



A Lightweight Authentication Scheme for RFID with Permutation Operation on Passive Tags

Alireza Abdellahi Khorasgani^a, Mahdi Sajadieh^{a,*}, Mohammad rooholah Yazdani^a

^aDepartment of Electrical Engineering, Khorasgan (Isfahan) Branch, Islamic Azad University, Isfahan, Iran.

ARTICLE INFO.

Article history:

Received: 23 June 2021

Revised: 22 January 2022

Accepted: 30 January 2021

Published Online: 22 February 2022

Keywords:

RFID, Replay attacks, Reader impersonation, Tag tracking.

ABSTRACT

Rapid and ever-increasing Internet of things (IoT) developments have brought about great hopes of improving the quality of human life. Radio-frequency identification (RFID) employed as a backup technology in the IoT is widely used in different aspects of life. Therefore, high priority should be given to security problems and user privacy protection. However, limited computational power and storage resources in passive tags have made the implementation of security measures difficult in RFID. In other words, the design of lightweight authentication protocols for RFID systems is still a major challenge in RFID security. A lightweight authentication protocol has been recently proposed for passive tags by Liu et al. Using specific inverse operations in the IOLAS protocol, they claimed that the lightweight bitwise operations would make this protocol resistant against known and potential attacks in RFID systems. This study aimed to show that the same inverse operations pose the main problem so that this protocol fails to guarantee backward security. It was also indicated that the IOLAS protocol is vulnerable to replay, reader impersonation, tag tracking attacks, and secret disclosure attack. Finally, we improved the IOLAS protocol and proposed the POLAS protocol, which is resistant to the currently known attacks. We analyze the security level of the proposed protocols and prove the security of the proposed design using BAN (Burrows-Abadi-Needham) logic. We also formally confirmed the security of the proposal using the Scyther simulation tool. According to security analysis, we can observe that this protocol have a high level of security. A comparison of the performance of the POLAS protocol shows that this protocol is comparable to similar protocols in terms of computational costs, storage costs, and communication costs.

© Research Article, 2021 JComSec. All rights reserved.

* Corresponding author.

Email addresses: alireza.abdellahi@khuisf.ac.ir,
m.sajadieh@khuisf.ac.ir, m.sajadieh@khuisf.ac.ir
<https://dx.doi.org/10.22108/JCS.2022.129023.1068>
ISSN: 2322-4460 © Research Article, 2021 JComSec. All rights reserved.

1 Introduction

The IoT systems are used in many practical areas such as intelligent transportation systems (ITSs), health-care systems, and smart home networking to improve the quality of human life [1]. The IoT interconnects all things by the Internet to exchange information; therefore, the IoT can intelligently process and ana-



lyze the physical environment [2]. Identification and recognition account for an important part of the IoT. Used extensively in different areas of the IoT, RFID is considered one of its common technologies. RFID applications have gradually developed with the increasing popularity of the IoT. According to the market research by IDTechEx, the entire market share of RFID will rise to 13.4 billion dollars in 2022. [3, 4].

A conventional RFID system consists of a tag, a reader, and a back-end database server. The tag is attached to an object, and the reader can communicate with the tag by radio signals by transmitting and receiving the tag information. The reader is also connected to the server through wire communication. The server is responsible for managing hidden data exchanged between tags and readers [5, 6]. Although information transmission by radio frequencies can increase efficiency and facilitate transmission, it is also associated with potential security risks and threats. In other words, the communication channel between a tag and a reader is not safe, and an attacker can intrude into it to carry out different attacks. Mutual authentication protocols are usually employed to cope with the attacks between a tag and a reader [7].

Authentication protocols are considered the key to many security problems; however, it is impossible to use advanced cryptography methods such as AES, DES, RSA, and SHA1 due to the limited resources of low-cost passive tags. Hence, lightweight cryptography techniques such as CRC and PRNG are used along with logical functions (XOR, AND, bitwise circular shift, and permutation) in most protocols proposed for this purpose. The most serious challenge to RFID authentication schemes is to design a reliable, efficient, and low-cost authentication protocol. Multiple protocols have been proposed to solve security problems in RFID systems; however, most of these protocols suffer from certain drawbacks and shortcomings making them vulnerable to various attacks [8–10]. Recently, Liu et al. [11] proposed a novel lightweight authentication protocol for RFID systems; they claim that it provides a high security level. To guarantee security in their protocol, they used lightweight inverse functions and bitwise circular shift $Rot(x, y)$, which were deemed suitable for passive tags of limited resources. Unfortunately, a security analysis revealed a major security defect in the inverse function used in their proposed protocol. The same defect deprives this protocol of backward security making it vulnerable to tag tracking, replay, and reader impersonation attacks. Accordingly, a novel lightweight and secure authentication protocol is proposed in this paper to improve the protocol developed by Liu et al.

The rest of this paper consists of the following sections. The literature and authentication protocols proposed in recent years are reviewed in Section 2. The IOLAS protocol is analyzed in Section 3, and its security flaws are addressed in Section 4. The IOLAS protocol is modified in Section 5, and the POLAS protocol is proposed. Section 6 analyzes the security of the proposed protocols. In Section 7, the security of the proposed protocols is proofed using BAN logic. Section 8 examines the security of the proposed protocols using the Scyther simulation tool. Section 9 also analyzes the performance of the proposed protocols. Finally, the research results are presented and discussed in Section 10.

2 Related Work

Numerous authentication protocols have been proposed to guarantee security features and privacy protection in RFID systems. Based on computational costs and storage resources required on the tag side, these protocols are classified into four major categories. The first category includes mature protocols using cryptographic algorithms such as symmetric-key cryptography, public-key cryptography, and one-way cryptography functions for authentication [12]. The simple protocols in the second category which are used on the tags that support pseudorandom number generators (PRNGs) and one-way hash functions [13, 14]. The third category includes the protocols that their tags support PRNGs and simple operators such as cyclic redundancy codes (CRC) but fail to support one-way hash functions. These protocols are also known as lightweight protocols [15, 16]. The ultra-lightweight authentication protocols are considered the fourth category, in which cryptography operations include only simple bitwise operations such as XOR, AND, OR, modular addition, and bitwise circular shift [17–19]. Given the limited computational power and storage capacity of low-cost passive tags, ultra-lightweight protocols seem to be the best option for implementation on these tags. It is, therefore, a daunting challenge to meet security requirements through simple bitwise operations. Different authentication protocols proposed for this purpose are quite suitable for tags with limited computational power. Kulseng et al. proposed a lightweight two-way authentication method in RFID systems. Their proposed solution is based on linear-feedback shift registers (LFSRs) and physically unclonable functions (PUFs) involving lightweight and brief operations [20]. However, Kardas indicated that their proposed protocol has several security holes and thus is vulnerable to email injection attacks [21].



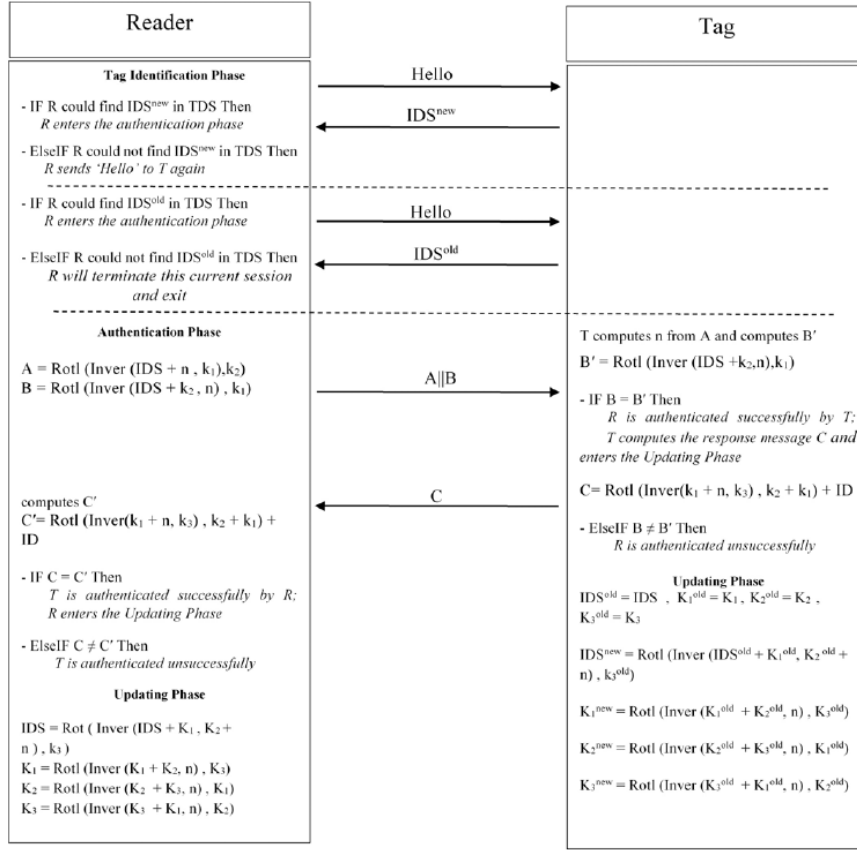


Figure 1. The IOLAS Protocol.

