Mathematical Modelling of Engineering Problems
Vol. 11, No. 12, December, 2024, pp. 3465-3477
Journal homepage: http://iieta.org/journals/mmep

An Enhanced Ultra-Lightweight Mutual Authentication Protocol for RFID: Securing Against Vulnerabilities with Optimized Performance

Ahmed Qasim Abd Alhasan[1,2*], Mohd Foad Rohani[1], Oras N. Hamad[3]

[1] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
[2] Department of Computer and Information Technology, Missan Oil Company (MOC), Maysan 62001, Iraq
[3] Department of Anatomy, Faculty of Medicine, University of Misan, Maysan 62001, Iraq

Corresponding Author Email: aalcorejy@gmail.com

https://doi.org/10.18280/mmep.111225

Received: 8 August 2024
Revised: 13 October 2024
Accepted: 20 October 2024
Available online: 31 December 2024

Keywords:
radio frequency identification, ultra-lightweight mutual authentication protocols, security attack, cryptanalysis

ABSTRACT

This paper reviews ultra-lightweight mutual authentication protocols (UMAPs) tailored for passive radio frequency identification (RFID) tags, which face limitations in computational power and storage capacity, rendering traditional cryptographic methods inadequate. We identify critical security vulnerabilities in existing UMAPs, including replay, desynchronization, full disclosure, and denial of service attacks. In response, we propose an innovative UMAP that enhances session key management and updates session data to maintain high authenticity while minimizing execution time and memory usage. Our methodology employs T-functions and bitwise operations (AND, OR, XOR, and rotation) to develop a UMAP suitable for RFID applications. Comprehensive security analyses conducted using formal verification tools (Scyther and AVISPA) confirm that the proposed protocol effectively mitigates identified security threats. Performance evaluations reveal that our protocol achieves a total execution time of 0.033 milliseconds, significantly lower than that of existing protocols while demonstrating reduced overhead. Furthermore, we provide a comparative analysis with recent protocols, highlighting that our proposed protocol requires less storage space and exhibits ultra-lightweight cryptographic demands.

1. INTRODUCTION

Using wireless methods, radio frequency identification (RFID) is an identifying technology that allows for object tracking and identification. RFID technology is now widely employed in a variety of fields, including security and healthcare. Governmental organizations and businesses were encouraged to use this technology into their applications due to its affordability and versatility.

An RFID system is made up of three primary parts: an RFID tags, an RFID reader, and a back-end server. An antenna is used to communicate between the RFID tags and readers, and integrated circuits (IC) is used for storage and computing. The RF module, control unit, and antenna are all part of the RFID reader. The reader or transceiver's job is to give the tag the energy it needs to function as well as the communication signals it needs to carry out particular tasks. The back-end server can handle hundreds of RFID tag queries since it has fast processing power and ample storage [1].

There are three categories for RFID tags: semi-active, active, and inactive. The semi-active tags have their own battery, but it is only used for the computing function; the active tags have their own battery, which is used for both transmission and computation. The passive tags, also known as low-cost RFID tags, do not have their own battery [2].

Due to their limited computational power and storage capacity, low-cost RFID tags are unable to carry out traditional security cryptography tasks. The computational capabilities of low-cost RFID tags are surpassed by the exorbitant power and memory needs of classic encryption algorithms and primitives. In order to improve the security of the RFID system with these obstacles, most researchers are tackling these issues by introducing UMAPs that simply employ basic bitwise operations.

1.1 RFID challenges

Among all automated identification systems (AIDs), RFID is the most widely used technology and maybe the most ubiquitous in history. Similar to other kinds of systems, RFID systems provide a number of dangers that make this technology very vulnerable to security breaches. A few elements of security threats are as follows:

(1) **Wireless channel:** The RFID tags and readers' usage of radio waves as a communication channel leaves them open to eavesdropping and other types of malicious attack. As a result, encryption is used as a countermeasure to protect the information.
(2) **Bidirectional communication:** The communication mechanism in RFID technology is bidirectional; RFID tags and readers are two parties that communicate with one another, and a certain protocol predetermines the

communication channel. In contrast to other optical identification systems that use a specific identification sign, such as a barcode or QR code, etc., the protocol guarantees the secrecy and anonymity of the transactions. Furthermore, as noted by the previous studies [3, 4], there are numerous advantages of employing RFID technology in various sectors. Unauthorized readers have the ability to interact with RFID tags and retrieve information. To prevent assaults, researchers and developers provide mutual authentication mechanisms, allowing only authorized RFID readers to communicate with a specific tag.

(3) **Resources limitation:** As previously indicated, RFID technology has become extensively utilized in recent times since it is less expensive than other automatic identifying technologies. As a result, RFID has a number of security problems. Therefore, it is essential to suggest secure, lightweight, or extremely lightweight alternatives for RFID technology.

(4) **Dense environment:** There is a chance of a collision occurring when many RFIDs are transmitting signals to the reader concurrently on the same frequency. A dense reader environment results from the presence of several RFID readers in one area. The likelihood of a collision increases in this type of setting. The dense reader environment weakens system performance by reducing network throughput and increasing the likelihood of tag identification errors.

(5) **RFID attacks:** Although the data exchanged between the reader and the tag is safe in most RFID systems, attacks can still occur. To prevent these attacks, the reader and tag authenticate one another using secret keys. It is possible for attackers to target RFID systems. Numerous attack types, including Desynchronization, Replay, Full-Disclosure, Denial of Service (DoS), and Eavesdropping attacks, can target RFID devices [5].

Since each of those difficulties raises a different kind of security risk, safe RFID authentication procedures must be implemented utilizing cryptography. In the past ten years, more than a thousand RFID authentication protocols have been presented [6].

1.2 Classification of UMAPs

The procedure of authentication involves an RFID tag demonstrating, through some indicator such as tag ID and secret keys, that it is an alleged identity for the RFID reader and back-end server. Physical solutions cannot facilitate this procedure; only RFID protocols may be used. There are two stages to the authentication process for RFID systems. In order to ensure that they are communicating with a valid partner, the RFID tags and RFID readers must first authenticate themselves before starting any connection. In order to verify the authenticity of the data received, the RFID tags and RFID readers exchange data during the second phase [7].

Two parties can verify each other's identities using mutual authentication. When the RFID reader and the RFID tag are both verifying each other, this happens. Before sharing any data or keys, the RFID tag and the RFID reader should conduct mutual authentication.

Cryptographic techniques must be used by RFID systems in order to establish safe RFID authentication protocols. RFID tags are not able to perform classical security cryptography operations like stream ciphers, AES, and hash functions due to their low processing capability and storage capacity [8].

Luo et al. [9] categorized RFID authentication protocols into four types according on the tags' capabilities:

(1) **Full-fledged:** pertains to protocols that need to be supported by traditional cryptographic features, such as symmetric encryption [8, 10].

(2) **Simple:** for protocols that enable random number generators on the tag and one-way hashing methods.

(3) **Light-weight:** for protocols that do not allow hashing functions but do support simple functions like the Cyclic Redundancy Code checksum (CRC) and random numbers generator [2, 9, 11].

(4) **Ultra-lightweight:** describes protocols that don't use anything more complicated than basic bitwise operations (like OR, AND, XOR, etc.) on tag [1, 12, 13].

In spite of the limitations of passive RFID tags (low-cost tags), the hash functions have been used by most of the proposed protocols [14-16], therefore, engineers are confronted by non-trivial problem when trying to implement cryptographic hash functions with only 250-4K gates [17, 18].

Because of their extremely limited capabilities and the extensive use that is anticipated of them, ultra-lightweight protocols provide a challenging security problem. Because using an RFID tag that uses a protocol based on basic bitwise operations to safeguard sensitive data, such as credit card numbers, health information, or e-passport information, may be difficult. This problem has drawn attention from researchers, and as a result, numerous novel ultra-lightweight techniques are put forth annually. The current Ultra-Lightweight protocol is insufficiently secured, despite the fact that numerous additional protocols are being developed [19].

