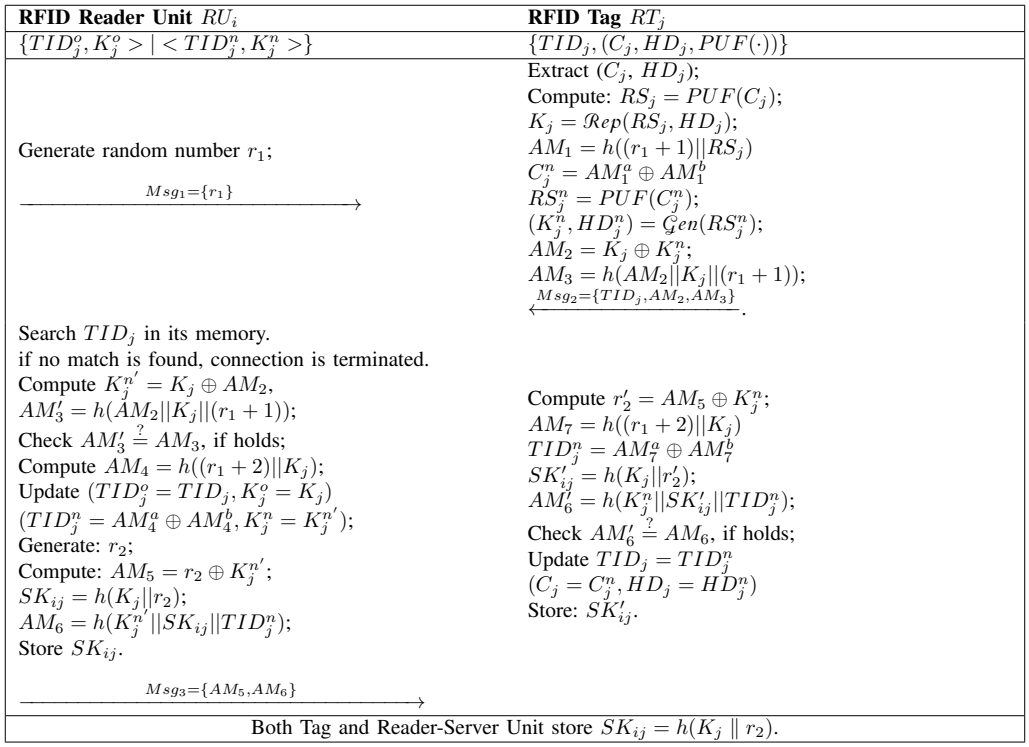


Fig. 3: Enhanced authentication and key agreement protocol for RFID systems.

Partnering. Two protocol instances $\mathcal{P}^{e_1}$ and $\mathcal{P}^{e_2}$ are considered *partners* if both are in the accept state and share the same SID.

Freshness. An instance $\mathcal{P}^e$ is *fresh* if its session key has not been revealed to the adversary $\mathcal{A}$, and none of its partnered instances have exposed the session key.

Adversary. Following the model in Section III-A, the adversary $\mathcal{A}$ is assumed to have full control over the communication channels and can perform various attacks, including eavesdropping, message modification, replay, and injection of arbitrary messages, aiming to compromise protocol security. To formally capture these capabilities, we define the following oracle queries:

- $Execute(\mathcal{P}^u, \mathcal{P}^j)$: Models a passive attack. The adversary $\mathcal{A}$ obtains all messages exchanged between $\mathcal{P}^u$ and $\mathcal{P}^j$ during a normal, honest protocol execution.
- $Send(\mathcal{P}^e, msg)$: Models active attacks such as replay and message tampering. The adversary sends a crafted message msg to instance $\mathcal{P}^e$ and obtains its response.
- $Test(\mathcal{P}^e)$: Used to evaluate the semantic security of the session key. This query can only be issued to a fresh instance in the accept state. A random bit $b \stackrel{\$}{\leftarrow} \{0,1\}$ is chosen:
  - If $b = 1$, the real session key is returned.
  - If $b = 0$, a random string of the same length as the session key is returned.

  The adversary's task is to guess the value of b. If it can correctly guess b with non-negligible probability, the protocol fails to achieve semantic security.

Definition 1. ***(Semantic Security of the Session Key)****: Within the ROR model, the session key established between the reader-server unit S and the RFID tag Tag_j is considered secure if no PPT adversary $\mathcal{A}$ can distinguish it from a random string with more than negligible advantage. Let Suc denote the event that $\mathcal{A}$ correctly guesses the hidden bit b in the Test query. The advantage of $\mathcal{A}$ against protocol P is defined as:*

$$Adv_P^{aka}(\mathcal{A}) = |2 \cdot \Pr[Suc] - 1|$$

Protocol P achieves semantic security if $Adv_P^{aka}(\mathcal{A}) \leq \epsilon(\lambda)$, where $\epsilon(\lambda)$ is negligible in the security parameter λ.

2) Security Proof

Theorem 1. *Let P denote the proposed protocol, and let $\mathcal{A}$ be a PPT adversary against its semantic security. Suppose $\mathcal{A}$ makes at most q_s Send queries, q_h hash queries, and q_p PUF queries. Let $|Hash|$ and $|PUF|$ be the range sizes of the one-way hash function $h(\cdot)$ and the PUF, respectively. Then:*

$$Adv_P^{aka}(\mathcal{A}) \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|}$$

Proof. We use a sequence of games G_i ($i = 0, 1, 2, 3$). Let Suc_i be the event that $\mathcal{A}$ correctly guesses b in game G_i.