Luo et al. designed a lightweight authentication protocol for low-cost tags in RFID systems. The designers claimed that their protocol would resist different attacks [22]; however, Saffkhani et al. analyzed it and proved its vulnerability to the desynchronization attack [23]. Liu et al. proposed a two-way authentication protocol for RFID systems. Despite using a hash function, they claimed to have managed to decrease both computational and storage costs. In their protocol, they divided the hash function results into the right and left sides for tag and reader authentication; however, hash operations are computationally heavy and thus inappropriate for low-cost tags [24].

Gao et al. proposed an ultra-lightweight RFID authentication protocol based on the cryptographic algorithm [25]; however, analyses indicated that the proposed protocol was not appropriate for EPC C1 Gen2 passive tags. Xu et al. designed a two-way lightweight RFID authentication protocol based on physical unclonable functions using simple logical bitwise operations as security components. They claimed to overcome the desynchronization attack by storing messages from a previous session [26]. However, studies showed that the protocol is not resistant to desynchronization and hidden information disclosure attacks

[27]. Zhang et al. developed a group lightweight RFID authentication protocol and claimed that it met all security requirements [28]. Gholami et al. analyzed Zhang’s protocol and proved that it is not resistant to the desynchronization attack and the runout problem [29].

Recently, Liu et al. [11] proposed a lightweight IOLAS protocol for RFID systems. They used the lightweight inverse function and bitwise circular shift operator $Rot(x, y)$, which were deemed suitable for passive tags with limited resources. After analyzing the IOLAS protocol, this paper shows that it increased the tag-side computational cost and storage space. The paper also indicates the security flaws that make it vulnerable to replay, reader impersonation, tag tracking attacks, and secret disclosure attacks. Moreover, it is proven that the protocol fails to guarantee backward security.

3 Review of the IOLAS Protocol

Liu et al. (2019) [11] proposed a lightweight RFID authentication protocol for implementation on low-cost passive tags. They claimed that the tag-side computational cost decreased using three simple bitwise opera-



tions, *i.e.* modular addition, inverse $Inver(x, y)$, and circular shift $Rot(x, y)$. Therefore, this protocol is very efficient in tags with limited resources [11]. In the IOLAS algorithm, they used a new method named the inverse function and believed that it could improve the system reliability and security while preventing security problems that might arise from unbalanced trigonometric operations. The protocol is described below. The IOLAS protocol authentication process is shown in Figure 1.

3.1 Notations

Table 1 shows the notations used in this paper. Simple bitwise operations such as addition mode 2^q , circular shift $Rot(x, y)$, and inverse operation are utilized.

3.2 Inverse Operation

A new inverse function has been proposed in the IOLAS protocol. They claimed that this inverse operation brought about acceptable security levels in addition to being a lightweight technique. This inverse operation is as follows.

Definition 1. Assume two L -bit strings of $X = x_1x_2 \dots x_L$ and $Y = y_1y_2 \dots y_L$. The inverse operator is defined as follows:

$$Inver(x, y) = \begin{cases} x_i & y_i = 0 \\ NOT(x_i \oplus y_i) & y_i = 1 \end{cases} \quad (1)$$

For instance, if $x = 01011100$ and $y = 10010110$, their inverse orders are then defined as $Inver(x, y) = 01011100$.

3.3 IOLAS protocol

3.3.1 Initial Configurations

In this stage, Certain components of valid identities are described as follows:

- TDS selects a PRNG function, the instance of which was mentioned in [30], $g : (0, 1)^k \rightarrow (0, 1)^{2k}$. Accordingly, random integers are generated.
- TDS generates the cryptographic key ($K = K_1||K_2||K_3$) and configures the keys on valid readers and tags.
- TDS stores a valid R and a valid T and saves values of IDS and K .

3.3.2 Tag Identification Phase

- A R sends a Hello message to a T .
- After the T receives the Hello message from the R , it sends the response IDS^{new} .

- After receiving IDS^{new} , the R starts searching for IDS^{new} in TDS .
- If the R finds the value of IDS^{new} in TDS , it enters the two-way authentication phase; otherwise, the Hello message is resent to the T . After that, the T sends the value of IDS^{old} as a new response to the R .

If the R finds the value of IDS^{old} in TDS , it enters the two-way authentication phase; otherwise, the current session is canceled. In fact, the T 's response has been invalid. If IDS^{old} is the matched identification, the R uses $(K_1^{old}, K_2^{old}, K_3^{old})$ to continue communications in the next phase. If IDS^{new} is the matched identification, then $(K_1^{new}, K_2^{new}, K_3^{new})$ is used in the next phase.

3.3.3 Mutual Authentication Phase

- After receiving IDS^{new} or IDS^{old} from TDS , the R generates an L -bit pseudorandom number (n). Then the messages $A = Rotl(Inver(IDS + n, K_1), K_2)$ and $B = Rotl(Inver(IDS + K_2, n), K_1)$ are generated. After that, the R sends the $A||B$ messages to the T .
- After receiving $A||B$, the T extracts the value of n through from A through IDS , K_1 , and K_2 . The local value of B' is then calculated as $B' = Rotl(Inver(IDS + K_2, n), K_1)$. If B' equals B , it means that the T has received the value of n correctly. The R is also authenticated successfully. If B' is not equal to B , then the T guarantees that the R is a fake reader; thus, the session ends.
- After the R is authenticated by the T , the T calculates $C = Rotl(Inver(K_1 + n, K_3), K_2 + K_1) + ID$. Then the T sends the value of C to the R .
- After receiving C , the R calculates C' as $C' = Rotl(Inver(K_1 + n, K_3), K_2 + K_1) + ID$. If C' equals C , the T is authenticated successfully. If C' is not equal to C , the R guarantees that the T is an illegitimate tag; thus, the session ends.

3.3.4 Updating Phase

- After sending C , the T updates its local values through the following functions:

$$\begin{aligned} IDS^{old} &= IDS, K_1^{old} = K_1, K_2^{old} = K_2, K_3^{old} = K_3 \\ IDS^{new} &= Rotl(Inver(IDS^{old} + K_1^{old}, K_2^{old} + n), K_3^{old}) \\ K_1^{new} &= Rotl(Inver(K_1^{old} + K_2^{old}, n), K_3^{old}) \\ K_2^{new} &= Rotl(Inver(K_2^{old} + K_3^{old}, n), K_1^{old}) \\ K_3^{new} &= Rotl(Inver(K_3^{old} + K_1^{old}, n), K_2^{old}) \end{aligned} \quad (2)$$



Table 1. Nomenclature.