1.3 Problem statement

The widespread of radio frequency identification (RFID) systems is significantly hindered by serious security and privacy vulnerabilities, especially in low-cost RFID tags. Traditional security mechanisms often unsuitable for low-cost RFID tags due to limited processing power and memory capacity. As attackers exploit these weaknesses, there is an urgent need for effective ultra-lightweight mutual authentication protocols that can operate within such limitations. Current ultra-lightweight mutual authentication protocols, specifically designed for resource-constrained environments, demonstrate considerable shortcomings. These include vulnerabilities to desynchronization, full-disclosure attacks, and inadequate defenses against tracking and impersonation. These flaws stem from reliance on weak cryptography functions, inefficient use of random nonces, and lack of robust alternative methods and consequently, these protocols many of them fail to provide effective mutual trust or provide excessive demands on RFID tags, compromising their feasibility in real-world applications.

1.4 Contributions

This article's main contributions are as follows:

- **Security analysis of existing UMAPs:** In this paper, we conducted a comprehensive review of existing UMAPs, to identify their weaknesses and vulnerabilities. Our analysis emphasized issues such as de-synchronization attacks, which undermine the reliability and security of these protocols. By

examining the structural and operational aspects of current UMAPs, we highlighted the critical areas where these protocols fail to provide robust protection, thereby paving the way for the development of more secure solutions.

- **Mutual authentication:** The proposed protocol ensures mutual authentication between legitimate entities, including the back-end server, RFID readers, and RFID tags, by utilizing indicators such as tag IDs and secret keys.
- **Privacy:** The proposed protocol ensures anonymity for RFID tags, keeping their data secure from unreliable third parties during transmission. It also guarantees untraceability by avoiding static or linked data that could allow tracking across sessions.
- **Security:** The proposed protocol is designed to resist desynchronization attacks, addressing vulnerabilities in the wireless communication channel between RFID readers and tags.
- **Performance:** The proposed protocol is optimized for RFID tags by minimizing storage space requirements, employing ultra-lightweight cryptographic functions, and reducing the number of communication messages needed for authentication.

The rest of the paper is structured as follows: In Section 1, we explain the most important challenges that is faced by this technology and summarization of the classification of authentication protocols based on the ability of tags is provided. In Section 2, the security analysis of several RFID UMAPs is presented. Section 3 describes the design of a new ultralight-weight mutual authentication protocol. The assumptions that we assumed, the new protocol's features, the notation used, and the description of the proposed protocol are explained. Section 4 describes the security analysis for the proposed protocol in terms of formal and informal analysis to prove the authenticity, privacy, and secrecy between the RFID tags and the back-end server by using official tools Scythe and AVISPA. Performance analysis of the proposed protocol depending on the storage cost, communication messages, and resistance to de- synchronization attacks is presented in Section 5. Section 6 describes the conclusion of this paper.

2. SECURITY ANALYSIS OF EXISTING UMAPS

In the past ten years, numerous RFID UMAPs have been presented [6]. Regretfully, the majority of the suggested protocols are open to multiple attacks [20]. We examine the RFID ultra-lightweight mutual authentication methods in depth in the following manner:

It was Lopez that designed the UMAPs basics in 2006. Three types of UMAPs were postulated by Peris-Lopez et al. [21-23], specifically M2AP [21], EMAP [22], and LMAP [23]. The three protocols that were designed made use of the triangular functions (AND, XOR, and OR). Each of these protocols has a computational cost of less than 300 gates. In 2007, the Lopez protocols underwent cryptanalysis. The researchers used the T-function's vulnerability to carry out a variety of attacks, including replay, full disclosure, and desynchronization assaults. The inability of the Lopez measures to stop these kinds of attacks was demonstrated by studies [24, 25].

Chien [8] proposed the SASI protocol in 2007. Rot(x, y), a left rotation function, was the non-T-function that Chien employed. As reported by Chien [8], the rotation function is described along with the prerequisites needed to put it into practice. The SASI protocol's cryptanalysis was first presented by Hernandez-Castro et al. [26] in 2008, and it was later provided by some researchers [27-29]. The rotation function employed by the SASI protocol has a flaw that the researchers intended to exploit in order to present attacks related to traceability, complete disclosure, and desynchronization.

A new UMAP (LMAP+) was proposed by Li in 2008 [30]. Li [30] achieved safe mutual authentication between the RFID tag and RFID reader by using only bitwise operations (T-function). Li [30] demonstrated how the suggested procedure prevents forgeries and is resistant to Man in the Middle attacks. Based on computational complexity, storage, and communication, the author compared the suggested protocol with the LMAP protocol, as indicated in Table 1. Safkhani et al. [31] suggested the first attack on the LMAP+ protocol. Safkhani demonstrates that LMAP+ fails to meet the security requirement of traceability and that the protocol is unable to thwart a desynchronization attack.

Subsequently, protocols were presented by Peris-Lopez et al. [32] and David and Prasad [33] that involved the development of a single non-T-function in order to enhance the earlier protocols. These protocols' susceptibilities to several types of assaults were revealed via cryptanalysis [34, 35].

Yeh et al. [34] presented a redesigned UMAP in 2010. This protocol was proposed by implemented a T-function, Rot function, and random number. The authors demonstrated that every passive assault can be resisted by this technique. These protocols' security study revealed that they are vulnerable to a number of different types of attacks.

Engels et al. [36] proposed the Hummingbird-1 protocol in 2011, using an extremely lightweight cryptography method to introduce a new UMAP. Further details on Hummingbird-1 are presented in the reference [36]. Subsequently, Saarinen [37] demonstrated that for some cryptographic applications, the Hummingbird-1 protocol might not provide sufficient security. This protocol, which was an improved version of Hummingbird-1, was suggested as Hummingbird-2. The Hummingbird-2 cryptographic algorithm was utilized in the protocol's construction, and its designers have demonstrated that it can withstand prevalent attacks aimed at jeopardizing the security and privacy of RFID systems. Zhang et al. [38] identified a few hummingbird-2 protocol flaws. The results of the cryptanalysis indicate that the Hummingbird-2 encryption is vulnerable to related key attacks.

The UMAPs were enhanced by employing several non-T-functions after 2011. One of the RFID UMAPs that was created by utilizing the permutation function is RAPP [39]. This protocol allows for mutual authentication between RFID readers and RFID tags by using rotation Rot(x, y) and permutation Per(x, y). However, this also increases the protocol execution time and memory requirements. The RAPP has been the target of full disclosure and desynchronization attacks by some researchers [40, 41] who took advantage of the permutation function's flaws.

For a concise and lightweight authentication protocol (SLAP), Luo et al. [9] developed a new ultra-lightweight primitive in 2016. They called it the conversion function Conv(x, y). Safkhani and Bagheri [42] presented a security study [9] and showed that a desynchronization attack could be carried out in just five authentication sessions between the RFID tags and the RFID readers.

Table 1. LMAP and LMAP+ comparison

Protocols	Computational Overhead (Tag: Reader)	Storage Overhead (Tag: Reader)	Communication Overhead (Bits)
LMAP [23]	+,∨,∧,⊕	480: 6NL 385: 5NL	384 288
LMAP+[30]	+,⊕		

Several cryptanalysis studies, including those found in references [12, 43-45], have been proposed in 2017 to demonstrate or present a number of security breaches against the earlier UMAPs. Zhuang et al. [20] introduced de-synchronization and replay attacks aimed at ultra-lightweight mutual authentication protocols. His findings demonstrated that the de-synchronization attack put out in this research is insurmountable for all UMAPs based on T-functions.

Based on the review, we can infer that several UMAPs have been developed in the last decade; nevertheless, cryptanalysis studies have demonstrated that all of these procedures are susceptible to various types of attacks, as shown in Table 2 which summarizes the advances in the proposed UMAPs and detected attacks.

Table 2. Current research on UMAPs

Year	Protocols	Year	Attacks
2006	M2AP [21], EMAP [22] and LMAP [23]	2007	[24, 25]
2007	(SASI) protocol [8]	2008 2009 2010 2011	[26] [27] [29] [28]
2008	Gossamer protocol [32] LMAP+ protocol [30]	2010 2011	[35] [31]
2010	Yeh et al. [34]	2010	[35]
2011	Hummingbird-1 [36]	2011	[37]
2012	RAPP protocol [39]	2013	[46]
2013	RAPLT protocol [46]	2015	[47]
2016	SLAP [9]	2016	[42]
2018	Zhuang et al. [20] showed that all mentioned ultralight-weight RFID protocols that used redundancy mechanism by store old keys on both side and one side will not be prevent of the proposed replay attacks and desynchronization attack.		