G_0: This is the real ROR security game. The adversary interacts with the protocol under normal conditions:

$$Adv_P^{aka}(\mathcal{A}) = |2 \cdot \Pr[Suc_0] - 1| \quad (1)$$

G_1: Here, $\mathcal{A}$ uses $Execute(\mathcal{P}^s, \mathcal{P}^t)$ to eavesdrop on communications between Tag_j and S. To determine $SK_{ij} = h(RS_j \parallel r_2)$, it would need both the fresh random value r_2 and the PUF response RS_j of Tag_j, neither of which are revealed. Thus:

$$\Pr[Suc_1] = \Pr[Suc_0] \quad (2)$$

G_2: This game adds the simulation of $Send$ and hash queries. For $\mathcal{A}$ to succeed, it must forge a valid message that passes

authentication, which requires finding a hash collision. Due to the inclusion of random nonces and secret credentials, the best it can do is exploit the birthday bound:

$$|\Pr[Suc_2] - \Pr[Suc_1]| \leq \frac{q_h^2}{2|Hash|} \tag{3}$$

G_3**:** We now include the simulation of PUF queries. Due to the unpredictability and uniqueness properties of the PUF, the probability of correctly guessing the required responses is bounded by:

$$|\Pr[Suc_3] - \Pr[Suc_2]| \leq \frac{q_p^2}{2|PUF|} \tag{4}$$

In G_3, the adversary's view is independent of b, so:

$$\Pr[Suc_3] = \frac{1}{2} \tag{5}$$

Combining results: From (1)-(5):

$$\begin{aligned} Adv_P^{aka}(\mathcal{A}) &= |2 \cdot \Pr[Suc_0] - 1| \\ &= 2\,|\Pr[Suc_0] - \Pr[Suc_3]| \\ &\leq 2\big(|\Pr[Suc_1] - \Pr[Suc_0]| \\ &\quad + |\Pr[Suc_2] - \Pr[Suc_1]| \\ &\quad + |\Pr[Suc_3] - \Pr[Suc_2]|\big) \\ &\leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} \end{aligned} \tag{6}$$

■

B. Informal Security Analysis

This section analyzes the security properties of the proposed protocols, assessing their resistance to common RFID threats through logical evaluation of the protocol mechanisms.

1) Mutual authentication

In the two proposed protocols, the mutual authentication between the RFID reader unit RU_i and the RFID tag RT_j is completed in steps 4 and 5 of the AKA stage. In step 4, RU_i calculates $AM_3' = h(AM_2 \,\|\, RS_j \,\|\, (r_1+1))$, and verifies whether $AM_3' = AM_3$ to authenticate RT_j and confirm the legality of Msg_2. Then, in step 5, RT_j calculates $AM_6' = h(RS_j^n||SK_{ij}'||TID_j^n)$, and verifies whether $AM_6' = AM_6$ to authenticate RU_i and confirm the legitimacy of Msg_3. This dual authentication mechanism ensures that both parties can confirm each other's identity: the reader verifies the tag through the parameters derived from the PUF response, while the tag verifies the identity of the reader through the session key and updated credential verification. Successful completion of two verifications can establish trusted communication and establish a secure session key between RU_i and RT_j.

2) Anonymity and untraceability

In the architecture of the two proposed protocols, RU_i and RT_j exchange three different messages: $Msg_1 = \{r_1\}$, $Msg_2 = \{TID_j, AM_2, AM_3\}$ and $Msg_3 = \{AM_5, AM_6\}$. These communications contain session-specific random numbers and confidential encryption parameters derived from the PUF response, effectively preventing $\mathcal{A}$ from trying to track a single session. The temporary identity TID_j for tag identification will be systematically refreshed through the dynamic update mechanism in each successful authentication cycle. Specifically, after successful verification, RT_j will calculate a new $TID_j^n = AM_7^a \oplus AM_7^b$ and replace its previous identifier to ensure the anonymity of the session and prevent session correlation by adversaries. This identity update only occurs after the successful authentication of both parties to maintain synchronization between the communicating parties. The combination of temporary random values, dynamic identity updates and PUF-derived cryptographic parameters provides a strong guarantee for the anonymity and untraceability during protocol operation.

3) Desynchronization attacks

The proposed protocols demonstrate the strong resistance to desynchronized attacks through the dual-credential storage mechanism. In the protocol for the ideal PUF scenario, RU_i maintains the current operating credentials (TID_j^o, RS_j^o) and outstanding update credentials (TID_j^n, RS_j^n) of each registered RFID tag RT_j. This redundant storage strategy ensures that the authentication capability is still sustainable even if $\mathcal{A}$ interferes with the message exchange in the AKA stage. Specifically, if Msg_3 is intercepted by $\mathcal{A}$, resulting in the failure of the credential synchronization between RU_i and RT_j, when RT_j initiates a subsequent session, RU_i can still use the old credential set to authenticate RT_j. The same double-credential method is also implemented in the protocol proposed for noise PUF scenarios, in which RU_i stores (TID_j^o, K_j^o) and (TID_j^n, K_j^n). More importantly, the credential update only occurs after mutual authentication is successful: RU_i refreshes its stored credentials after verifying AM_3, while RT_j only updates its credentials after verifying AM_6. This conditional update logic is combined with the credential redundancy storage mechanism to ensure that the proposed protocols can maintain continuity when affected by desynchronization attacks.