Notation	Description
T	Tag
R	Reader
TDS	Trusted database, which contains IDS and K between T and R
A, B	Messages from reader to tag in IOLAS protocol
C	Message from tag to reader IOLAS protocol
X_L	The left half of X bits
X_R	The Right half of X bits
A_{R1}	Message from reader to tag in POLAS protocol
A_{T1}, A_{T2}	Messages from tag to reader POLAS protocol
ID	Identification of T
IDS	Pseudonym of T in the current session
IDS^{old}	Pseudonym of T in the previous session
IDS^{new}	Pseudonym of T in the next session
$Inver(x, y)$	Inverse operation of x according to y
K	Secret key of T shared between TDS and T
K_1, K_2, K_3	Sub-secret key of K in current session. Each sub-key is 96 bits
$K_1^{old}, K_2^{old}, K_3^{old}$	Sub-secret key of K in the previous session
$K_1^{new}, K_2^{new}, K_3^{new}$	Sub-secret key of K in the next session
P	Mutual-authentication session between R and T
$PRNG$	Pseudo-Random Number Generator
$Rotl(x, y)$	Circular shift on the value of x , by $(y \bmod 96)$ positions to the left
$Rotr(x, y)$	Circular shift on the value of x , by $(y \bmod 96)$ positions to the right
$X_{>>y}$	shift on the value of X , by $(y \bmod 48)$ positions to the left
$X_{<<y}$	shift on the value of X , by $(y \bmod 48)$ positions to the right
$FXper^z(x, y)$	New permutation operation that combines general modified $Xper$ permutation and feistel structure operation
\oplus	XOR operation
$+$	Addition mod $2^q, q = 96$

The T also keeps the previous values of IDS_{old} , K_1^{old} , K_2^{old} , K_3^{old} .

- If the R authenticates the T successfully, the R updates its local values through the following functions:

$$\begin{aligned}
 IDS &= Rotl(Inver(IDS + K_1, K_2 + n), K_3) \\
 K_1 &= Rotl(Inver(K_1 + K_2, n), K_3) \\
 K_2 &= Rotl(Inver(K_2 + K_3, n), K_1) \\
 K_3 &= Rotl(Inver(K_3 + K_1, n), K_2)
 \end{aligned} \tag{3}$$

After updating these values, the R sends them to TDS .

4 Analysis of IOLAS

This section describes the security problems and defects of the IOLAS protocol to show its vulnerability to tag tracking, replay, and reader impersonation attacks. The main defect of this protocol lies in the inverse operations. The analysis of the inverse operations revealed that they put the input (x) in the output without any change. In other words, the output of $Inver(x, y)$ is, in fact, x without considering y ($Inver(x, y) = x$). This result is now proven for the sake of clarity. Assume that $\mathbf{x} = x_1x_2\dots x_L, x_i \in \{0, 1\}$ and $\mathbf{y} = y_1y_2\dots y_L, y_i \in \{0, 1\}$ are two L -bit strings. As discussed in Section 3.2, the corresponding output bit of every input put should be obtained through the inverse operations to determine the output of $F =$



$Inver(x, y)$.

- If $y_i = 0$, then the corresponding output bit is x_i ($F_i = x_i$).
- If $y_j = 1$, then
 - If $x_j = 0$, its corresponding output is $NOT(1+0) = 0$ or ($F_j = x_j$)
 - If $x_j = 1$, its corresponding output is $NOT(1+1) = 1$ or ($F_j = x_j$).

Accordingly, it can be stated that y has no effect on the output of $Inver(x, y)$ operations; therefore, $Inver(x, y) = x$. This can be seen in the example of the main article [11].

Considering the above inverse operations, the equivalences of the messages existing in the IOLAS protocol can be obtained as follows:

$$\begin{aligned}
 A &= Rotl(Inver(IDS + n, K_1), K_2) = Rotl(IDS + n, K_2) \\
 B &= Rotl(Inver(IDS + K_2, n), K_1) = Rotl(IDS + K_2, K_1) \\
 C &= Rotl(Inver(K_1 + n, K_3), K_2 + K_1) + ID = Rotl(K_1 + n, K_2 + K_1) + ID \\
 IDS^{new} &= Rotl(Inver(IDS^{old} + K_1^{old}, K_2^{old} + n), K_3^{old}) = Rotl(IDS^{old} + K_1^{old}, K_3^{old}) \\
 K_1^{new} &= Rotl(Inver(K_1^{old} + K_2^{old}, n), K_3^{old}) = Rotl(K_1^{old} + K_2^{old}, K_3^{old}) \\
 K_2^{new} &= Rotl(Inver(K_2^{old} + K_3^{old}, n), K_1^{old}) = Rotl(K_2^{old} + K_3^{old}, K_1^{old}) \\
 K_3^{new} &= Rotl(Inver(K_3^{old} + K_1^{old}, n), K_2^{old}) = Rotl(K_3^{old} + K_1^{old}, K_2^{old})
 \end{aligned}$$

The resulting values indicate that the random number n had no role in updating the common hidden values. Therefore, this problem might cause backward security violation, which will be discussed in the next section. Moreover, n had no effect on the B message; therefore, the random number n had also no role in reader authentication by the tag. It is used only in tag authentication by the reader.

An attacker can take advantage of this defect to carry out a replay, tag tracking, and reader impersonation attacks.

Another drawback of this protocol is observed in the tag identification phase when the reader requests IDS^{old} from the tag by sending the Hello message. The design flaw is detected when a malicious attacker uses a specific mechanism to prevent a successful session between the tag and the reader. In this mechanism, a malicious resends the Hello message to the tag by intruding into every session in every execution of the protocol immediately after receiving IDS^{new}

by the reader. In this case, the tag concludes that the previous session has been unsuccessful and sends IDS^{old} to the attacker and uses old values as the calculation basis. If the reader uses the new values as the calculation basis, its messages will not be validated by the tag; therefore, the tag and the reader will have no successful sessions.

Another problem with the IOLAS is that despite the limited storage spaces of passive tags, the storage spaces of the tags corresponding to this protocol are larger than the similar cases. The tag must store all hidden common items from the current and previous sessions in this protocol to prevent a desynchronization attack. These items are stored in either the reader or the server in most similar protocols. Consequently, this protocol is characterized by a larger number of exchanged messages than similar protocols.

4.1 Backward Security Violation

It is shown in this section that if a malicious attacker manages to obtain the key to a session between the tag and the reader, it will be able to calculate all hidden common items between them in all next sessions.

As explained in the Section 4, the random number n had no role in updating $K = K_1 || K_2 || K_3$ and IDS . Therefore, these values are, updated as follows:

$$\begin{aligned}
 IDS^{new} &= Rotl(IDS^{old} + K_1^{old}, K_3^{old}) \\
 K_1^{new} &= Rotl(K_1^{old} + K_2^{old}, K_3^{old}) \\
 K_2^{new} &= Rotl(K_2^{old} + K_3^{old}, K_1^{old}) \\
 K_3^{new} &= Rotl(K_3^{old} + K_1^{old}, K_2^{old})
 \end{aligned} \tag{4}$$

Now assume that an attacker has the key to a session like the first session ($K = K_1^1 || K_2^1 || K_3^1$). The attacker also knows the value of IDS by intruding into the first session.

According to the formula for updating the common secret values, attacker can calculate the key of the next session ($K = K_2^2 || K_2^2 || K_2^2$) and the IDS^2 .

$$\begin{aligned}
 IDS^2 &= Rotl(IDS^1 + K_1^1, K_3^1) \\
 K_1^2 &= Rotl(K_1^1 + K_2^1, K_3^1) \\
 K_2^2 &= Rotl(K_2^1 + K_3^1, K_1^1) \\
 K_3^2 &= Rotl(K_3^1 + K_1^1, K_2^1)
 \end{aligned} \tag{5}$$

Similarly, the attacker can now calculate the key ($K = K_3^3 || K_3^3 || K_3^3$) and IDS^3 for the third session using the key $K = K_2^2 || K_2^2 || K_2^2$ and IDS^2 . The attacker can also calculate all the values of K and IDS



for all next sessions in the same way. Therefore, if the malicious attacker manages to obtain the key to one session, it will be able to calculate all keys and IDS values of the next sessions. It is then fair to conclude that the IOLAS protocol fails to provide backward security because the pseudorandom number n has no role in updating the hidden common values in this protocol.

4.2 Replay and Reader Impersonation Attack

In these attacks, an attacker tries to send the messages obtained from the previous session to the target tag and impersonate the real reader. This attack consists of two phases.