Based on the operators employed in ultra-lightweight RFID authentication protocols, we categorize these protocols into three classes, highlighting their respective limitations and vulnerabilities as follows:

(1) T-function
- **Operators:** Commonly includes bitwise operators such as AND, OR, and XOR.
- **Weakness:** In T-functions, not all output bits are influenced by all input bits, which poses significant concerns in UMAPs.

(2) Non-T-function
- **Operators:** Utilizes rotation operators, such as Left Rot and Circular Rot functions.
- **Weakness:** The rotated bits remain unchanged, leading to a limited set of possible outputs (L possibilities). This limitation compromises untraceability, even under passive attack scenarios.

(3) Multiple Non-T-functions
- **Operators:** Incorporates bitwise XOR, left rotation, permutation, conversion functions, and merge and separation operations.
- **Weaknesses:** Besides existing vulnerabilities, the use of multiple non-T-functions increases both protocol execution time and memory requirements.

3. PROPOSED PROTOCOL

The new ultralight-weight mutual authentication protocol (UMAP) designed and developed by implementing the rotation operator, the mechanism of the secret keys, and T-function. In this section, we explain the proposed protocol in detail.

3.1 Assumptions

Based on the following assumptions, we present an ultralight-weight RFID mutual authentication protocol:
- We assumed that the communication channel between the RFID readers and the server is secured.
- The communication channels between the RFID tags and RFID readers are wireless which are susceptible to active attacks.
- Each RFID tag shares secrets including secret keys, pre-shares a pseudonym (IDS) and static identification (ID) with a server.
- The back-end server has database to holds the all-RFID tag's details including the secret keys, pre-shares a pseudonym (IDS) and static identification (ID).
- The RFID tag's memory type is a Flash memory or EEPROM memory to make the RFID tag's data can be updated.

3.2 Protocol design

There are several features that our proposed protocol possesses as following:
- In our protocol, the implementation of all costly computing operations is executed by the RFID reader. The RFID reader has enough computing resources.
- the RFID reader generates the required random number without needs to do it in tag's side, only RFID tag need to do a very simple bitwise operation like (OR bitwise, AND bitwise and XOR bitwise) and Rot (a, b) left rotate the value of a with the bits of b.
- In order to resist the possible desynchronization attack by this proposed protocol, each tag and back-end server keeps two entries of secret keys and IDS: old values from the last session and the other are new for the potential next session.
- The length of each of secret keys, pre-shares a pseudonym (IDS) and static identification (ID) is 96 bits.
- Any update or store to the RFID tag's data on the RFID reader's side is not required.

Before starting any tag readings, the back-end server designates the initial values in the database, and in the RFID, tag as summarized below:
- The back-end server assigns (*IDS, SID, K1, K2*) for each tag, then set the values for the record in the tag (IDS_{new}=*IDS,* $K1_{new}$=*K1,* $K2_{new}$=*K2, SID=SID*).

3.3 Protocol description

The proposed protocol consists of three phases: the

Identification Phase, the Server Authentication and Update Phase, and the Tag Authentication and Update Phase. An overview of the protocol is illustrated in Figure 1, and it is executed according to the steps depicted in Figure 1.

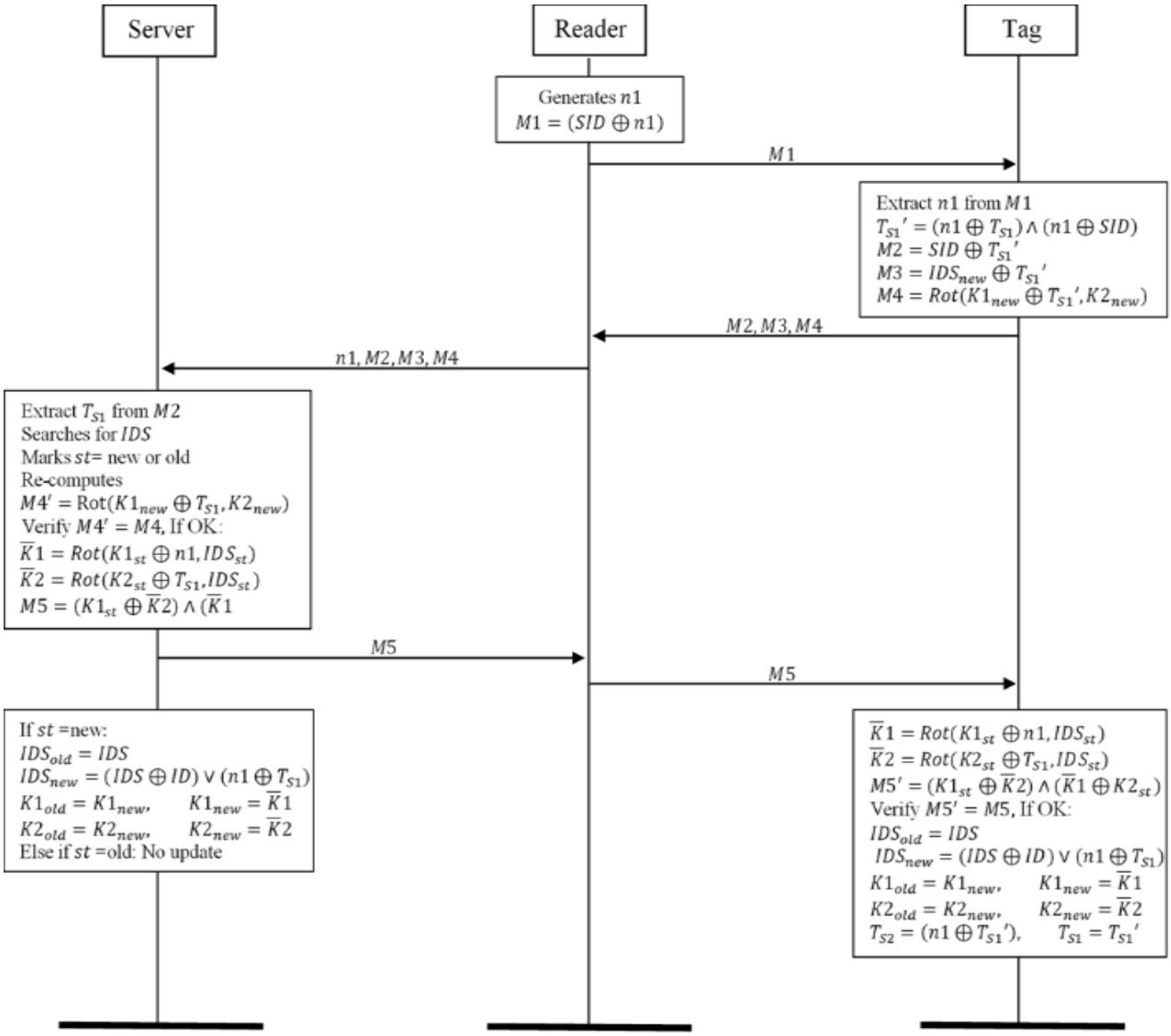

Figure 1. The proposed Ultralight-weight RFID mutual authentication protocol

Identification phase

- **Step 1.** Reader: The authentication protocol starts with the RFID reader's side by generates random number *n*1 and computes message *M*1 then send *M*1 to the RFID tag.

$$M1 = (SD \oplus n1) \quad (1)$$

- **Step 2.** Tag: From the message *M*1, the tag will obtain the random number *n*1 and computes two messages as follows:

$$T_{S1}' = (n1 \oplus T_{S1}) \wedge (n1 \oplus SID) \quad (2)$$

$$M2 = (SID \oplus T_{S1}') \quad (3)$$

$$M3 = (IDS_{new} \oplus T_{S1}') \quad (4)$$

$$M4 = Rot(K1_{new} \oplus T_{S1}', K2_{new}) \quad (5)$$

And sends *M2, M3*, and *M*4 to the RFID reader.

- **Step 3.** Reader: After received *M2, M3* and *M4* the reader forwards these messages to the server.