4) Replay attacks

The proposed protocol prevents replay attacks by embedding cryptographic freshness mechanisms in all exchanged message. At the beginning of each session, RU_i will generate a new random nonce r_1 in $Msg_1 = \{r_1\}$, which will dynamically initialize the subsequent authentication parameters. The tag's response $Msg_2 = \{TID_j, AM_2, AM_3\}$ incorporates this nonce through the computation $AM_3 = h(AM_2 \,\|\, RS_j \,\|\, (r_1+1))$ for ideal PUFs (or $AM_3 = h(AM_2 \,\|\, K_j \,\|\, (r_1+1))$ for noisy PUFs), binding the response to the current session. Similarly, $Msg_3 = \{AM_5, AM_6\}$ derives its validity from reader-generated r_2 and real-time credential updates. Session-related operation chains, including random nonce transformations (r_1+1, r_1+2), irreversible PUF response and cryptographic hashing, to ensure that all authentication parameters are temporary and non-reusable. Therefore, $\mathcal{A}$ can not exploit replayed messages: the replayed Msg_1 will cause RT_j to generate fresh cryptographic values, but RU_i will reject Msg_2 due to the mismatch of the session context during the verification process; the replayed Msg_2 will fail the verification of AM_3 at RU_i (step 4) due to random nonce binding errors; the replayed Msg_3 will be rejected at RT_j (step 5) due to outdated credentials and session key

mismatch. This comprehensive freshness architecture ensures strong replay protection.

5) Physical tampering or cloning attacks

The proposed protocols fundamentally mitigate physical tampering and cloning attacks through the intrinsic properties of PUFs. Each RFID tag RT_j incorporates a PUF that generates a unique, unpredictable response $RS_j = \text{PUF}(C_j)$ based on its challenge C_j. The PUF's unclonability stems from nanoscale manufacturing variations that cannot be reproduced or physically characterized, rendering extraction of the exact challenge-response behavior infeasible even with advanced probing techniques. In the ideal-PUF protocol, this guarantees the secrecy of the (C_j, RS_j) pair, while in the noisy-PUF variant, the fuzzy extractor ensures the derived cryptographic key K_j remains secure despite helper data HD_j storage. Crucially, successful cloning would require either: 1) Perfect duplication of the PUF's physical microstructure (physically impossible), or 2) Mathematical reconstruction of the challenge-response map (computationally infeasible due to PUF nonlinearity and the large CRP space). Furthermore, any invasive physical tampering attempt would alter the PUF's physical characteristics, thereby changing its challenge-response behavior and rendering the tag inoperative. Consequently, both protocols provide robust protection against physical tampering and cloning attempts.

6) Man-in-the-middle and impersonation attacks

The proposed protocols provide robust protection against man-in-the-middle and impersonation attacks through cryptographic binding of all messages to physically unclonable secrets. To successfully execute such attacks, adversary $\mathcal{A}$ must forge valid $Msg_2^{\mathcal{A}} = \{TID_j^{\mathcal{A}}, AM_2^{\mathcal{A}}, AM_3^{\mathcal{A}}\}$ and $Msg_3^{\mathcal{A}} = \{AM_5^{\mathcal{A}}, AM_6^{\mathcal{A}}\}$ that pass verification in Steps 4 and 5. In the ideal-PUF protocol, generating authentic $Msg_2^{\mathcal{A}}$ requires computation of valid $AM_2^{\mathcal{A}} = RS_j \oplus RS_j^n$ and $AM_3^{\mathcal{A}} = h(AM_2^{\mathcal{A}} \parallel RS_j \parallel (r_1 + 1))$, which necessitates knowledge of the tag's secret PUF response RS_j. Similarly, forging $Msg_3^{\mathcal{A}}$ demands access to the updated response RS_j^n and valid session key material. These parameters are physically irreproducible: RS_j is inherently bound to the tag's PUF microstructure and cannot be extracted through physical tampering, while RS_j^n is dynamically generated from the PUF during each session. In the noisy-PUF protocol, equivalent protection is achieved through the fuzzy-extracted key K_j, which remains secure even if $\mathcal{A}$ obtains C_j and HD_j from tag memory, as K_j can only be regenerated using the physical PUF. Consequently, without access to the actual PUF device, $\mathcal{A}$ cannot fabricate messages that satisfy the cryptographic verification checks, rendering both man-in-the-middle and impersonation attacks computationally infeasible.

7) Perfect forward secrecy

The proposed protocols ensure perfect forward secrecy by deriving session keys from ephemeral parameters that are never stored long-term. Even if adversary $\mathcal{A}$ compromises the long-term secrets stored in RT_j's memory (such as C_j and HD_j) or the current session key SK_{ij}, the security of past and future sessions remains unaffected. This protection arises because each session key is derived from fresh random nonces (r_1, r_2) and dynamically updated PUF responses that are discarded after each session. Specifically, the session key SK_{ij} is computed as $h(RS_j \parallel r_2)$ (or $h(K_j \parallel r_2)$ for noisy PUFs), where both RS_j (or K_j) and r_2 are ephemeral values that change with each authentication instance. The PUF responses themselves are never stored permanently and are regenerated from updated challenges in each session. Consequently, compromise of long-term stored credentials or current session keys does not enable $\mathcal{A}$ to compute past or future session keys, thereby guaranteeing perfect forward secrecy.

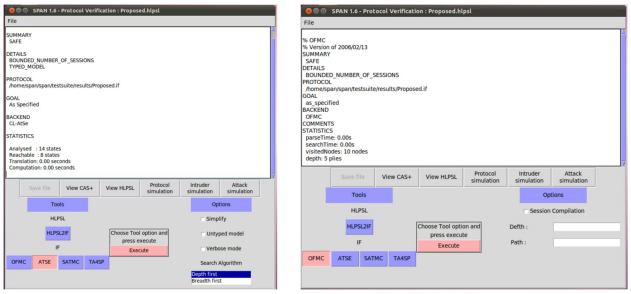


Fig. 4: Results of security verification using CL-AtSe and OFMC back-ends in AVISPA.

C. Security Verification

In this section, we conduct a formal security verification of the proposed protocol using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [35]. AVISPA employs the High-Level Protocol Specification Language (HLPSL) [36] to formally model the protocol and its intended security properties, including the confidentiality of the established session key and mutual authentication between communicating entities.