- **Phase 1:** The attacker intrudes into a session between the target tag and the reader in this attack. At the end of the session, the attacker prevents message C, sent by the tag, from reaching the reader to prevent updating in the reader.
- **Phase 2:** In this phase, the attacker impersonates a reader. The attacker sends a Hello message to the target tag which then sends IDS^{new} to the attacker. Once IDS^{new} is received, the attacker obtains $A||B$ values from the previous session and sends them to the tag (it can be anything as the value of A has no role in authenticating the reader). Receiving $A||B$, the tag extracts the value of n from A, authenticates the attacker as a legitimate reader, and sends the C message to the attacker. Upon receiving C, the attacker concludes that it has been authenticated by the target tag. Therefore, the attack is successful.

In this attack, the attacker managed to introduce itself as a legitimate reader to the target tag using the messages obtained from the previous session. This attack occurs, first because the tag uses no random numbers in its messages. Second, the reader is first authenticated in this protocol, and the tag is then authenticated.

4.3 Tag Tracking Attack

In this attack, an attacker tries to track a target tag. This attack is carried out in two ways.

Method 1: The attack consists of two phases.

- **Phase 1:** In this phase of the attack, the attacker eavesdrops on the messages exchanged between the tag and the reader while executing the protocol. The attacker stores the values of IDS^{new} , $A||B$, and C, and finally prevents the C message from reaching the reader. Therefore, the reader does not update IDS and $K = K_1||K_2||K_3$ and keeps the old values; however, the tag updates

these values and also stores the values of the previous session.

- **Phase 2:** At this stage, the attacker eavesdrops on the messages exchanged between the tags and the reader. The attacker knows that due to the failure to update the IDS and $K = K_1||K_2||K_3$ values by the reader, the reader sends the hello value twice, to receive the IDS^{old} value from the tag. Therefore, when the IDS^{old} value is reached, the Reader generates $A = Rotl(IDS+n, K_2)$ and $B = Rotl(IDS + K_2, K_1)$ value and sends $A||B$ messages to the tag. The random number n has no effect on B and the reader doesn't update. So, the value of B equals that value of B in the previous session. As a result, the attacker compares the message B in this phase with the value of B from Phase 1. If these two values are equal, it can then be concluded that the tag from Phase 1 matches the tag from Phase 2. In this case, the attacker can identify and track the target tag among the other tags.

Method 2: This attack consists of two phases.

- **Phase 1:** In this phase, the attacker eavesdrops on the messages exchanged between the tag and the reader while executing the protocol. The attacker stores IDS^{new} . At the end of the session, the tag stores IDS^{new} as IDS^{old} in its memory and updates IDS^{new} .
- **Phase 2:** In this phase, the attacker starts the session as a reader and sends a Hello message to the tag. According to the protocol procedure, the tag sends IDS^{new} to the attacker upon receiving the Hello message. Once the IDS^{new} is received, the attacker resends the Hello message to the tag so that the tag will send IDS^{old} to the attacker. Upon receiving IDS^{old} , the attacker compares it with the stored value obtained from the previous session. If IDS^{old} equals IDS^{new} , obtained from the previous session, it can be concluded that the stolen tag matches the target tag; therefore, the attacker can identify and track the target tag among the other tags.

This attack occurs, first because the tag saves both IDS^{new} and IDS^{old} . Moreover, the tag gives IDS^{new} and IDS^{old} to the attacker without checking anything when a Hello message is sent by the attacker.

4.4 Secret disclosure attack

This section shows that the attacker can disclose the secret parameter of the target tag by eavesdropping on the messages of the two sessions. This attack consists of the following two phases.

Phase1: This phase consists of two stages:



1- The attacker eavesdrops on a session between the target tag and the reader in this attack. The attacker stores the values IDS , $A = Rotl(IDS + n_1, K_2)$, $B = Rotl(IDS + K_2, K_1)$ and $C = Rotl(K_1 + n_1, K_2 + K_1) + ID$. At the end of the session, the attacker prevents the message C. Therefore, the shared secret information between the tag and the reader is not updated in the reader.

2- In this section, the attacker eavesdrops on the next session between the target tag and the reader, and stores the values IDS , $A' = Rotl(IDS + n_2, K_2)$, $B' = Rotl(IDS + K_2, K_1)$ and $C' = Rotl(K_1 + n_2, K_2 + K_1) + ID$. As it is known, due to the reader not updating the information in the first session, the value of message B is equal to B' .

Phase2: In the second phase, the attacker executes the following algorithm with the information obtained from the eavesdropping phase.

$$\begin{aligned}
 & \text{For } i = 0 \text{ to } 96 \\
 & K'_2 = Rotr(B, i) - IDS \\
 & n'_1 = Rotr(A, K'_2) - IDS \\
 & n'_2 = Rotr(A', K'_2) - IDS \\
 & m = Rotl(i + n'_1, K'_2 + i) - Rotl(i + n'_2, K'_2 + i) \\
 & \text{if } m = C - C' \text{ then} \\
 & K_1 = K'_1 \\
 & K_2 = K'_2 \\
 & \text{Next For}
 \end{aligned} \tag{6}$$

The attacker uses the above algorithm to obtain K_1 and K_2 values. Therefore, by obtaining the values of K_1 and K_2 , the attacker can also reach the values of K_3 , and in the same way, the value of ID is revealed. Having K and IDS values, the attacker will achieve all these values in the next sessions. Now, using the revealed secret information, the attacker can also execute the tag Impersonation attack.

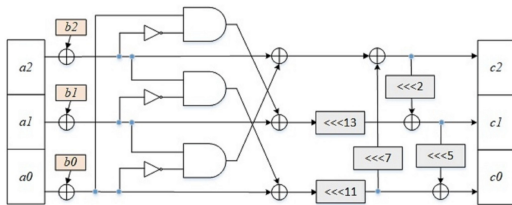


Figure 2. Modified $Xper(A, B)$.

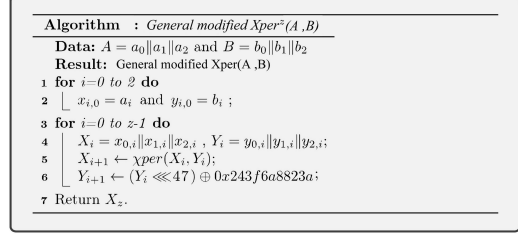


Figure 3. General Modified $Xper^z(A, B)$.

5 The Modified Protocol

A lightweight authentication protocol named POLAS is proposed, in which the security flaws of the IOLAS have been fixed. Providing a high level of security, the proposed protocol yields acceptable outputs on the tag side. The following measures are taken in POLAS to fix the security flaws of IOLAS:

- (1) Bitwise new permutation operations are used instead of inverse operations.
- (2) A new random number is generated by the tag in every session to be used in messages.
- (3) Like most authentication protocols, the values of K^{new} , IDS^{new} , K^{old} , and IDS^{old} are stored in the reader or the server to reduce the storage space on the tag side, and the tag only saves K^{new} and IDS^{new} .

5.1 New Permutation ($FXper^z(\cdot)$)

Ultra-lightweight functions such as circular shift $Rotl(x, y)$ (or $Rotr(x, y)$) and lightweight permutation $FXper^z(\cdot)$ are used in POLAS. The input and output lengths of $Rotl(x, y)$, $Rotr(x, y)$, and $FXper^z(\cdot)$ operations are 96 bits in this paper. The permutation operation used in this protocol is the integration of the general modified $Xper^z$ permutation scheme with the *feistel* structure which is called $FXper^z$. The modified $Xper^z$ operation uses the modified $Xper$ permutation operation introduced by Adeli et al. In [31]. Adeli et al. presented the $Xper$ permutation by using the nonlinear function used in the *Keccak* algorithm [32]. The modified $Xper$ permutation scheme is shown in Figure 2. Then, using the algorithm in Figure 3, The general modified $Xper^z$ permutation is introduced. The values of the variables z and w in general modified $Xper^z$ are $w = 16$ and $z > 4$. The variables z and w provide an exchange between performance and security. We place the general modified $Xper^z$ operation in a *feistel* structure with 4 rounds and the $FXper^z$ permutation operation which is a bijective function. Figure 4 describes the $FXper^z(\cdot)$ operation, which combines the general modified $Xper^z$ function with 4 rounds of the *feistel* structure. Since the general modified $Xper^z$ operation is combined with a *feistel* structure, the



general modified $Xper^z$ operation must be 48 bits, meaning that the input and output lengths of the general modified $Xper^z$ function must be 48 bits. As shown in Figures 2 and 3, the $Xper$ function and the general modified $Xper^z$ algorithm are designed with input and output length of 48 bits.