Server authentication and update phase

- **Step 1.** The server XORs its local value *SID* with the received *M2* to get T_{S1}.
- **Step 2.** For all the stored IDS, the server computes *M3′* = IDS⊕TS1 until it finds the matching with received *M*3:
  - If $IDS=IDS_{new}$ then $st=new$, (IDS_{new}, $K1_{new}$, $K2_{new}$) and re-computes $M4'=Rot$ ($K1_{new} \oplus T_{S1}$, $K2_{new}$) in order authenticate the RFID tag.
  - If $IDS=IDS_{old}$ then $st=old$, (IDS_{old}, $K1_{old}$, $K2_{old}$) and re-computes $M4'=Rot(K1_{old} \oplus T_{S1}, K2_{old})$ in order authenticate the RFID tag.
  - Else if the server doesn't find any match with *IDS* in the database or *M4′*, *M*4, then the server sends an ending notification to the reader in order to terminate this session.
- **Step 3.** The server computes *M*5 as follows:

$$K1' = Rot(K1_{st} \oplus n1, IDS_{st}) \quad (6)$$

$$K2' = Rot(K2_{st} \oplus T_{s1}, IDS_{st}) \quad (7)$$

$$M5 = Rot(K1_{st} \oplus K2') \wedge (K1' \oplus K2_{st}) \quad (8)$$

- **Step 4.** The server transmits *M5* to the reader, in turn sends *M5* to the tag.
- **Step 5.** The server's data updates as follows:
  - If *st=new*

$$IDS_{old} = IDS_{new} \quad (9)$$

$$IDS_{new} = (IDS \oplus ID) \vee (n1 \oplus T_{s1}) \quad (10)$$

$$K1_{OLD} = K1_{new}, K1_{new} = K1' \quad (11)$$

$$K2_{OLD} = K2_{new}, K2_{new} = K2' \quad (12)$$

  - Else if *st=old* where the *IDS* is found in IDS_{old}: No update

Tag authentication and update phase

- **Step 1.** After received *M5* the RFID tag checks if received *M5=M5′*

$$K1' = Rot(K1_{new} \oplus n1, IDS_{new}) \quad (13)$$

$$K2' = Rot(K2_{new} \oplus T_{s1}, IDS_{new}) \quad (14)$$

$$M5' = (K1_{new} \oplus K2') \wedge (K1' \oplus K2_{new}) \quad (15)$$

- **Step 2.** If the RFID tag find matching between received *M5* and computed *M5′*, then the RFID tag authenticated the back-end server and updates its values to:

$$IDS_{new} = (IDS \oplus ID) \vee (n1 \oplus T_{s1}) \quad (16)$$

$$K1_{new} = K1', K2_{new} = K2', T_{S1} = T_{S1}' \quad (17)$$

Else if *M5* is not received, or the computed *M5′* is not equal to the received *M5*, the RFID tag neglects all previous steps and keeps all current values unchanged.

4. PROTOCOL ANALYSIS

This section examines how our proposed UMAP aligns with the objectives of this study by conducting both formal and informal security analyses. Through these analyses, we aim to provide a comprehensive evaluation of the protocol's security features, demonstrating its resilience against various threats and vulnerabilities. By employing rigorous methodologies, we intend to substantiate our claims regarding the effectiveness of the proposed UMAP in ensuring secure and efficient communications in relevant applications. Integrating both formal verification and practical security assessments will enhance our understanding of the protocol's robustness and capacity to address the challenges identified in the literature.

4.1 Informal analysis

- **Mutual authentication:** In our proposed protocol the RFID tag and server can authenticate each other because only a legitimate server and tags who has the keys *K1, K2*, and *SID*. By these secret keys and *SID*, the only legitimate server and tags can generate such messages that will be accepted between them and preventing others from creating and recovering any valid messages. The calculations of the freshness of these messages are ensured by involving share secret keys, server random number, tag's local secret key, *IDS, SID*, and potential next keys.
  - **Authentication of the tag:** The back-end server can authenticate a tag by check tag's messages *M3* and *M4* to check whether there are matching *IDS* and the *K1, K2* values. *M1* and *M2* are messages that are computed by using secret keys in addition to the random number *n1* and T_{SI} that only known by the legitimate back-end server and tag. Therefore, only the rightful back-end server can verify the legitimacy of the messages. The tag's authenticity can determine by the correctness of messages *M3* and *M4*.
  - **Authentication of the server:** Once the server successfully authenticates the RFID tag, the back-end server computes a message M5 and transmits to the RFID tag. After received of back-end server's message the RFID tag authenticates the server by using it is local secret keys to checking the correctness and legitimacy of received messages.
- **Tag content privacy:** In RFID systems for each RFID tag, there is a static unique identity *ID*. During the transmission session, the *ID* should be transmitted confidentially. In our proposed protocol, the back-end server and tags are using local and shared secrets keys in addition to random numbers *n1* and T_{SI} in order to compute an internal secret keys $K1_{new}$ and $K2_{new}$. The proposed protocol by using the tuple of (*IDS, SID, n1,* T_{SI}, $K1_{new}$ and $K2_{new}$) computes several messages to confidentially transmitting of *ID*. In each session, the attacker will obtain new messages in every time eavesdropping this session because in the proposed protocol, the tag's responses are changed with new updated values and fresh random numbers and if the previous session the mutual authentication had been failed, and the tag's data not changed then the *M1-M5* messages will change due to the existence of random numbers (*n1* and T_{SI}) generated by the reader and tag respectively.
- **Prevent de-synchronization attacks:** Due to the wireless communication between RFID tags and readers, the channel is susceptible to eavesdropping and various attacks. A common vulnerability is the desynchronization attack, which occurs when an attacker successfully disrupts the mutual authentication process between the RFID reader and the tag. To address this challenge, we implement a mechanism to resist desynchronization attacks. Specifically, during each session, the back-end server will update the tag's data only if it identifies a match among the values of (IDS_{new}, $K1_{new}$, $K2_{new}$). Conversely, if the server detects a match with the values of (IDS_{old}, $K1_{old}$, $K2_{old}$) or if it fails to find a match with (IDS_{old}, $K1_{old}$, $K2_{old}$), the server will refrain from updating its data. This approach ensures the integrity and synchronization of the authentication process.
- **Prevent replay attacks:** The use of r $n1$ and T_{SI} in the

message generation process ensures that old messages cannot be reused successfully. Each session generates unique values, making it impossible for an attacker to replay intercepted messages without detection.

- **Prevent full-disclosure attacks:** By utilizing secret keys and pseudonyms, the protocol ensures that even if an attacker intercepts message exchanges, they cannot derive the identities or keys involved. The dynamic nature of message generation fortifies this protection, as each session produces distinct messages.
- **Protection against traceability:** The use of pseudonyms *IDS* and *SID* adds a layer of anonymity, making it difficult for attackers to trace the identities of RFID tags through message analysis. The frequent updating of these values further complicates any attempts at tracing.
- **Protection against DoS attacks:** The protocol's verification steps before data updates reduce the risk of successful DoS attacks. If an attacker attempts to disrupt communication by sending invalid messages, the server will not proceed with updates, thus preserving the system's functionality.

The robust security provided by our proposed protocol, particularly its resilience against various attacks, surpasses that of the other protocols illustrated in Table 3.

Table 3. Security threats in current UMAPs

Protocol	Attacks				
	De-Synchronization	Replay	Full-Disclosure	Traceability	DoS
[8]	✗	✓	✗	✓	✗
[9]	✗	✗	✗	✓	✓
[21]	✗	✓	✗	✗	✓
[22]	✗	✗	✓	✓	✗
[32]	✗	✗	✗	✓	✓
[39]	✗	✗	✓	✓	✗
[46]	✗	✓	✓	✓	✗
[47]	✗	✓	✓	✓	✗
[48]	✗	✗	✓	✓	✗
Proposed protocol	✓	✓	✓	✓	✓

4.2 Formal analysis

In this section, we formally analyze our proposed protocol and prove the authenticity and secrecy between the RFID tags and the back-end server are achieved, for this purpose the Scyther [49] and AVISPA [50] tools are used.