For this purpose, a complete AVISPA environment was deployed on SPAN-Ubuntu 10.10 using VirtualBox. Among the suite of integrated backends, the On-the-Fly Model Checker (OFMC) and the Constraint-Logic-based Attack Searcher (CL-ATSE) are widely recognized and employed for security assessments. As summarized in Fig. 4, the analysis results from both OFMC and CL-ATSE confirm that no attack trace was identified within the bounded model, which substantiates the robustness of the proposed protocol under the DY model, with no vulnerabilities or traces of exploitable attacks being reported.

V. Performance Analysis and Comparison

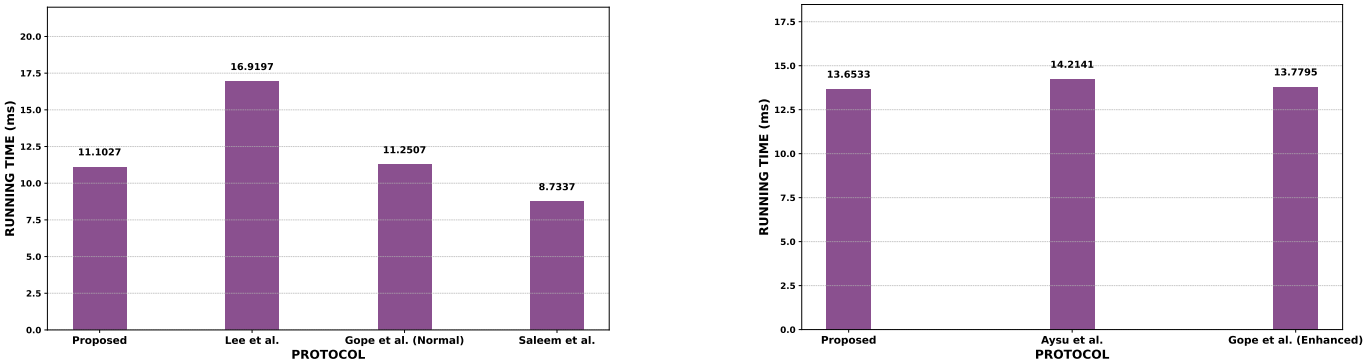
This section provides a comprehensive evaluation of the proposed protocols and the state-of-the-art protocols. In view of the strict resource constraints of RFID tags, time costs, communication costs and storage costs must be strictly evaluated to determine their practical feasibility. In addition, the comparative analysis of security assurance remains a critical measure to evaluate the effectiveness of any AKA protocol. Therefore,

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3645565

TABLE III: Comparison of computational overheads

Type	**Protocols**	**Tag** (ms)	**Reader/Server** (ms)	**TCO** (ms)
Ideal PUF	Lee *et al.* [15]	$6\mathrm{T_h} + 2\mathrm{T_p} \approx 1.0274$	$5\mathrm{T_h} + 2\mathrm{T_p} \approx 1.0229$	2.0503
	Gope *et al.* [17] (Normal)	$4\mathrm{T_h} + 2\mathrm{T_p} \approx 1.0184$	$4\mathrm{T_h} \approx 0.0180$	1.0364
	Saleem *et al.* [16]	$3\mathrm{T_h} + \mathrm{T_p} \approx 0.5137$	$3\mathrm{T_h} + 2\mathrm{T_s} \approx 0.4339$	0.9476
	Proposed protocol	$5\mathrm{T_h} + 2\mathrm{T_p} \approx 1.0229$	$4\mathrm{T_h} \approx 0.0180$	1.0409
Noisy PUF	Gope *et al.* [17] (Enhanced)	$4\mathrm{T_h} + 2\mathrm{T_p} + \mathrm{T_f} \approx 1.0286$	$5\mathrm{T_h} + \mathrm{T_f} \approx 0.0327$	1.0613
	Aysu *et al.* [37]	$3\mathrm{T_h} + 2\mathrm{T_p} + \mathrm{T_f} + \mathrm{T_s} \approx 1.2343$	$3\mathrm{T_h} + \mathrm{T_f} + \mathrm{T_s} \approx 0.2339$	1.4682
	Proposed protocol	$5\mathrm{T_h} + 2\mathrm{T_p} + 2\mathrm{T_f} \approx 1.0433$	$4\mathrm{T_h} \approx 0.0180$	1.0613

Note: **TCO**: Total Computation Overhead; $\mathrm{T_h}$: Average execution time of cryptographic hash; $\mathrm{T_p}$: Average execution time of PUF; $\mathrm{T_f}$: Average execution time of fuzzy extractor; $\mathrm{T_s}$: Average execution time of symmetric encryption/decryption.



(a) Ideal PUF scenario.

(b) Noisy PUF scenario.

Fig. 5: Comparison of runting time.

we first benchmark the proposed protocol with the protocol of Lee *et al.* [15], Gope *et al.* [17], and Saleem *et al.* [16] under the ideal PUF hypothesis. Then, we compare the enhanced protocol designed for the noise PUF environment with the protocols of Gope *et al.* [17] and Aysu *et al.* [37].