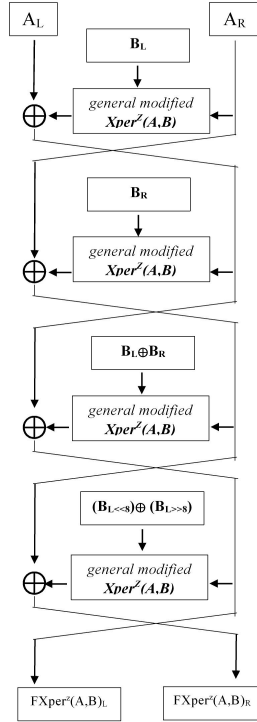


Figure 4. $FXper^z(A, B)$.

5.2 POLAS Protocol

The proposed protocol consists of three phases of (1) initialization, (2) authentication, and (3) updating. Figure 5 illustrates a schematic view of POLAS.

(1) Phase 1: Initialization

This phase is implemented in the exact same way as IOLAS.

(2) Phase 2: Authentication

In POLAS, the authentication phase consists of the following steps:

- **Step 1:** The reader generates a random number (n) and sends it with a Hello message to the tag.
- **Step 2:** After receiving the message, the tag generates another random number r and calculates A_{T1} and A_{T2} . Then the tag sends $IDS||A_{T1}||A_{T2}$ to the reader.

$$A_{T1} = Rotl(FXper^z(IDS + r, K_1 + n), K_1 + K_2)$$

$$A_{T2} = Rotl(FXper^z(ID + K_2, r + n), K_2 + K_3)$$

- **Step 3:** Upon receiving the message, the reader analyzes whether the IDS exists in the TDS database. If the IDS, sent by the tag, is found in the values existing in old and new records of the database, the reader takes the following actions; otherwise, the tag is fake, and the session ends.

Now the reader calculates r using $K = K_1||K_2||K_3$ and the IDS obtained from the A_{T1} message. The reader then calculates the local value of A'_{T2} through $A'_{T2} = Rotl(FXper^z(ID + K_2, r + n), K_2 + K_3)$. If A'_{T2} equals A_{T2} , the reader has obtained r correctly, and the tag has been authenticated successfully. If A'_{T2} and A_{T2} are not equal, the reader guarantees that the tag is fake and the session ends. After tag authentication by the reader, the reader calculates the A_{R1} message and sends it to the tag.

$$A_{R1} = Rotl(FXper^z(ID + K_3 + n, r + K_2), K_1 + K_3)$$

- **Step 4:** After receiving A_{R1} , the tag calculates A'_{R1} through $A'_{R1} = Rotl(FXper^z(ID + K_3 + n, r + K_2), K_1 + K_3)$. If A'_{R1} equals A_{R1} , the reader is authenticated successfully; otherwise, the tag guarantees that the reader is illegitimate, and the session ends.

(3) Updating

- **Step 1:** Once the tag is authenticated by the reader, the session enters the updating phase, divided into two situations. If the reader uses K^{old} and IDS^{old} for authentication, $K = K_1^{old}||K_2^{old}||K_3^{old}$ and IDS^{old} are left unchanged, whereas IDS^{new} and K^{new} are updated through the following relationships.

$$IDS^{new} = Rotl(FXper^z(IDS + K_1, K_2 + n), r + K_3)$$

$$K_1^{new} = Rotl(FXper^z(K_1 + K_2, n + r), K_3 + r)$$

$$K_2^{new} = Rotl(FXper^z(K_2 + K_3, n + r), K_1 + r)$$

$$K_3^{new} = Rotl(FXper^z(K_3 + K_1, n + r), K_2 + r).$$

If the reader uses K^{new} and FID^{new} for authentication, K^{old} and IDS^{old} are first initiated through $IDS^{old} \leftarrow IDS^{new}$, $K_1^{old} \leftarrow K_1^{new}$, $K_2^{old} \leftarrow K_2^{new}$, and $K_3^{old} \leftarrow K_3^{new}$. FID^{new} and K^{new} are then up-



6.3 Backward Security

In POLAS, the values of $K = K_1||K_2||K_3$ and IDS are updated through random values r and n when every session end. At the same time, the random value r is transmitted through an encrypted text; therefore, it is hidden from the attacker. In this case, even if the malicious attacker wishes to access the values of $K = K_1||K_2||K_3$ in a session, it will be impossible to access the keys to the next sessions without knowing r . Therefore, there will be no messages to be used by the attacker to access the confidential information of the next sessions. The proposed protocol, therefore, guarantees backward security.

6.4 Resistance to Replay Attack

In the proposed method, all messages are generated through random numbers r and n . On the other hand, the values of $K = K_1||K_2||K_3$ and IDS are updated after the execution of every session. Therefore, an attacker is unable to repeat the messages obtained from the previous sessions and cannot impersonate identities existing in the protocol. It can be concluded that the optimized protocol is resistant to replay attacks.

6.5 Resistance to Tag Impersonation Attack

In POLAS, an attacker is unable to send an anticipated response to the reader because the random number n is calculated by the reader and sent to the tag, which uses it in A_{T1} and A_{T2} . Hence, the attacker fails to achieve the desired goal by repeating messages from the previous sessions. The attacker is also unable to see the values of $K = K_1||K_2||K_3$ and IDS in the messages sent by the tag. Even if it is assumed that the attacker has managed to obtain the values of K , ID is still hidden; therefore, the attacker fails to calculate the messages of the tag. As a result, the proposed algorithm is secure against the tag impersonation attack.

6.6 Resistance to Reader Impersonation Attack

In the proposed protocol, the reader uses the random numbers r and n as well as the values of $K = K_1||K_2||K_3$ and IDS to calculate A_{R1} messages; however, the values of r , $K = K_1||K_2||K_3$, and IDS are hidden and transmitted through encrypted messages. Therefore, an attacker is unable to create an A_{R1} message and cannot repeat A_{R1} obtained from the previous session. Consequently, the attacker fails to deceive the tag.

6.7 Resistance to Tag Tracking Attack

In the proposed protocol, the tag calculates all messages using the new random numbers n and r ; therefore, the responses given by either the tag or the reader are neither constant nor predicted by the attacker. As a result, the attacker is deprived of the chance to track the target tag. The attacker will be able to track the tag.

6.8 Resistance to Desynchronization Attack

If an attacker is assumed to be able to block the last protocol message A_{R1} when the tag fails to update $K = K_1||K_2||K_3$ and IDS to carry out the desynchronization attack, TDS stores $K = K_1^{old}||K_2^{old}||K_3^{old}$, IDS^{old} , $K = K_1^{new}||K_2^{new}||K_3^{new}$, and IDS^{new} and authenticate the tag using $K = K_1^{old}||K_2^{old}||K_3^{old}$ and IDS^{old} . Therefore, the attacker is unable to make the tag enter the desynchronization state.

In Table 2, the security comparison between POLAS protocol and some other authentication protocols is expressed. In this table, the symbol 'Yes' represents that the authentication protocol resists against an attack and the symbol 'No' denotes that the authentication protocol does not resist the attack.

7 Formal Security Analysis

In this section, the security level of the POLAS protocol is evaluated according to the formal method based on BAN logic [37]. The analysis of a protocol using BAN logic is done in 5 steps. (1) Describing the messages of the protocol, (2) idealizing the messages of the protocol, (3) expressing explicit assumptions, (4) presenting security goals of the proposed scheme, (5) proving the proposed goals. Notations used in the BAN logic proof method are shown in Table 3.

