future internet

Article

On the Security of Rotation Operation Based Ultra-Lightweight Authentication Protocols for RFID Systems

Masoumeh Safkhani [1,*,†], **Nasour Bagheri** [2,3,†] **and Mahyar Shariat** [1,†]

[1] Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran 16788-15811, Iran; m.shariat@sru.ac.ir
[2] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran 16788-15811, Iran; Nbagheri@sru.ac.ir
[3] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran 19538-33511, Iran
* Correspondence: Safkhani@sru.ac.ir; Tel.: +98-21-229-70117
† These authors contributed equally to this work.

Received: 7 July 2018; Accepted: 17 August 2018; Published: 21 August 2018

Abstract: Passive Radio Frequency IDentification (RFID) tags are generally highly constrained and cannot support conventional encryption systems to meet the required security. Hence, designers of security protocols may try to achieve the desired security only using limited ultra-lightweight operations. In this paper, we show that the security of such protocols is not provided by using rotation functions. In the following, for an example, we investigate the security of an RFID authentication protocol that has been recently developed using rotation function named ULRAS, which stands for an Ultra-Lightweight RFID Authentication Scheme and show its security weaknesses. More precisely, we show that the ULRAS protocol is vulnerable against de-synchronization attack. The given attack has the success probability of almost '1', with the complexity of only one session of the protocol. In addition, we show that the given attack can be used as a traceability attack against the protocol if the parameters' lengths are an integer power of 2, e.g., 128. Moreover, we propose a new authentication protocol named UEAP, which stands for an Ultra-lightweight Encryption based Authentication Protocol, and then informally and formally, using Scyther tool, prove that the UEAP protocol is secure against all known active and passive attacks.

Keywords: RFID; ULRAS; UEAP; mobile commerce; RR method; authentication; de-synchronization attack; traceability attack

1. Introduction

Today, many researchers are trying to develop systems that use mobile phones to reach beyond the boundaries of communications and convert a mobile device into a remote authenticator device or a remote control switch. We regularly use computers, mobile phones, and other smart communication systems as devices for electronic interactions, bank payments and pay bills remotely. All of these technologies, in order to provide comfort for their users, are seeking security and preserving privacy. To address this requirement, a lot of authentication protocols have been proposed for such environments. Some of the protocols' designers have designed their protocols using rotation operations to retain the protocol's ultra-weight.

RFID is one of the technologies that is often used in these devices, which identifies objects by using radio waves. RFID has three main components including tags, readers, and a back-end database. Tags are small electronic chips which connected to a product, an object, or a person that we aim to track or authenticate it. Readers, which can be implemented in our cell phones, tablets and etc., are electronic equipment that detect the presence of the tags in an environment and they retrieve the information stored in the tags. The back-end database which stores the extra information about the readers and the tags can be integrated with the reader in our cell phones or similar communication devices or on the separate server outside these devices.

There are two important issues in the RFID systems: Identification and Authentication. Identification means that the reader or tag can identify each other. When the reader broadcasts the query signals to identify or search a special tag, it is possible more than one tag receives the reader's request and replies simultaneously, where their data collide on the reader side with each other and the collision occurs and data is destroyed. This is also the case for readers. If two or more requests arrive to a particular tag from two or more readers, the collision will occur and the data will be destroyed. So there are three kinds of collisions: The tag-tag collision, the reader-reader collision and the tag-reader collision. To counter this problem, anti-collision algorithms have been introduced which have their own literature, e.g., [1–11]. There are many issues in the field of anti-collision in RFID systems which researchers try to solve, e.g., increasing the number of read tags by the reader. Since the efficiency of RFID systems depends on the number of tags read at a specific time, much effort is being made to increase the number of tags that are read by the reader [5,10,11]. Once the tag or the reader has been successfully identified, in the next step it should be authenticated, in order to solve the RFID security issues. In this phase, which is known as the authentication phase of their communication, the rest of the readers and the tags in the vicinity are remaining-silent, to avoid collision. It should be noted in this paper that we assume the reader and the tag are using proper anti-collision protocol and our concentration is on the authentication phase of a reader to a tag communication.