A. Experimental Setup

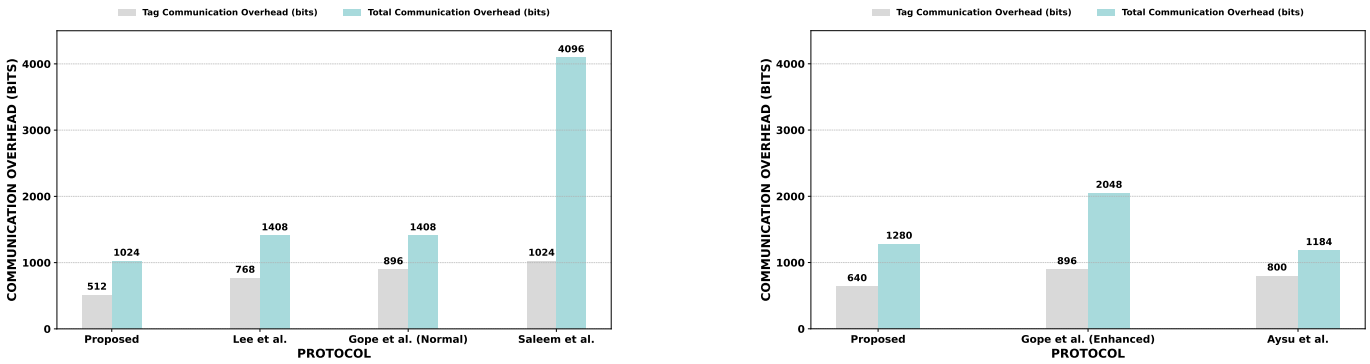
This section elaborates on the experimental settings used to evaluate the performance of the proposed protocols in comparison with representative benchmark protocols. All the core cryptographic primitives used in the protocols are implemented and simulated in Python 3.9.13, and established cryptographic libraries such as *hashlib*, *pypuf* and *cryptography* are utilized. The primitives of the evaluation include: the cryptographic hash function (T_h), physical unclonable function (T_p), fuzzy extractor (T_f), and AES symmetric encryption/decryption function (T_s). It should be noted that operations with negligible computational costs (e.g., XOR and concatenation) are not included in the performance evaluation. In order to ensure the fairness and consistency of the benchmarking process, all cryptographic primitives are encapsulated in functions with standardized input/output specifications. In order to maintain the consistency of parameter size, identities and random nonces are set to 128 bits, and the output of the hash function is 256 bits. The PUF primitive uses the ArbiterPUF from the pypuf library to accept a 128-bit challenge and generate a 128-bit response. Moreover, a fuzzy extractor is used to export a stable cryptographic key from the PUF response. Specifically, the generation function $\mathcal{G}en(\cdot)$ accepts a 128-bit PUF response RS and outputs a 256-bit cryptographic key K and a 144-bit helper data HD. In addition, the reconstruction function $\mathcal{G}en(\cdot)$ can successfully recover the original key under the condition of a given disturbance response RS' and helper data HD, provided that the Hanming distance meets $\mathrm{HamD}(RS, RS') \leq 8$.

In order to realistically reflect the performance in the limited environment, all cryptographic primitives are benchmarked on Raspberry Pi 5, which is equipped with 4GB of memory and runs Linux kernel version 6.12.2.25+rpt-rpi-2712. Each primitive is executed 100 times to get the average running time. The measurement results are summarized as follows: $\mathrm{T_h} \approx 0.0045$ms, $\mathrm{T_p} \approx 0.5002$ ms, $\mathrm{T_f} = 0.0102$ ms, $\mathrm{T_s} = 0.2102$ ms.

B. Comparison of Time Overhead

Based on the experimental configuration detailed in section V-A, this subsection comprehensively evaluates the time overhead of the proposed protocols compared with the representative baseline protocols. The evaluation mainly focuses on the computational burden caused by the cryptographic primitives on RFID tags and reader/server components during the authentication process. Table III summarizes the omputational cost of each protocol determined by the cryptographic primitives under the ideal and noise PUF settings. Under the ideal PUF scenario, the proposed protocol introduces a computational cost of $5\mathrm{T_h} + 2\mathrm{T_p} \approx 1.0229$ ms at the RFID tag side, which is

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3645565



(a) Ideal PUF scenario.

(b) Noisy PUF scenario.

Fig. 6: Comparison of communication overheads.

marginally lower than the 1.0274 ms required by the protocol of Lee *et al.* [15] In addition, on the reader/server side, the proposed protocol demonstrates a significantly reduced computational overhead of approximately $4\mathrm{T_h} \approx 0.0180$ ms, outperforming the respective costs of 1.0229 ms and 0.4339 ms reported in the works of Lee *et al.* [15] and Saleem *et al.* [16], respectively. In terms of overall system overhead, the proposed protocol yields a total computational cost of 1.0409 ms, which is substantially lower than the 2.0503 ms reported for Lee *et al.*'s protocol [15], representing a reduction of approximately 49.89%. Although the proposed protocol introduces slightly higher overall computational overhead compared to the protocols of Gope *et al.* [17] and Saleem *et al.* [16], this marginal increase is justified by its enhanced security functionality (as detailed in Table V). Furthermore, under the noise PUF scenario, the computational overhead of the proposed enhanced protocol for RFID tags is $5\mathrm{T_h} + 2\mathrm{T_p} + 2\mathrm{T_f} \approx 1.0433$ ms, and the total system overhead is 1.0613 ms The result is comparable to the overall time cost reported by Gope *et al.* [17], and is significantly better than the protocol proposed by Aysu *et al.* [37]. Given that the proposed protocol has comprehensive security feature coverage (as shown in Table V), it shows excellent practicality and competitiveness in resource-limited RFID systems.

In order to further evaluate the practical performance of each protocol, we realized the complete protocol process and tested it under the experimental conditions described in Section V-A. Each protocol is executed 100 times to get its average running time, and the corresponding result is shown in Fig. 5. The results show that in the ideal PUF environment, the average running time of the proposed protocol is 11.1027 ms, which is significantly better than the 16.9197 ms of Lee *et al.* [15] and the 11.2507 ms of Gope *et al.* [17]. Although it is slightly higher than the 8.7337 ms reported by Saleem *et al.* [16], this small increase is a reasonable trade-off compared with the additional security and functional enhancements provided by the proposed protocol (see Table V). In addition, in the noisy PUF environment, the average running time of the proposed protocol is 13.6533 ms, which is better than the 14.2141 ms of Aysu *et al.* [37] and 13.7795 ms of Gope *et al.* [17]. These results confirm the suitability of the proposed protocol in resource-constrained RFID systems.