The BAN logic consists of 19 rules. In this paper, 3 rules have been used. These two rules are stated as follows:

- (R_1) - $\frac{P|\equiv P \stackrel{Y}{\leftarrow} Q \cdot P \triangleleft \{X\}_Y}{P|\equiv Q|\wedge X}$: This message means: If P believes that Y is the shared secret between P and Q , and if P believes the message $\{X\}_Y$, then P believes that Q sent the message X .
- (R_2) - $\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$: This message means: If P believes that the message X is fresh, then it accepts that the message (X, Y) is fresh.
- (R_3) - $\frac{P|\equiv Q|\wedge (X,Y)}{P|\equiv Q|\wedge X}$: This message means: If P believes that Q sent the message set (X, Y) then P accepts that Q sent the message X .



Table 2. Security comparison between RAPP, LPCP, ULRAS, NULAMP, IOLAS and POLAS.

<i>Authentication protocol</i>	<i>RDS</i>	<i>RR</i>	<i>RTT</i>	<i>FBS</i>	<i>RDO</i>	<i>RDE</i>	<i>RTI</i>	<i>RRI</i>
<i>RAPP</i> [33]	No	No	No	No	No	No	Yes	Yes
<i>LPCP</i> [34]	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
<i>ULRAS</i> [35]	No	No	No	No	No	Yes	No	No
<i>NULAMP</i> [36]	No	Yes	No	No	No	No	Yes	Yes
<i>IOLAS</i> [11]	No	No	No	No	No	Yes	No	No
<i>POLAS</i>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

RDS : ResistancetoDisclosureofsecretValues

RR : ResistancetoReplayAttacks

RTT : ResistancetoTagTrackingAttack

FBS : Forward – BackwardSecurity

RDO : ResistancetoDOSAttacks

RDE : ResistancetoDesynchronizationAttacks

RTI : ResistancetoTagImpersonationAttack

RRI : ResistancetoReaderImpersonationAttack

Table 3. BAN-Logic Notation.

Notation	Description
$P \equiv X$	P believes X
$P \triangleleft X$	P receives X
$P \rightsquigarrow X$	P sends X
$\#(X)$	X is fresh
$\{X\}_k$	X encrypted with secret k
$P \stackrel{X}{\equiv} Q$	X is the shared secret of P and Q
$\frac{P}{Q}$	If P then Q

The proof of proposed protocol based on BAN logic is as follows :

Part1: Protocol Description : In this part, the messages transmitted between the tag and the reader are described.

$PM_1 : (R \rightarrow T) : \{Hello, n\}$

$PM_2 : (T \rightarrow R) : \{IDS, Rotl(FXper^z(IDS + r, K_1 + n), K_1 + K_2), Rotl(FXper^z(ID + K_2, r + n), K_2 + K_3)\}$

$PM_3 : (R \rightarrow T) : \{Rotl(FXper^z(ID + K_3 + n, r + K_2), K_1 + K_3)\}$

Part2: Protocol Idealization: In this part, the protocol messages are converted to the ideal form according to the BAN logic.

$IM_1 : (R \rightarrow T) : T \triangleleft \{n\}$

$IM_2 : (T \rightarrow R) : R \triangleleft \{IDS, Rotl(FXper^z(IDS + r, K_1 + n), K_1 + K_2), Rotl(FXper^z(ID + K_2, r + n), K_2 + K_3)\}$

$IM_3 : (R \rightarrow S) : S \triangleleft \{Rotl(FXper^z(ID + K_3 + n, r +$

```

Scyther: POLAS.spdl
File Verify Help
Protocol description Settings
1 settype Data:Key,Nonce,Data;
2 const Add: Function;
3 const Rotl: Function;
4 const RotR: Function;
5 const FXperz: Function;
6 const InAdd: Function;
7 const InFXperz: Function;
8 const ID;
9
10 protocol POLAS(Tag,Reader)
11
12 role Tag
13 {
14   const AR1,AR1';
15   secret IDS,ID,K1,K2,K3;
16   fresh n,Nonce;
17   var r: Nonce;
18
19   recv_1(Reader,Tag,n);
20   macro AT1=Rotl(FXperz(Add(IDS,r),Add(K1,n)),Add(K1,K2));
21   macro AT2=Rotl(FXperz(Add(IDS,K2),Add(r,n)),Add(K2,K3));
22   send_2(Tag,Reader,IDS,AR1,AT2);
23   recv_3(Reader,Tag,AR1);
24   macro AR1'=Rotl(FXperz(Add(ID,K3,n),Add(r,K3)),Add(K1,K3));
25   match (AR1,AR1');
26   macro IDS=Rotl(FXperz(Add(IDS,K1),Add(K2,n)),Add(K1,K3));
27   macro K1=Rotl(FXperz(Add(K1,K2),Add(r,n)),Add(K1,r));
28   macro K2=Rotl(FXperz(Add(K2,K3),Add(r,n)),Add(K1,r));
29   macro K3=Rotl(FXperz(Add(K3,K1),Add(r,n)),Add(K2,r));
30   claim(Tag,Secret,IDS);
31   claim(Tag,Secret,K1);
32   claim(Tag,Secret,K2);
33   claim(Tag,Secret,K3);
34   claim(Tag,Secret,IDS);
35   claim(Tag,Nisagree);
36   claim(Tag,Nisynch);
37   claim(Tag,Alive);
38   claim(Tag,Weakagree);
39
40 }
41
42 role Reader
43 {
44   const AT1,AT1',AT2,AT2';
45   secret IDS,ID,K1,K2,K3;
46   fresh n,Nonce;
47   var r: Nonce;
48
49   send_1(Reader,Tag,n);
50   recv_2(Tag,Reader,IDS,AT1,AT2);
51   macro r=InAdd(InFXperz(RotR(AT1,Add(K1,K2)),Add(K2,n)),IDS);
52   match (AT1,AT2);
53   macro AR1=Rotl(FXperz(Add(ID,K3,n),Add(r,K3)),Add(K1,K3));
54   send_3(Reader,Tag,AR1);
55   macro IDS=Rotl(FXperz(Add(IDS,K1),Add(K2,n)),Add(r,K3));
56   macro K1=Rotl(FXperz(Add(K1,K2),Add(r,n)),Add(K1,r));
57   macro K2=Rotl(FXperz(Add(K2,K3),Add(r,n)),Add(K1,r));
58   macro K3=Rotl(FXperz(Add(K3,K1),Add(r,n)),Add(K2,r));
59   claim(Reader,Secret,IDS);
60   claim(Reader,Secret,K1);
61   claim(Reader,Secret,K2);
62   claim(Reader,Secret,K3);
63   claim(Reader,Secret,IDS);
64   claim(Reader,Nisagree);
65   claim(Reader,Nisynch);
66   claim(Reader,Alive);
67   claim(Reader,Weakagree);
68
69 }

```

Figure 6. SPDL Code of POLAS Scheme.

$K_2), K_1 + K_3\}$

Part3: Initial Assumptions : The initial explicit assumptions of the proposed protocol are as follows :

$$\begin{aligned} A_1 : R| &\equiv \#(n) \\ A_2 : T| &\equiv \#(r) \\ A_3 : R| &\equiv R \stackrel{IDS}{\rightleftharpoons} T \\ A_4 : R| &\equiv R \stackrel{K}{\rightleftharpoons} T \\ A_5 : T| &\equiv T \stackrel{IDS}{\rightleftharpoons} R \\ A_4 : T| &\equiv T \stackrel{K}{\rightleftharpoons} R \end{aligned}$$

Part4: Proving Goals : In this part, the security goals of the proposed protocol are shown.

$$\begin{aligned} G_1 : R| &\equiv T| \sim r \\ G_2 : R| &\equiv T| \sim ID \\ G_3 : R| &\equiv \#\{Rotl(FXper^z(ID + K_2, r + n), K_2 + K_3)\} \\ G_4 : T| &\equiv R| \sim ID \\ G_5 : R| &\equiv \#\{Rotl(FXper^z(ID + K_3 + n, r + K_2), K_1 + K_3)\} \end{aligned}$$

Part5: Proof Process: In the current part, the security level of the protocol is analysed by applying BAN logic rules.