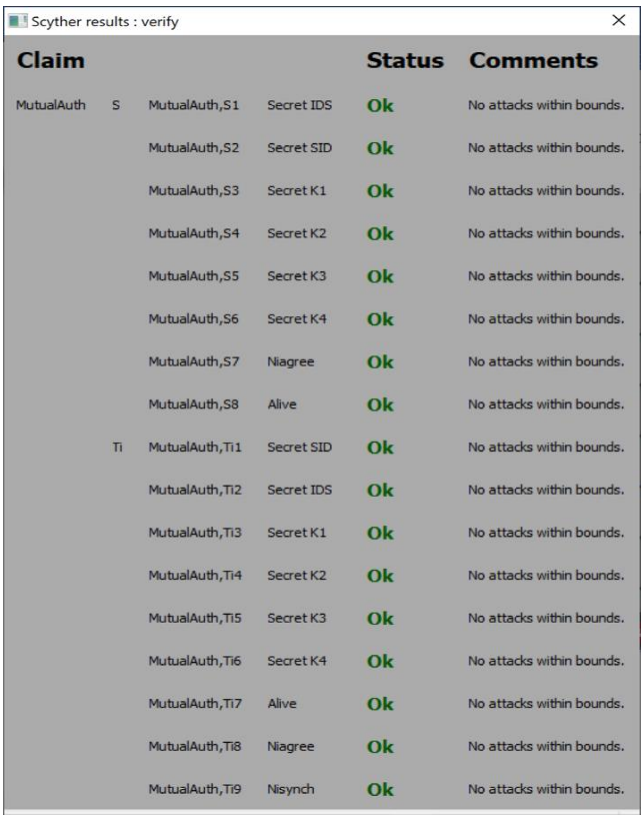

Figure 2. The Scyther analysis results

4.2.1 Scyther analysis

One of the most important formal security analysis tools to check the authenticity of the transmitted messages between the back-end server and tags in the protocols is Scyther. It employs a model-checking approach to analyze the correctness of protocols against specific security properties. Scyther allows users to specify security properties that need verification, such as secrecy (ensuring that certain information is not disclosed) and authenticity (ensuring that messages are genuinely from the claimed sender). The tool systematically searches for attacks against the specified security properties. It looks for scenarios where an attacker could intercept, modify, or inject messages to violate the protocol's security guarantees. We used Scyther to perform a formal security analysis of our proposed protocol based on the Dolev and Yao model [51], which is suitable for analyzing security protocols with an unbounded number of instances. We made the formal analysis of the proposed protocol depending on three goals secret, aliveness, and agreement. There are three roles defined, namely as Server (S), Reader (R), and Tag (Ti). N1 and N2 are random numbers defined as Nonce; and IDS (tag identifier), SID (server identifier) and K1, K2, K4, K4 (tag shares keys) are defined as Data. The XoR is defined as global functions.

The role Server and role Tag are sharing a secret goal through the six values of shares secret keys: IDS, SID, K1, K2, K3 and K4 as follows:

claim_S1(S, Secret, IDS);
claim_S2(S, Secret, SID);
claim_S3(S, Secret, K1);
claim_S4(S, Secret, K2);
claim_S5(S, Secret, K3);
claim_S6(S, Secret, K4);
claim_Ti1(Ti, Secret, SID);
claim_Ti2(Ti, Secret, IDS);
claim_Ti3(Ti, Secret, K1);

claim_Ti4(Ti, Secret, K2);
claim_Ti5(Ti, Secret, K3);
claim_Ti6(Ti, Secret, K4);

Also, both roles claim to be alive and share the agreement and synchronization goals as follows:

claim_S7(S, Niagree);
claim_S8(S, Alive);
claim_Ti7(Ti, Alive);
claim_Ti8(Ti, Niagree);
claim_Ti9(Ti, Nisynch);

The Scyther verified proposed protocol and shows there are no attacks within bounds as shown in Figure 2.

4.2.2 AVISPA analysis

The other formal security analysis tool is AVISPA. This tool is used for validation of the security protocols by using formal modelling [50]. To describe any protocol in AVISPA we must be using a language called High-Level Protocol Specification Language (HLPSL), this language validates the security protocol specifies. We made a formal analysis of the proposed protocol with consideration to required security properties such as authentication and secrecy in HLPSL.

We made a formal analysis of the proposed protocol with consideration to required security properties such as authentication and secrecy in HLPSL.

(1) **Authentication:** In our AVISPA script, the authentication is modelled by witness and request predicates:

witness (Server, Tag, Protocolid, Information)
request (Tag, Server, Protocolid, Information)

When the RFID tag *wants* to authenticate the back-end server the *witness* predicate is used that is mean the back-end server is the witness for the *Information*. Whenever the RFID tag wants to requests to verify the *Information* the *request* predicate is used. The *authentication_on id* is the goal's section of authentication property in the HLPSL script where the *id* is the label of type *protocol_id*. If any tools of the back-end will find a trace that produces by an agent except the *Server* and in this trace found the *request* event was preceded by a *witness* event. Therefor will be reported about this attack, Moreover, if there is no valid *witness* found for *request* then will be reported about an attack trace.

(2) **Secrecy:** In our AVISPA script, the secrecy is modelled by the secret predicate:

secret (secret-information, protocol_id, Tag, Server)

This declares the secret *information*(*secret-information*) as a secret shared between *Tag* and *Server*. The label *protocol_id* of type (*protocolid*) is used to HLPSL goal identity. The statement (*secrecy_of ids*) in HLPSL goal's section must be presented in order to refer to it.

Tag and server defined as basic roles in our script, each one of them shares key (*K*) and text (*IDS*). *N1* and *N2* defined as random numbers and we used new function to freshly generated them. *Channel*(*dy*) also has been declared. The intruder is identified in the environment *role* section and we assumed that agents (*Tag* and *Server*) are known by the intruder.

OFMC and CL-AtSe back-ends are support *XOR* properties, when the others back-ends (SATMC and TA4SP) did not. By OFMC and CL-AtSe back-ends our protocol results have come secure as shown in Figures 3 and 4, which means that the protocol successfully meets the specified goal. Therefore, we can say that our protocol satisfies mutual authentication and confidentiality of sensitive data, compared to a passive intruder, as specified in the role of the environment.