C. Communication Overhead

Based on the experimental settings outlined in section V-A, Fig. 6 comprehensively evaluates the communication overhead of the proposed protocol compared with several representative baseline protocols in different scenarios. As shown in Fig. 6 (a), under the ideal PUF setting, the proposed protocol requires RFID tags to send messages $Msg_2 = \{TID_j, AM_1, AM_2\}$ at the AKA stage, resulting in a communication cost of $128+128+256 = 512$ bits. Compared with the 768 bits of Lee *et al.* [15], the 896 bits of Gope *et al.* [17] and the 1024 bits of Saleem *et al.* [16], the proposed protocol has significantly improved in terms of communication efficiency, particularly suited for resource-constrained RFID tags. Moreover, the total communication overhead of the proposed protocol is only 1024 bits, far lower than the 1408 bits and 4096 bits reported by Lee *et al.* [15] and Saleem *et al.* [16] respectively, and the communication cost can be reduced by up to 75%. In addition, as shown in Fig. 6 (b), under the noise PUF setting, the proposed enhanced protocol only applies a 640-bit communication load to the RFID tag, which is significantly less than the 896 bits required in Gope *et al.*'s protocol and the 800 bits in Aysu *et al.* [37]'s protocol. Furthermore, the total communication overhead of the proposed protocol is 1280 bits, which is better than the 2048 bits of the Gope *et al.* [17]'s protocol. Although it is slightly higher than the 1200 bits in the protocol of Aysu *et al.* [37], this increase is considered acceptable considering the enhanced security feature coverage of the proposed protocol (see Table V)).

D. Storage Overhead

Under the parameter normalization described in Section V-C, Table V-A presents a detailed comparison of storage overheads for each protocol. This evaluation considers the secret credentials that an RFID tag must store after initialization and registration, as well as the growth of the total system storage overhead with respect to the number of deployed tags n. In the ideal PUF setting, the proposed protocol requires only 256 bits of storage per tag-representing savings of 50% compared to the 512 bits of Lee *et al.* [15], a 33% reduction compared to the 384 bits of Gope *et al.* [17] (Normal), and a

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3645565

TABLE IV: Comparison of Storage Overheads

Type	Protocols	Tag S. Overhead	Total S. Overhead
Ideal PUF	Lee *et al.* [15]	512 bits	768n bits
	Gope *et al.* [17] (Normal)	384 bits	1024n bits
	Saleem *et al.* [16]	640 bits	(640n+128) bits
	Proposed protocol	256 bits	768n bits
Noisy PUF	Gope *et al.* [17] (Enhanced)	656 bits	1424n bits
	Aysu *et al.* [37]	256 bits	768n bits
	Proposed protocol	400 bits	1168n bits

Note: **S. Overhead**: Storage Overhead; n: Number of deployed RFID tags

TABLE V: Comparison of Security Properties

Types	Protocols	SP_1	SP_2	SP_3	SP_4	SP_5	SP_6	SP_7	SP_8
Ideal PUF	Lee *et al.* [15]	×	✓	✓	✓	✓	✓	✓	✓
	Gope *et al.* [17] (Normal)	✓	×	×	×	✓	✓	✓	✓
	Saleem *et al.* [16]	×	✓	✓	✓	×	✓	✓	✓
	Proposed Protocol	✓	✓	✓	✓	✓	✓	✓	✓
Noisy PUF	Gope *et al.* [17] (Enhanced)	✓	×	×	×	✓	✓	✓	✓
	Aysu *et al.* [37]	✓	✓	×	✓	✓	✓	✓	✓
	Proposed Protocol	✓	✓	✓	✓	✓	✓	✓	✓

Note: SP_1: Mutual authentication; SP_2: Session key security; SP_3: Anonymity and untraceability; SP_4: Mitigate desynchronization attacks; SP_5: Mitigate replay attacks; SP_6: Mitigate physical tampering or cloning attacks; SP_7: Mitigate impersonation attacks; SP_8: Mitigate man-in-the-middle attacks; ✓: Achieved; ×: Not Achieved.

60% reduction compared to the 640 bits of Saleem *et al.* [16]. As the deployment scale increases, the total storage overhead of the proposed protocol grows at $768n$ bits, which is lower than the $1024n$ bits of Gope *et al.* [17] (Normal) and equal to that of Lee *et al.* [15]. Although slightly higher than the $(640n+128)$ bits of Saleem *et al.* [16], this marginal difference is offset by the proposed protocol's enhanced security features (see Table V). In the noisy PUF setting, the proposed protocol requires 400 bits of storage per tag-39.02% less than the 656 bits of Gope *et al.* [17] (Enhanced), and its total system overhead of $1168n$ bits is also lower than the $1424n$ bits required by Gope *et al.* [17] (Enhanced). While its per-tag storage slightly exceeds the 256 bits of Aysu *et al.* [37], the proposed protocol offers a broader range of security capabilities (see Table V), which justifies this additional cost.

E. Security and Functionality Features Comparison

Table V compares the proposed protocol with existing protocols across eight security and functionality attributes: SP_1: Mutual authentication; SP_2: Session key security; SP_3: Anonymity and untraceability; SP_4: Mitigation of desynchronization attacks; SP_5: Mitigation of replay attacks; SP_6: Mitigation of physical tampering or cloning attacks; SP_7: Mitigation of impersonation attacks; and SP_8: Mitigation of man-in-the-middle attacks. In Table V, a check mark (✓) indicates that the feature is achieved, whereas a cross mark (×) denotes that it is not achieved.