Authentication protocols are protocols that ensure that the parties involved in the protocol are the same as they claim, but the identification protocols do not provide that assurance. The authentication protocols can be one-way, that is, in the course of the process they are assured of one's identity, or they can be mutual, that is to say, they must ensure the identity of the parties during execution.

Problem Definition: Assuming that a reader and a tag decided to communicate in the identification phase of their communication, to provide the security of RFID users, security protocols are also required. Security protocols, such as authentication protocols, are expected to provide the CIA triangle of security which is Confidentiality, Integrity, and Availability. Confidentiality means all of the secret information of protocols' parties must be kept secret. To contradict this property, secret disclosure attack and traceability attack were proposed. Integrity means the adversary cannot change and control protocol messages without the protocol parties' notice. Impersonation attacks can contradict integrity property. Availability means the protocols' parties can authenticate each other at any time and be synchronized with each other. De-synchronization attacks can contradict this property, e.g., by blocking protocol messages or forcing protocols parties to update their shared secret values to different values, where the protocols' parties do not authenticate each other any more and availability of service is destroyed.

Many protocols have been proposed in the literature [12–15] that have attempted to address CIA security principles, but unfortunately, there have been several reports of attacks [16–23] against them that indicate they have failed to provide the desired security. Hence, efforts to design a secure protocol are still ongoing and the new attacks that are developing provide designers with new insight on how to (not) design a protocol. In this way, these attacks and security analyses have contributed to the development of the protocols.

Our contributions: The contributions of this paper are summarized as follows:

- We show that the ULRAS protocol [24], a protocol which has been designed based on rotation function, is not secure and fixing the security problem by any particular mode of rotation function may not be possible.
- An improved protocol named UEAP has also been proposed using lightweight encryption functions in which the ULRAS protocol's security pitfalls are solved.

- The security proof of the UEAP protocol has been done through an informal way and also a formal way through Scyther tool.

In fact, in this paper, we show that the ULRAS protocol, consistent with the SASI protocol [12] and the Gossamer protocol [13], is not secure. Precisely, we present a de-synchronization attack against ULRAS protocol. Hence, employing it in any application is not recommended. In this regard, by using the ULRAS protocol as an example, we show that designing a secure protocol using only the rotation operation without the use of cryptography primitives is not possible.

Paper's organization: The rest of this paper is structured as follows: Section 2 introduces required preliminaries including a brief review of rotation-based RFID authentication protocols and the explanation of the ULRAS protocol. We present the security analysis of the protocol in Section 3. We proposed an improved protocol in Section 4 and its security evaluation is explained in Section 4.1. Finally, we conclude the paper in Section 5.

2. Preliminaries

In this section, we introduce the preliminaries used in this manuscript, as well as the work already done in this field and also the ULRAS protocol as an example for rotation-based RFID authentication protocol.

2.1. The Adversary Model

As our assumption, which is used in this paper, the adversary is an active man in the middle adversary who can eavesdrop, modify or block any transferred message between the tag and the reader. The adversary can also do reasonable amounts of offline computations.

2.2. Related Work

A rotation-based protocol is a protocol for which most of the operations performed on the parties involved in the protocol are rotation operations, combined with other ultra-lightweight operations, e.g., bitwise operations such as AND, OR and XOR, and no cryptographic primitives are used in them.

Designing an RFID authentication protocol based on rotation function began with the SASI protocol [12]. However, soon after there were attacks such as [16–19] that revealed that the protocol was not safe against various attacks. After that, Peris et al. tried to improve the disadvantages of SASI protocol to provide resistance against traceability and de-synchronization attacks, which led to proposing the Gossamer protocol [13]. However, it has been shown in [20] that the Gossamer protocol is vulnerable against denial of service, de-synchronization attack, and replay attacks. Tewari and Gupta in [14], following the method used by previous protocols, proposed another rotation based protocol. This time, the reports such as [21,22] were released on the vulnerability of this protocol against various attacks. Another example is ULRMAPC protocol [15] which [23] proved its vulnerability against DoS, impersonation and de-synchronization attacks.