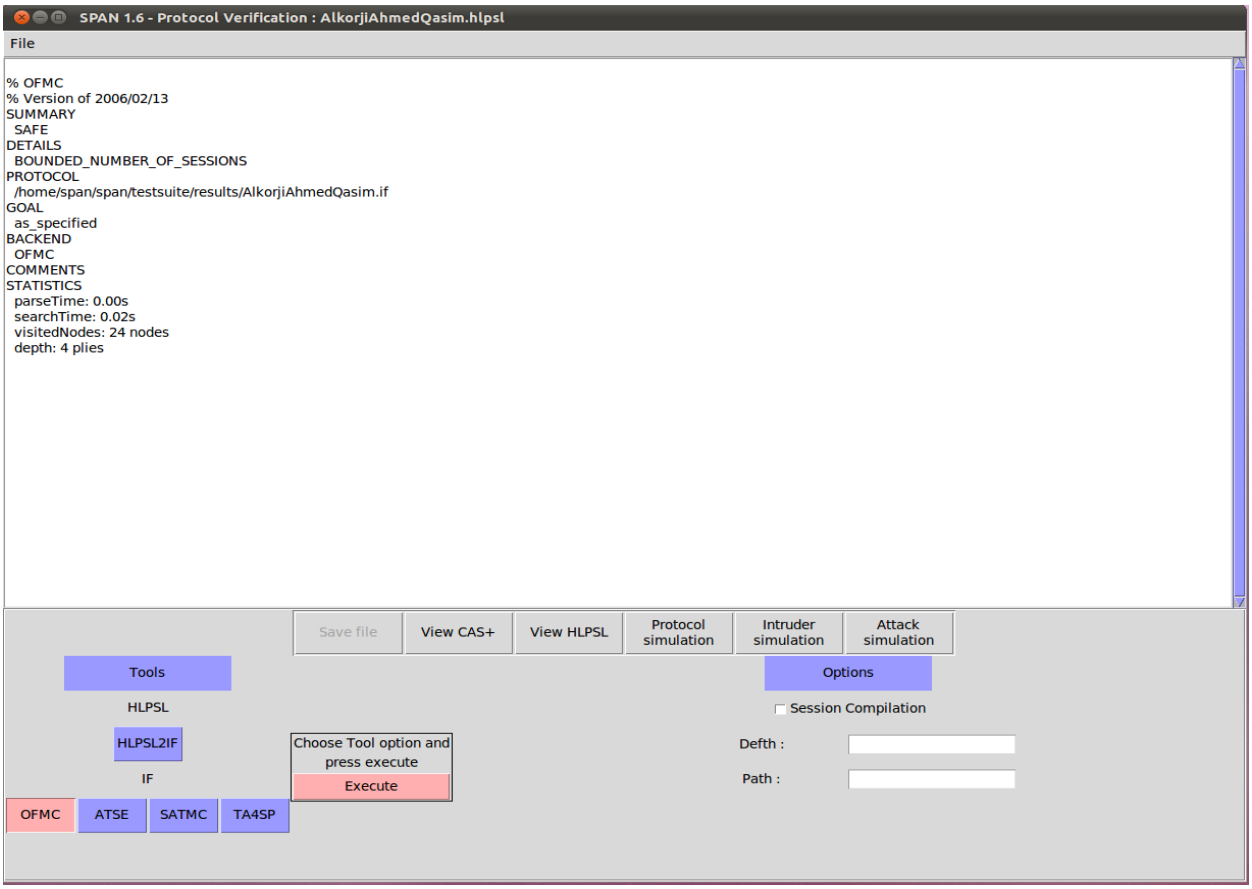

Figure 3. OFMC analysis results

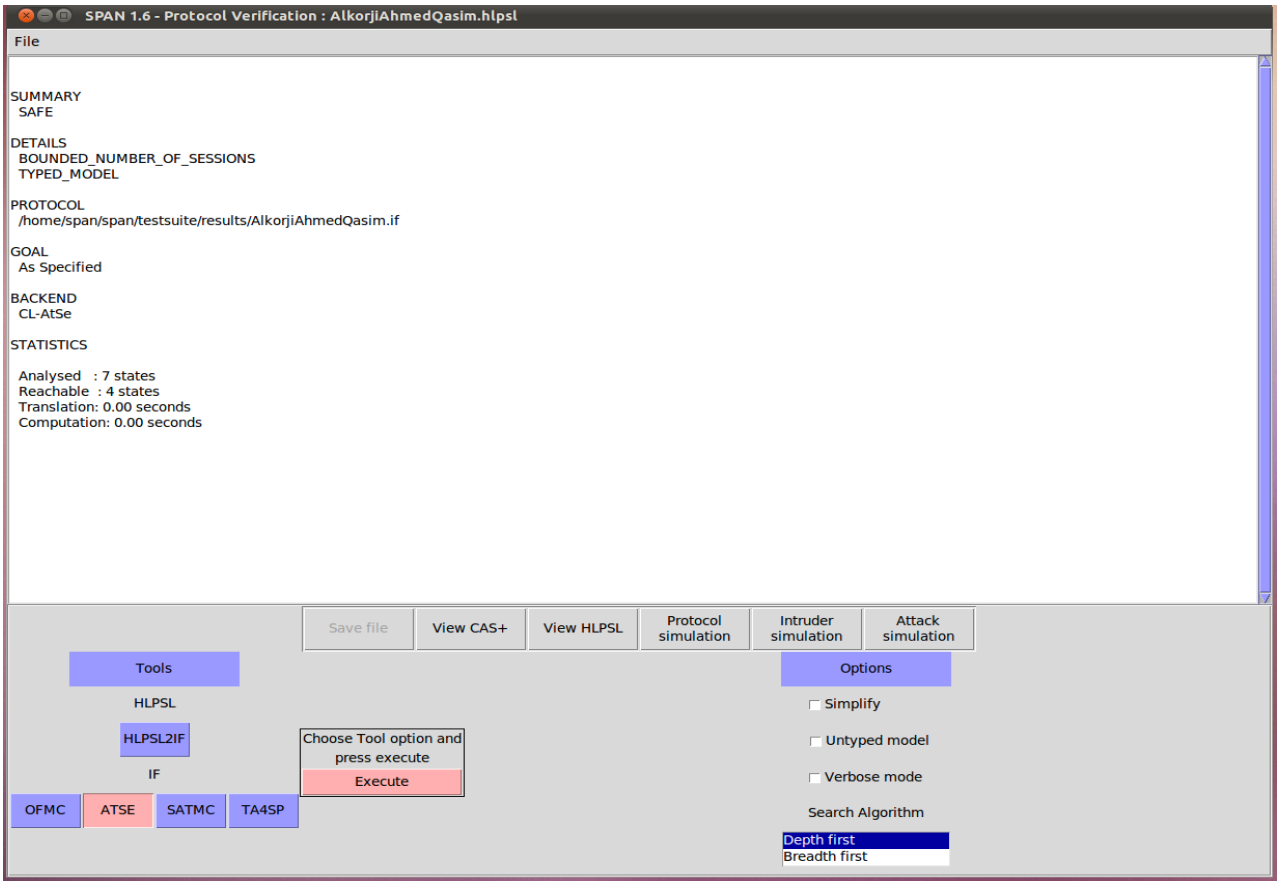

Figure 4. CL-at analysis results

5. PERFORMANCE ANALYSIS

This section presents a comparative analysis of the performance parameters related to ultra-lightweight mutual authentication protocols, evaluating our proposed protocol's effectiveness in meeting the objectives of this study.

(1) **Storage:** In this proposed protocol we assumed the length of all used string is L where L=96 bits. We only measure the RFID tag's side from the proposed protocol because the hardware resources of the back-end server are sufficient in the general case. In this protocol, the required storage for each tag is 6L bits which are (*ID, IDS, SID, K1, K2,* T_{SI}).

(2) **Communication messages:** In our proposed protocol the total transmitted communication messages between Server and Tag in one protocol session are 4L bits, while only 2L bits sent by the tag during this session. these influential properties will improve the authentication speed as well as reducing the power-consuming in authentication communication compared with other protocols.

Table 4 illustrates the efficiency of our proposed protocol in relation to existing alternatives. Notably, our protocol necessitates less storage (6L) compared to the other protocols, all of which require 7L. Additionally, it achieves a competitive total communication message count for authentication (4L). Furthermore, the protocol minimizes the tag's communication messages to 2L, which aligns with our objective of enhancing efficiency without compromising security. These metrics emphasize the advantages of our approach, demonstrating its potential for practical implementation in RFID systems.

(3) **The computational time:** The computational time cost refers to the time required for the RFID tag and back-end server during the mutual authentication protocol session. To facilitate a more effective comparison of computational costs across different protocols, we define specific symbols representing the execution time of computational operations. The experimental simulation was conducted using the following environment: Intel Core i7-2.40GHz processor with 8GB of RAM, utilizing Python as the programming language. To ensure precision and mitigate variability in each experiment, the average execution time was calculated after executing each computational operation 100 times. The average execution time for each computational operation shows in Table 5.

The total execution time for the solution proposed by Mujahid et al. [19] is calculated based on the number of operations required. Specifically, the solution necessitates nine XOR operations (T_{xor}), fifteen hash function operations (T_h), and ten rotation function operations (T_{rot}). Consequently, the overall execution time is expressed as $9T_{xor}+15T_h+10T_{rot}=3.824$ milliseconds. The protocol proposed described by Sun et al. [14] used twelve XOR operations (T_{xor}) and four hash function operations (T_h), Therefore, the total execution time is $12T_{xor}+4T_h=1.024$ milliseconds. The protocol proposed described by Wei et al. [13] used seven hash functions Therefore, the total execution time is $7T_h=1.771$ milliseconds. Our proposed protocol needs eighteen XOR operation T_{xor}, five of AND and OR operations, and five of rotation function operations (T_{rot}) therefore, the total execution time is $18T_{xor}+5T_{(and,\ or)}+5T_{rot}=0.033$ milliseconds, as shown in Table 6.