In the ideal PUF setting, Lee *et al.* [15] fail to achieve mutual authentication (SP_1) due to the improper utilization of the PUF primitive. Gope *et al.* [17] (Normal) do not provide session key security (SP_2) or anonymity and untraceability (SP_3), and the absence of robust state and synchronization mechanisms leaves them vulnerable to desynchronization attacks (SP_4). In addition, the protocol proposed by Saleem *et al.* [16] lacks both mutual authentication (SP_1) and the ability to resist replay attacks (SP_5), which makes the protocol vulnerable to message replay attacks. In the scenario of noise PUF, the enhanced protocol of the integrated fuzzy feature extractor proposed by Gope *et al.* [17] has the same defects as the ordinary version proposed by them, failing to achieve session key security (SP_2), provide anonymity and untraceability (SP_3), and resistance to desynchronization attacks (SP_4). Moreover, although the protocol of Aysu *et al.* [37] implements most of the security properties, it can not guarantee anonymity and untraceability (SP_3), as pointed out by Gope *et al.* [17]. In contrast, only the proposed protocols achieve eight key security and functional properties in ideal and noisy PUF scenarios, thus providing comprehensive protection against the listed threats and ensuring strong practicality.

F. Critical Discussion

The proposed protocol achieves robust mutual authentication between RFID tags and readers by integrating arbiter-PUF-based hardware security mechanisms with lightweight cryptographic primitives, thereby maintaining efficiency while exhibiting strong resilience to common attack vectors. Furthermore, an enhanced protocol is introduced through the incorporation of a fuzzy extractor to mitigate the influence of

environmental noise on PUF outputs. It should be noted that unlike the work of Aysu *et al.* [37], the proposed protocols rely on libraries such as pypuf to simulate the arbiter PUF and fuzzy extraction process, and their performance evaluation is carried out on the Raspberry Pi platform with limited resources. In contrast, Aysu *et al.* implemented SRAM PUF and an interleaved BCH-based fuzzy extractor at the hardware level, addressing the key engineering challenges that the proposed protocols have not yet targeted. As part of the future work, we plan to implement arbitrator PUF and fuzzy extractor on actual low-cost RFID chips to further verify the feasibility of the protocols in the RFID system. At the same time, we will also work to improve the scalability of the protocols in large-scale IoT deployment, including the development of group authentication mechanisms that support multiple tag concurrent authentication.

VI. Conclusion

This paper proposes two lightweight anonymous AKA protocols for RFID systems with limited resources: one is based on the ideal PUF hypothesis, and the other is an enhanced protocol designed for the noise PUF environment. Both of these protocols take advantage of the uniqueness and unclonability of PUF to effectively mitigate a variety of security threats while protecting user privacy. Comprehensive security and functional analysis confirms that the proposed protocols meet the essential authentication and privacy protection requirements, and can effectively resist known attacks. Performance evaluation covers time, communication and storage overhead. The results show that both variants are superior to the state-of-the-art protocols of their kind, significantly reducing resource consumption while maintaining strong security. The inclusion of noise PUF variants further ensures its practicality in practical application scenarios, and can cope with factors such as noise, environmental changes and aging, so as to achieve secure deployment in various IoT and RFID applications.

References

[1] S. R. Khan, A. L. Bernassau, and M. P. Y. Desmulliez, "Passive and battery-free RFID-based wireless healthcare and medical devices: A review," *IEEE J. Radio Freq. Identif.*, vol. 8, pp. 724–742, Aug. 2024.

[2] Z. Ye, M. Yang, Y. Ren, C.-H. J. Hung, C.-T. M. Wu, and P.-Y. Chen, "Review on recent advances and applications of passive harmonic RFID systems," *IEEE J. Radio Freq. Identif.*, vol. 7, pp. 118–133, May 2023.

[3] X. Ai, A. Badshah, S. Tu, M. Waqas and I. Ahmad, "An improved ultra-lightweight anonymous authenticated key agreement protocol for wearable devices," *IEEE Trans. Mob. Comput.*, vol. 24, no. 5, pp. 4543–4557, May 2025.

[4] W. Chen, J. Childs, S. Ray, B. S. Lee, and T. Xia, "RFID technology study for traffic signage inventory management application," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17809–17818, Oct. 2022.

[5] K. Lin, H. Chen, Z. Li, N. Yan, H. Xue, and F. Xia, "Efficiently identifying unknown COTS RFID tags for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 987–997, Jan. 2024.

[6] H. A. Abdulghani, N. A. Nijdam, and D. Konstantas, "Analysis on security and privacy guidelines: RFID-based IoT applications," *IEEE Access*, vol. 10, pp. 131528–131554, Dec. 2022.

[7] D.-Z. Sun and Y. Mu, "Security of grouping-proof authentication protocol for distributed RFID systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 2, pp. 254–257, Apr. 2018.

[8] B.-Q. Zhao, H.-M. Wang, and P. Liu, "Safeguarding RFID wireless communication against proactive eavesdropping," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11587–11600, Dec. 2020.

[9] T.-Y. Wu, L. Wang, and C.-M. Chen, "Enhancing the security: A lightweight authentication and key agreement protocol for smart medical services in the IoHT," *Mathematics*, vol. 11, no. 17, Art. no. 3701, Aug. 2023.

[10] D. Chaudhary, T. Soni, K. L. Vasudev, and K. Saleem, "A modified lightweight authenticated key agreement protocol for Internet of Drones," *Internet Things*, vol. 21, Art. no. 100669, Apr. 2023.