Result 1: According to IM_2, A_3, A_4, R_1 and R_3 can be concluded that $R| \equiv T| \sim R$

Result 2: According to IM_2, A_3, A_4, R_1 and R_3 can be concluded that $R| \equiv T| \sim ID$

Result 3: According to IM_2, A_1 and R_2 can be concluded that $S| \equiv \#\{Rotl(FXper^z(ID + K_2, r + n), K_2 + K_3)\}$

Result 4: According to IM_1, A_5, A_6, R_1 and R_3 can be concluded that $R| \equiv R| \sim ID$

Result 5: According to IM_1, A_2 and R_2 can be concluded that $T| \equiv \#\{Rotl(FXper^z(ID + K_3 + n, r + K_2), K_1 + K_3)\}$

Based on the results, it can be seen that all the security goals of the proposed protocol have been met.

8 Security Verification Using Scyther Simulation Tool

In order to more accurately verify the security validity of the proposed protocol, a formal security analysis tool called Scyther is used. Scyther is a widely accepted automatic push-down security simulation tool used to validate security protocols [38, 39]. Specifications or codes of protocols are expressed in this tool based on the Security Protocol Description Language (SPDL). The SPDL is based on the syntax of languages such as C, Java, etc. The SPDL language helps to evaluate the

security claims of protocols by describing roles and expressing events (such as receiving (recv) and sending (send)). If the protocol claim is incorrect, the protocol is vulnerable to at least one of the existing attacks. So if the Scyther tool is unable to detect the attack, the message "OK" is displayed. If an attack is detected, a "Fail" message is displayed. The Scyther tool consists of several predefined claims such as Secret, Niagree, Aliveness, Weakagree, Nisynch, and session-key reveal, etc. The match event is used to determine pattern matching. The SPDL codes used for tags and reader are shown in Figure 6. The results of the Scyther simulation in Figure 7 show that the POLAS protocol is resistant to all active and inactive attacks.

Claim	Status	Comments
POLAS.Tag	ok	No attacks within bound
POLAS.Tag2	ok	No attacks within bound
POLAS.Tag3	ok	No attacks within bound
POLAS.Tag4	ok	No attacks within bound
POLAS.Tag5	ok	No attacks within bound
POLAS.Tag6	ok	No attacks within bound
POLAS.Tag7	ok	No attacks within bound
POLAS.Tag8	ok	No attacks within bound
POLAS.Tag9	ok	No attacks within bound
POLAS.Tag10	ok	No attacks within bound
POLAS.Tag11	ok	No attacks within bound
POLAS.Tag12	ok	No attacks within bound
POLAS.Tag13	ok	No attacks within bound
POLAS.Tag14	ok	No attacks within bound
POLAS.Tag15	ok	No attacks within bound
Reader	ok	No attacks within bound
POLAS.Reader2	ok	No attacks within bound
POLAS.Reader3	ok	No attacks within bound
POLAS.Reader4	ok	No attacks within bound
POLAS.Reader5	ok	No attacks within bound
POLAS.Reader6	ok	No attacks within bound
POLAS.Reader7	ok	No attacks within bound
POLAS.Reader8	ok	No attacks within bound
POLAS.Reader9	ok	No attacks within bound
POLAS.Reader10	ok	No attacks within bound
POLAS.Reader11	ok	No attacks within bound
POLAS.Reader12	ok	No attacks within bound
POLAS.Reader13	ok	No attacks within bound
POLAS.Reader14	ok	No attacks within bound
POLAS.Reader15	ok	No attacks within bound
POLAS.Reader16	ok	No attacks within bound
POLAS.Reader17	ok	No attacks within bound

Figure 7. Scyther Simulation Results of POLAS Scheme.

9 Performance Analysis

Table 4 shows a comparison of the performance of lightweight protocols similar to the POLAS protocol. The comparison shows that the proposed protocol is efficient enough. In Table 4, x is the symbol for the Xor operation. ad is the symbol for the addition modulo $2L$. rot also represents the rotation operation. pr represents the *pseudo-randomnumbergenerator*. cr represents the cyclic redundancy check operation. $FXper^z$ is the symbol for the generally modified permutation. In RAPP, the tag stores (IDS, K_1, K_2, K_3) ; therefore, the storage space in the tag is $4L$. In the LPCP protocol, the tag stores $(TID, Key_{TH}, Key_{TM}, Key_{TL})$, so, the storage space in the tag is $4L$. In the ULRAS protocol, the storage space in the tag is $4L$, the



Table 4. Performance Comparison.

Authentication protocol	COT	SST	LMT	NMT
<i>RAPP</i> [33]	$7pr + 2rot + 9x$	4L	2L	2
<i>LPCP</i> [34]	$8pr + 14cr + 15x$	4L	2L	2
<i>ULRAS</i> [35]	$4rot + 7x + 2ad$	4L	2L	1
<i>NULAMP</i> [36]	$4rot + 7x$	3L	3L	2
<i>IOLAS</i> [11]	$4in + 4rot + 6ad$	9L	3L	3
<i>POLAS</i>	$3FXper^Z + 5rot + 10ad$	5L	3L	1

COT : Computational operations on the tag

SST : Storage space on the tag

LMT : The length of Messages transmitted by the tag

NMT : The number of Messages transmitted by the tag

tag stores (ID, IDS, K, T_T) . In NULAMP protocol, the storage space in the tag is 3L, the tag stores $(IDS_{old}, IDS_{new}, K)$. In IOLAS protocol, the storage space in the tag is 9L, the tag stores $(ID, IDS_{old}, IDS_{new}, K_1^{old}, K_1^{new}, K_2^{old}, K_2^{new}, K_3^{old}, K_3^{new})$. In the proposed protocol proposal the storage space in the tag is 5L, where the tag stores the tuple (ID, IDS, K_1, K_2, K_3) . it can be said that this comparison shows that POLAS still has advantages over some similar models. The tag storage space in this protocol is smaller than the IOLAS protocol. In this protocol, the tag transmits fewer messages than other similar protocols.

10 Conclusions

The security of a lightweight authentication protocol for passive tags, IOLAS, proposed by Liu et al. was analyzed. Specific inverse operations were developed in IOLAS for message cryptography, and they claimed that the use of inverse operations not only made the proposed protocol a lightweight protocol but also provided it with significant security features. It was proved in this study that they made a mistake in designing inverse operations. The inputs of these operations are repeated in the outputs. This flaw caused the loss of backward security features in IOLAS. Their proposed algorithm is also vulnerable to replay, reader impersonation, tag tracking attacks, and secret disclosure attack. Furthermore, the POLAS protocol was proposed to improve IOLAS. The lightweight POLAS is compatible with the IoT. Simple permutation operations were used instead of inverse operations in POLAS to increase security levels against different attacks and enhance the efficiency on the tag side. Various studies using BAN logic and

Syther tools showed that the POALS protocol has a high level of security. Comparison of POLAS protocol with other similar protocols shows that this protocol has a high level of security and despite the increase of some executive operations in this protocol, it still has an acceptable level of performance.