Table 4. Comparison of communication and storage metrics for UMAPs and our proposed protocol

Metric	Ref. [19]	Ref. [9]	Ref. [43]	Ref. [44]	Our Protocol
Storage	7L	7L	7L	7L	6L
Total communication messages for authentication	6L	4L	4L	17L	4L
Tag's communication messages	2L	2L	3L	5L	2L

Table 5. Execution times of operations

Symbols	Descriptions	Execution Time (Millisecond)
T_{xor}	Execution time for the XOR operation	0.001
T_{or}	Execution time for the OR operation	0.001
T_{and}	Execution time for the AND operation	0.001
T_{add}	Execution time for the Addition operation	0.001
T_{rot}	Execution time for the Rot operation	0.002
T_h	Execution time for the hash function	0.253

Table 6. Comparison of protocols' computational overhead and execution time

Protocols	Total Computational Overhead	Total Execution Time (ms)
Ref. [19]	$9T_{xor}+15T_h+10T_{rot}$	3.824
Ref. [14]	$12T_{xor}+4T_h$	1.024
Ref. [13]	$7T_h$	1.771
Proposed protocol	$18T_{xor}+5T_{(and,\ or)}+5T_{rot}$	0.033

6. CONCLUSIONS

Since the communications between the RFID tags and the RFID readers are carried out through an unprotected wireless channel, RFID systems, like other wireless technologies, face a new set of difficulties. As a result, there are various kinds of attacks that can target RFID systems. Furthermore, it was noted that every Mutual Authentication Protocol that had previously been suggested was open to several replay and desynchronization attacks. The subsection 1.2 noted that the RFID tags have limited processing power and storage capacity, which prevents them from carrying out the traditional security cryptography functions.

The computational capabilities of low-cost RFID tags are surpassed by the exorbitant power and memory needs of classic encryption algorithms and primitives. Because of this, the majority of academics are tackling these problems by creating a protocol that strengthens the security of the RFID system while overcoming these obstacles.

In this paper, we claim that there are two constraints in terms of RFID tag security present in most proposed protocols. First off, since RFID tags are limited in their ability to serve certain functions, the authors did not indicate which class of tags the suggested protocol is meant for. Secondly, numerous writers failed to indicate which class their suggested protocols fall into. Low-cost RFID tags are inappropriate for implementing UMAPs since they are based on hash functions, which are the most frequent type of protocol.

Because using an RFID tag that uses a protocol based on basic bitwise operations to safeguard sensitive data, such as credit card numbers, health information, or e-passport information, may be difficult. This problem has drawn attention from researchers, and as a result, numerous novel Ultra-Lightweight techniques are put forth annually. The current UMAP is insufficiently protected, despite the fact that numerous additional protocols are being devised.

In conclusion, the development of a new UMAP is critical for ensuring the security of RFID systems against prevalent attacks discussed in this paper. Our proposed UMAP is designed to meet the initial objectives outlined, with security analyses conducted using formal verification tools (Scyther and AVISPA) confirming its resilience to various attacks. Performance analysis reveals that our protocol requires only 6L of storage and achieves a competitive total communication message count for authentication of 4L, while reducing the tag's communication messages to 2L, thereby enhancing efficiency without compromising security. The total execution time of our proposed protocol is a mere 0.033 milliseconds, demonstrating its practicality for real-world applications. However, several limitations must be considered: the protocol's performance may be affected by environmental constraints, such as interference from external signals and signal collisions in densely populated areas; scalability issues may arise as the number of RFID tags increases, potentially creating bottlenecks in session key management; and while the protocol is designed to resist common attacks, it may not be robust against advanced threats like side-channel attacks or physical tampering. Moreover, effective key management is crucial; compromised keys could jeopardize overall security, and variations in hardware capabilities may limit applicability for extremely low-cost RFID tags. Future research should focus on enhancing robustness to environmental factors, investigating scalable architectures, strengthening defenses against advanced attacks, innovating key management practices, and conducting empirical studies across diverse RFID tag models. By addressing these limitations and pursuing the suggested areas for future research, the proposed UMAP can be further improved, ensuring its effectiveness and security across a broader range of scenarios.

REFERENCES

[1] Younis, M.I., Abdulkareem, M.H. (2017). ITPMAP: An improved three-pass mutual authentication protocol for secure RFID systems. Wireless Personal Communications, 96: 65-101.

https://doi.org/10.1007/s11277-017-4152-0
[2] Geng, L., Wei, X.Z., Zhang, J., Feng, X. (2023). A semi-numerical analysis of observations in passive tag-to-tag communications. Mathematical Modelling of Engineering Problems, 10(1): 31-38. https://doi.org/10.18280/mmep.100104
[3] Juels, A. (2006). RFID security and privacy: A research survey. IEEE Journal on Selected Areas in Communications, 24(2): 381-394. https://doi.org/10.1109/JSAC.2005.861395
[4] Jones, P., Clarke-Hill, C., Hillier, D., Comfort, D. (2005). The benefits, challenges and impacts of radio frequency identification technology (RFID) for retailers in the UK. Marketing Intelligence & Planning, 23(4): 395-402. https://doi.org/10.1108/02634500510603492
[5] Cao, T., Bertino, E., Lei, H. (2008). Security analysis of the SASI protocol. IEEE Transactions on Dependable and Secure Computing, 6(1): 73-77. https://doi.org/10.1109/TDSC.2008.32
[6] Khalid, M., Mujahid, U., Najam-ul-Islam, M. (2018). Cryptanalysis of ultralightweight mutual authentication protocol for radio frequency identification enabled Internet of Things networks. International Journal of Distributed Sensor Networks, 14(8). https://doi.org/10.1177/1550147718795120
[7] Khattab, A., Jeddi, Z., Amini, E., Bayoumi, M. (2017). RFID security threats and basic solutions. In RFID Security: A Lightweight Paradigm, Springer, Cham, pp. 27-42. https://doi.org/10.1007/978-3-319-47545-5_2
[8] Chien, H.Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Transactions on Dependable and Secure Computing, 4(4): 337-340. https://doi.org/10.1109/TDSC.2007.70226
[9] Luo, H., Wen, G., Su, J., Huang, Z. (2018). SLAP: Succinct and lightweight authentication protocol for low-cost RFID system. Wireless Networks, 24: 69-78. https://doi.org/10.1007/s11276-016-1323-y
[10] Chou, J.S., Chen, Y., Wu, C.L., Lin, C.F. (2011). An efficient RFID mutual authentication scheme based on ECC. Cryptology ePrint Archive, pp. 1-20. https://ia.cr/2011/418.
[11] Shen, J., Tan, H., Chang, S., Ren, Y., Liu, Q. (2015). A lightweight and practical RFID grouping authentication protocol in multiple-tag arrangements. In 2015 17th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea (South), pp. 681-686. https://doi.org/10.1109/ICACT.2015.7224882
[12] Khalid, M., Mujahid, U. (2017). Security framework of ultralightweight mutual authentication protocols for low cost RFID tags. In 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan, pp. 26-31. https://doi.org/10.1109/C-CODE.2017.7918896
[13] Wei, C.H., Hwang, M.S., Chin, A.Y.H. (2017). A secure privacy and authentication protocol for passive RFID tags. International Journal of Mobile Communications, 15(3): 266-277. https://doi.org/10.1504/IJMC.2017.083462
[14] Sun, H., Li, P., Xu, H., Zhu, F. (2019). An improvement RFID security authentication protocol based on hash function. In Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 12th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018), Springer, Cham, pp. 375-384. https://doi.org/10.1007/978-3-319-93554-6_35
[15] Al-Adhami, A., Ambroze, M., Stengel, I., Tomlinson, M. (2017). A 256 bit implementation of ECC-RFID based system using Shamir secret sharing scheme and Keccak hash function. In 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, pp. 165-171. https://doi.org/10.1109/ICUFN.2017.7993768
[16] Cho, Y., Kim, K.M., Iqbal, A., Lee, T.J. (2017). Efficient traffic control using hash function filter for massive IoT computational RFID communications. In Proceedings of the 2017 International Conference on Information Technology, pp. 297-301. https://doi.org/10.1145/3176653.3176725
[17] López, P.P., Castro, D.D.J.C.H., Garnacho, D.A.R. (2008). Lightweight cryptography in radio frequency identification (RFID) systems. Computer Science Department, Carlos III University of Madrid, p. 270.
[18] Ranasinghe, D.C., Engels, D.W., Cole, P.H. (2005). Low cost RFID systems: Confronting security and privacy. Paper Auto-ID Labs White Paper Journal, 1.
[19] Mujahid, U., Najam-ul-Islam, M., Jafri, A.R., Qurat-ul-Ain, Ali Shami, M. (2016). A new ultralightweight RFID mutual authentication protocol: Sasi using recursive hash. International Journal of Distributed Sensor Networks, 12(2): 9648971. https://doi.org/10.1155/2016/9648971
[20] Zhuang, X., Zhu, Y., Chang, C.C., Peng, Q. (2018). Security issues in ultralightweight RFID authentication protocols. Wireless Personal Communications, 98(1): 779-814. https://doi.org/10.1007/s11277-017-4895-7
[21] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A. (2006). M 2 AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In Ubiquitous Intelligence and Computing: Third International Conference, UIC 2006, Wuhan, China, pp. 912-923. https://doi.org/10.1007/11833529_93
[22] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A. (2006). EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In on the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. OTM 2006, Berlin, Heidelberg. https://doi.org/10.1007/11915034_59
[23] Peris-Lopez, P., Hernandez-Castro, J.C., Estévez-Tapiador, J.M., Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In Proceedings of 2nd Workshop on RFID Security, vol. 6.
[24] Li, T., Deng, R. (2007). Vulnerability analysis of EMAP-an efficient RFID mutual authentication protocol. In the Second International Conference on Availability, Reliability and Security (ARES'07), Vienna, Austria, pp. 238-245. https://doi.org/10.1109/ARES.2007.159
[25] Li, T., Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In IFIP International Information Security Conference, Boston, US, pp. 109-120. https://doi.org/10.1007/978-0-387-72367-9_10
[26] Hernandez-Castro, J.C., Tapiador, J.M., Peris-Lopez, P., Quisquater, J.J. (2008). Cryptanalysis of the SASI