[11] Z. Chen, H. Jiang, J. You, X. Wang, and P. Z. H. Sun, "RFID lightweight authentication mechanism for smart factories based on blockchain," *IEEE J. Radio Freq. Identif.*, vol. 8, pp. 19–30, Jan. 2024.

[12] A. Q. A. Alhasan, M. F. Rohani, and M. S. Abu-Ali, "Ultra-lightweight mutual authentication protocol to prevent replay attacks for low-cost RFID tags," *IEEE Access*, vol. 12, pp. 50925–50934, Apr. 2024.

[13] M. Shariq, K. Singh, C. Lal, M. Conti, and T. Khan, "ESRAS: An efficient and secure ultra-lightweight RFID authentication scheme for low-cost tags," *Comput. Netw.*, vol. 217, Art. no. 109360, Nov. 2022.

[14] X. Wang, K. Fan, K. Yang, X. Cheng, Q. Dong, H. Li, and Y. Yang, "A new RFID ultra-lightweight authentication protocol for medical privacy protection in smart living," *Comput. Commun.*, vol. 186, pp. 121–132, Mar. 2022.

[15] T.-F. Lee, K.-W. Lin, Y.-P. Hsieh, and K.-C. Lee, "Lightweight cloud computing-based RFID authentication protocols using PUF for e-healthcare systems," *IEEE Sens. J.*, vol. 23, no. 6, pp. 6338–6349, Mar. 2023.

[16] M. A. Saleem, X. Li, K. Mahmood, T. Tariq, M. J. F. Alenazi, A. K. Das, 'Secure RFID-assisted authentication protocol for vehicular cloud computing environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 12528–12537, Sep. 2024.

[17] P. Gope, J. Lee, T. Q. S. Quek, 'Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2831–2843, Nov. 2018.

[18] B. K. Meher, R. Amin, A. K. Das, and M. K. Khan, "KL-RAP: An efficient key-less RFID authentication protocol based on ECDLP for consumer warehouse management system," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3411–3420, Sep.-Oct. 2022.

[19] M. Tanveer, A. U. Khan, N. Kumar, A. Naushad, and S. A. Chaudhry, "A robust access control protocol for the smart grid systems," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6855–6865, May 2022.

[20] Y. Wu, H. Guo, Y. Han, S. Li, and J. Liu, "A security-enhanced authentication and key agreement protocol in smart grid," *IEEE Trans. Ind. Informat.*, vol. 20, no. 9, pp. 11449–11457, Sep. 2024.

[21] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2649–2656, Feb. 2022.

[22] L. Zhang, J. Xu, M. S. Obaidat, X. Li, and P. Vijayakumar, "A PUF-based lightweight authentication and key agreement protocol for smart UAV networks," *IET Commun.*, vol. 16, no. 10, pp. 1142–1159, 2022.

[23] C. Tian, J. Ma, T. Li, J. Zhang, C. Ma, and N. Xi, "Provably and physically secure UAV-assisted authentication protocol for IoT devices in unattended settings," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 4448–4463, Mar. 2024.

[24] S. Kumar, H. Banka, and B. Kaushik, "Ultra-lightweight blockchain-enabled RFID authentication protocol for supply chain in the domain of 5G mobile edge computing," *Wireless Netw.*, vol. 29, no. 5, pp. 2105–2126, Feb. 2023.

[25] N. Egide and F. Li, "HAP-SG: Heterogeneous authentication protocol for smart grid," *Peer-to-Peer Netw. Appl.*, vol. 16, no. 3, pp. 1365–1379, May 2023.

[26] Q. Cheng, Y. Ma, F. Wei, and X. Li, "An efficient anonymous certificateless authentication and key agreement scheme for smart grids," *Comput. Electr. Eng.*, vol. 124, Art. no. 110369, May 2025.

[27] X. Zhou, S. Wang, K. Wen, B. Hu, X. Tan, and Q. Xie, "Security-enhanced lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9599–9609, Mar. 2024.

[28] C. Lee, M. Oh, D. Kwon, Y. Park, and Y. Park, "PLAKA-MD: PUF-based lightweight authentication and key agreement scheme for medical devices in IoHT," *IEEE Internet Things J.*, vol. 12, no. 12, pp. 19338–19353, Jun. 2025.

[29] J. Subramani, A. Maria, A. Sivaraman, P. Vijayakumar, F. Alqahtani, and A. Tolba, "An efficient anonymous authentication scheme for blockchain-assisted and fog-enabled smart grid," *Comput. Electr. Eng.*, vol. 119, Art. no. 109508, Oct. 2024.

[30] K. Mahmood, Z. Ghaffar, M. Farooq, K. Yahya, A. K. Das, and S. A. Chaudhry, "A security enhanced chaotic-map-based authentication

protocol for Internet of Drones," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 22301–22309, Jun. 2024.

[31] J. Sharma and P. S. Mehra, "HCFAIUN: A novel hyperelliptic curve and fuzzy extractor-based authentication for secure data transmission in IoT-based UAV networks," *Veh. Commun.*, vol. 49, p. 100834, Oct. 2024.

[32] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[33] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Advances in Cryptology (EUROCRYPT)*, Berlin, Germany, Apr. 2002, pp. 337–351.

[34] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. Public Key Cryptography (PKC)*, Berlin, Germany, 2005, pp. 65–84.

[35] A. Armando, *et. al.*, "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proc. Computer Aided Verification*, Berlin, Germany, 2005, pp. 281–285.

[36] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. APPSEM Workshop*, Tallinn, Estonia, 2005, pp. 1–17.

[37] A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung, "End-to-end design of a PUF-based privacy preserving authentication protocol," in *Proc. Cryptographic Hardware and Embedded Systems (CHES)*, Berlin, Germany, Sep. 2015, pp. 556–576.