References

- [1] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu. Authentication Protocols for Internet of Things: A Comprehensive Survey. *Security and Communication Networks*, 55(2017), 2017. doi:10.1155/2017/6562953.
- [2] K. Fan, W. Wang, W. Jiang, H. Li, and Y. Yang. Secure ultra-lightweight RFID mutual authentication protocol based on transparent computing for IoV. *Peer-to-Peer Networking and Applications*, 11(4):723–734, 2018. doi:10.1007/s12083-017-0553-9.
- [3] J. Wang, H. Hassanieh, D.Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. *ACM SIGCOMM Computer Communication Review*, 42(4):61–72, 2012. doi:10.1145/2377677.2377685.
- [4] L. Xiao, H. Xu, F. Zhu, R. Wang, and P. Li. SKINNY-Based RFID Lightweight Authentication Protocol. *Sensors*, 20(5):1366, 2020. ISSN 1424-8220. doi:10.3390/s20051366.
- [5] T. Yeh and C. Wu. An enhanced ultralightweight RFID authentication protocol. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pages 799–804. IEEE, 2009. ISBN 978-1-4244-5227-9. doi:10.1109/JCPC.2009.5420075.
- [6] G. Avoine, C. Lauradoux, and T. Martin. When compromised readers meet RFID. In *Interna-*



- tional Workshop on Information Security Applications*, pages 36–50. Springer, 2009. ISBN 978-3-642-10837-2. doi:10.1007/978-3-642-10838-9_4.
- [7] R. Madhusudhan, M. Hegde, and I. Memon. A secure and enhanced elliptic curve cryptography-based dynamic authentication scheme using smart card. *International Journal of Communication Systems*, 31(11):69–78, 2018. doi:10.1002/dac.3701.
- [8] S. Karthikeyan and M. Nesterenko. RFID security without extensive cryptography. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, page 63–67. ACM, 2005. doi:10.1145/1102219.1102229.
- [9] K. Michael and R. Monteleone. Microchipping People is a. *Bad Idea”: An Interview with Andreas Sjoström*, *IEEE Technology and Society Magazine*, 38(2):18–21, 2019.
- [10] S. F. Aghili, H. Mala, P. Kaliyar, and M. Conti. SecLAP: Secure and lightweight RFID authentication protocol for Medical IoT. *Future Generation Computer Systems*, 101:621–634, 2019. doi:10.1016/j.future.2019.07.004.
- [11] Yali Liu, Xinchun Yin, Yongquan Dong, and Keke Huang. Lightweight authentication scheme with inverse operation on passive RFID tags. *Journal of the Chinese Institute of Engineers*, 42(11):74–79, 2019. doi:10.1080/02533839.2018.1537811.
- [12] C. Jin, C. Xu, X. Zhang, and J. Zhao. A secure RFID mutual authentication protocol for healthcare environments using elliptic curve cryptography. *Journal of Medical Systems*, 39(3):1–8, 2015. doi:10.1007/s10916-015-0213-7.
- [13] L. Heng, G. Fei, X. Yanming, and F. Shuo. Research of RFID Authentication Protocol Based on Hash Function. In *Advances in Wireless Networks and Information Systems*, pages 177–182. Springer, 2010. ISBN 978-3-642-14349-6. doi:10.1007/978-3-642-14350-2_22.
- [14] Y. Zhou and D. Feng. Design and analysis of RFID security protocol. In *Chin. J. Comput*, pages 581–590, 2006. ISBN 978-1-4244-5227-9.
- [15] G. Wei and H. Zhang. A lightweight authentication protocol scheme for RFID security. *Wuhan University Journal of Natural Sciences*, 18(6):504–510, 2013. doi:10.1007/s11859-013-0964-2.
- [16] P. ope and T. Hwang. A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system. *Computers & Security*, 55:271–280, 2015. doi:10.1016/j.cose.2015.05.004.
- [17] P. Peris-Lopez, J. Hernandez-Castro Cesar, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*, pages 352–361. Springer, 2006. ISBN 978-3-540-48269-7. doi:10.1007/11915034_59.
- [18] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Tapiador, and A. Ribagorda. Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In *International Workshop on Information Security Applications*, pages 56–68. Springer, 2008. ISBN 978-3-642-00305-9. doi:10.1007/978-3-642-00306-6_5.
- [19] H. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, 2007. doi:10.1109/TDSC.2007.70226.
- [20] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan. Lightweight mutual authentication and ownership transfer for RFID systems. In *2010 Proceedings IEEE INFOCOM*, pages 1–5. IEEE, 2010. ISBN 978-1-4244-5836-3. doi:10.1109/INFOCOM.2010.5462233.
- [21] S. Kardaş, M. Akgün, M. Kiraz Sabir, and H. Demirci. Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems. In *2011 Workshop on Lightweight Security & Privacy: Devices, Protocols, and Applications*, pages 20–25. IEEE, 2011. ISBN 978-1-61284-170-0. doi:10.1109/LightSec.2011.10.
- [22] H. Luo, G. Wen, J. Su, and Z. Huang. SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system. *Wireless Networks*, 24(1):69–78, 2018. doi:10.1007/s11276-016-1323-y.
- [23] M. Safkhani and N. Bagheri. Generalized Desynchronization Attack on UMAP: Application to RCIA, KMAP, SLAP and SASI+ protocols. *Cryptology ePrint Archive*, 2016.
- [24] B. Liu, B. Yang, and X. Su. An improved two-way security authentication protocol for RFID system. *Information*, 9(4):86, 2018. ISSN 2078-2489. doi:10.3390/info9040086.
- [25] X. Gao, S. Lv, H. Zhang, X. Li, W. Ji, Y. He, and X. Li. A kind of RFID Security Protocol Based on the Algorithm of Present. In *2018 5th International Conference on Systems and Informatics (ICSAI)*, pages 50–55. IEEE, 2018. ISBN 978-1-7281-0120-0. doi:10.1109/ICSAI.2018.8599339.
- [26] H. Xu, J. Ding, P. Li, F. Zhu, and R. Wang. A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function. *Sensors*, 18(3):760, 2018. ISSN 1424-8220. doi:10.3390/s18030760.
- [27] Y. Bendavid, N. Bagheri, M. Safkhani, and S. Rostampour. IoT Device Security: Challenging “A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function”. *Sensors*, 18(12):4444, 2018. ISSN 1424-8220.



- doi:10.3390/s18124444.
- [28] W. Zhang, S. Liu, S. Wang, B. Yi, and L. Wu. An Efficient Lightweight RFID Authentication Protocol with Strong Trajectory Privacy Protection. *Wireless Personal Communications*, 96(1): 1215–1228, 2017. doi:10.1007/s11277-017-4232-1.
- [29] V. Gholami and M. R. Alagheband. Provably privacy analysis and improvements of the lightweight RFID authentication protocols. *Wireless Networks*, 26(3):2153–2169, 2020. doi:10.1007/s11276-019-02037-z.
- [30] P. Peris-Lopez, J. Hernandez-Castro Cesar, J. M. Estevez-Tapiador, and A. Ribagorda. LAMED—a PRNG for EPC class-1 generation-2 RFID specification. *Computer Standards & Interfaces*, 31(1):88–97, 2009. doi:10.1016/j.csi.2007.11.013.
- [31] M. Adeli, N. Bagheri, S. Sadeghi, and S. Kumari. χ perbp: a Cloud-based Lightweight Mutual Authentication Protocol. *Cryptology ePrint Archive*, 2021.
- [32] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Keccak. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 313–314. Springer, 2013. ISBN 978-3-642-38347-2. doi:10.1007/978-3-642-38348-9_19.
- [33] Y. Tian, G. Chen, and J. Li. A New Ultralightweight RFID Authentication Protocol with Permutation. *IEEE Communications Letters*, 16(5):702 – 705, 2012. ISSN 1089-7798. doi:10.1109/LCOMM.2012.031212.120237.
- [34] L. Gao, M. Ma, Y. Shu, and Y. Wei. An ultralightweight RFID authentication protocol with CRC and permutation. *Journal of Network and Computer Applications*, 41:37–46, 2014. doi:10.1016/j.jnca.2013.10.014.
- [35] K. Fan, N. Ge, Y. Gong, H. Li, R. Su, and Y. Yang. An ultra-lightweight RFID authentication scheme for mobile commerce. *Peer-to-peer Networking and Applications*, 10(2):368–376, 2017. doi:10.1007/s12083-016-0443-6.
- [36] A. Tewari and B. B. Gupta. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. *The Journal of Supercomputing*, 73(3):1085–1102, 2017. doi:10.1007/s11227-016-1849-x.
- [37] M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. In *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, pages 233–271. The Royal Society London, 1989. doi:10.1098/rspa.1989.0125.
- [38] C. J. Cremers. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In *International Conference on Computer Aided Verification*, pages 414–418. Springer, 2008. ISBN 978-3-540-70543-7. doi:10.1007/978-3-540-70545-1.38.
- [39] C. J. F. Cremers. *Scyther: Semantics and verification of security protocols*. Eindhoven university of Technology Eindhoven, Netherlands, 2006.



includes RFID and IoT systems security.



coding.



electrical and Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran. His research interests include soft-switching converters, EMI modeling and reduction techniques, signal integrity and EMC issues.