ultralightweight RFID authentication protocol with modular rotations. arXiv Preprint arXiv:0811.4257. https://doi.org/10.48550/arXiv.0811.4257

[27] Phan, R.C.W. (2008). Cryptanalysis of a new ultralightweight RFID authentication protocol-SASI. IEEE Transactions on Dependable and Secure Computing, 6(4): 316-320. https://doi.org/10.1109/TDSC.2008.33

[28] Sun, H.M., Ting, W.C., Wang, K.H. (2009). On the security of Chien's ultralightweight RFID authentication protocol. IEEE Transactions on Dependable and Secure Computing, 8(2): 315-317. https://doi.org/10.1109/TDSC.2009.26

[29] Avoine, G., Carpent, X., Martin, B. (2010). Strong authentication and strong integrity (SASI) is not that strong. In Radio Frequency Identification: Security and Privacy Issues. RFIDSec 2010, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-16822-2_5

[30] Li, T. (2008). Employing lightweight primitives on low-cost RFID tags for authentication. In 2008 IEEE 68th Vehicular Technology Conference, Calgary, AB, Canada, pp. 1-5. https://doi.org/10.1109/VETECF.2008.290

[31] Safkhani, M., Bagheri, N., Naderi, M., Sanadhya, S.K. (2011). Security analysis of LMAP++, an RFID authentication protocol. In 2011 International Conference for Internet Technology and Secured Transactions, Abu, pp. 689-694.

[32] Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, J.M., Ribagorda, A. (2009). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In Information Security Applications: 9th International Workshop, WISA 2008, Jeju Island, Korea, pp. 56-68. https://doi.org/10.1007/978-3-642-00306-6_5

[33] David, M., Prasad, N.R. (2009). Providing strong security and high privacy in low-cost RFID networks. In Security and Privacy in Mobile Information and Communication Systems: First International ICST Conference, MobiSec 2009, Turin, Italy, pp. 172-179. https://doi.org/10.1007/978-3-642-04434-2_15

[34] Yeh, K.H., Lo, N.W., Winata, E. (2010). An efficient ultralightweight authentication protocol for RFID systems. In Radio Frequency Identification System Security, pp. 49-60. https://doi.org/10.3233/978-1-60750-485-6-49

[35] Tagra, D., Rahman, M., Sampalli, S. (2010). Technique for preventing DoS attacks on RFID systems. In SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks, Split, Croatia, pp. 6-10.

[36] Engels, D., Fan, X., Gong, G., Hu, H., Smith, E.M. (2010). Hummingbird: Ultra-lightweight cryptography for resource-constrained devices. In International Conference on Financial Cryptography and Data Security, pp. 3-18. https://doi.org/10.1007/978-3-642-14992-4_2

[37] Saarinen, M.J.O. (2011). Cryptanalysis of hummingbird-1. In Fast Software Encryption: 18th International Workshop, FSE 2011, Lyngby, Denmark, pp. 328-341. https://doi.org/10.1007/978-3-642-21702-9_19

[38] Zhang, K., Ding, L., Guan, J. (2012). Cryptanalysis of hummingbird-2. IACR Cryptology ePrint Archive, Report, 2012.

[39] Tian, Y., Chen, G., Li, J. (2012). A new ultralightweight RFID authentication protocol with permutation. IEEE Communications Letters, 16(5): 702-705. https://doi.org/10.1109/LCOMM.2012.031212.120237

[40] Wang, S.H., Han, Z.J., Liu, S.J., Chen, D.W. (2012). Security analysis of RAPP an RFID authentication protocol based on permutation. IACR Cryptology ePrint Archive, 2012: 327.

[41] Ahmadian, Z., Salmasizadeh, M., Aref, M.R. (2013). Desynchronization attack on RAPP ultralightweight authentication protocol. Information Processing Letters, 113(7): 205-209. https://doi.org/10.1016/j.ipl.2013.01.003

[42] Safkhani, M., Bagheri, N. (2016). Generalized desynchronization attack on UMAP: Application to RCIA, KMAP, SLAP and SASI $^+$ protocols. Cryptology ePrint Archive, pp. 905-912.

[43] Tewari, A., Gupta, B.B. (2017). Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. The Journal of Supercomputing, 73: 1085-1102. https://doi.org/10.1007/s11227-016-1849-x

[44] Aghili, S.F., Ashouri-Talouki, M., Mala, H. (2018). DoS, impersonation and de-synchronization attacks against an ultra-lightweight RFID mutual authentication protocol for IoT. The Journal of Supercomputing, 74: 509-525. https://doi.org/10.1007/s11227-017-2139-y

[45] Wang, K.H., Chen, C.M., Fang, W., Wu, T.Y. (2018). On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags. The Journal of Supercomputing, 74: 65-70. https://doi.org/10.1007/s11227-017-2105-8

[46] Jeon, I.S., Yoon, E.J. (2013). A new ultra-lightweight RFID authentication protocol using merge and separation operations. International Journal of Mathematical Analysis, 7(52): 2583-2593.

[47] Wang, S., Liu, S., Chen, D. (2015). Security analysis and improvement on two RFID authentication protocols. Wireless Personal Communications, 82: 21-33. https://doi.org/10.1007/s11277-014-2189-x

[48] Rama, N., Suganya, R. (2010). SSL-map: A more secure gossamer-based mutual authentication protocol for passive RFID tags. International Journal on Computer Science and Engineering, 2: 363-367.

[49] Cremers, C.J. (2008). The Scyther tool: Verification, falsification, and analysis of security protocols: Tool paper. In International Conference on Computer Aided Verification, pp. 414-418. https://doi.org/10.1007/978-3-540-70545-1_38

[50] Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., Drielsma, P.H., Heám, P.C., Kouchnarenko, O., Mantovani, J., Mödersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Viganò, L., Vigneron, L. (2005). The AVISPA tool for the automated validation of internet security protocols and applications. In Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, pp. 281-285. https://doi.org/10.1007/11513988_27

[51] Dolev, D., Yao, A. (1983). On the security of public key protocols. IEEE Transactions on Information Theory, 29(2): 198-208. https://doi.org/10.1109/TIT.1983.1056650

NOMENCLATURE

ID	The RFID tag's unique identity
IDS	RFID tag's pre-shares a pseudonym shared between tag and server
SID	Server pre-shares a pseudonym shared between server and RFID tag's
T_{S1}	Secret Key for each RFID tag
K1/K2	Secret Keys for each RFID tag with the back-end server
IDS_{old}	The old value of IDS
IDS_{new}	The new value of key IDS
$K1_{old}$	The old value of K1
$K1_{new}$	The new value of K1
$K2_{old}$	The old value of K2
$K2_{new}$	The new value of K2
n1	Random numbers generated by reader
st	Back-end server value is kept old or new to show if the tag uses new or old of values K1/K2 and IDS
$\oplus$	Bitwise-XOR operation
$\wedge$	Bitwise-AND operation
$\vee$	Bitwise-OR operation
Rot (a, b)	Left rotate the value of a with b bits
R $\rightarrow$T: M	R sends to T, message M