Recently, in this regard, an ultra-lightweight authentication protocol named ULRAS was proposed by Fan et. al. [24]. The designers of ULRAS have claimed that because of using a special rotation operation, called the RR method, and dividing the protocol secret key into four sub-keys, to update the secret key, their protocol provides forward security and resists against the known active and passive attacks, e.g., de-synchronization (DoS) attack. However, Aghili and Mala in [25], presented reader impersonation attack and secret disclosure attack against the ULRAS protocol and then proposed a new improved protocol.

In this paper, we will present in more depth security analysis of ULRAS protocol [24] and its improvement, proposed by Aghili and Mala [25], and show that, same as their predecessors, they are also vulnerable.

The long history of rotation function based protocol's vulnerabilities and also the current analysis have shown that designing an ultra-lightweight protocol which satisfies all desired security targets may not be feasible. On the other hand, recent advances in symmetric cryptography provided many secure primitives that could be implemented in a constrained environment such as passive RFID tags. For example, implementation of SIMON96/96 [26], which provides 96 bits security and its block length is also 96 bits, only needs 955 NAND gates equivalent (GE). Hence, we suggest employing such cryptographically-sound primitives in designing a protocol rather than attempting to design a secure ultra-lightweight protocol.

2.3. The ULRAS Protocol

The designers of ULRAS only use exclusive-or operation $\oplus$ and a special left rotation operation called RR method in the structure of their protocol, inspired by Gossamer protocol [13]. In the RR method, to compute the left rotation of X by using variable Y, which is of the same length, i.e., $RR(X, Y)$, one can do as follows:

- presents X and Y in their binary forms;
- computes $X' = Reverse(X, Y)$, which inverses only those bits of X for which their correspondence bit-place in Y are "1";
- computes $RR(X, Y)$ as $Rot(X', Y)$ which is the left rotation of X' by amount of $Y \ mod \ L$, where L is the length of X and Y.

In this section, we give a brief description of the ULRAS protocol, where we follow the notations that are represented in Table 1. While the designers [24] have used "$Rot(X, Y)$ through RR method" to denote $RR(X, Y)$, in our description, we use $RR(X, Y)$ for the sake of simplicity. As shown in Figure 1, the ULRAS protocol runs as below:

1. The reader starts the protocol by generating and sending a random time stamp T_R and Query to the tag.
2. The tag, once received the message, verifies whether $T_R \overset{?}{>} T_t$. If $T_R > T_t$, the tag:
    - generates a random number R_t;
    - calculates M_1 as below:
      $M_1 = RR(RR(ID \oplus K \oplus R_t \oplus T_R, ID + R_t), K \oplus R_t)$;
    - and sends IDS, M_1 and R_t to the reader.
3. Upon reception of the message, the reader sends IDS, M_1, R_t and T_R to the back-end database.
4. Once the back-end database receives the message, it verifies whether the received IDS matches with IDS_{new} or IDS_{old}. If the back-end database does not find any match, stops the protocol; otherwise, the database:
    - calculates $M'_1 = RR(RR(ID \oplus K_X \oplus R_t \oplus T_R, ID + R_t), K_X \oplus R_t)$ which X is *new* or *old*. Then it verifies whether $M'_1 \overset{?}{=} M_1$. If $M'_1 \neq M_1$, the back-end database stops the protocol; otherwise, it does as follows:
        - authenticates the tag;
        - generates $i_{sub} \in \{1, 2, 3, 4\}$ and computes M_2 and M_3 as below:
          $M_2 = RR(RR(ID \oplus R_t \oplus T_R, ID \oplus R_t), K_X + R_t)$;
          $M_3 = RR(i_{sub} \oplus K_X, K_X \oplus R_t \oplus T_R)$;
        - generates sub-key as below:
          $subkey = Rot(K_X(i_{sub}), K_X \oplus R_t \oplus T_R)$;
        - updates its values as below:
          $IDS_{old} = IDS_{new}$;
          $K_{old} = K_{new}$;
          $IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$;

K_{new} is generated by replacing $K_{i_{sub}}$;

- and sends M_2 and M_3 through reader to the tag.

5. Upon receipt of the messages, the tag calculates $M_2' = RR(RR(ID \oplus R_t \oplus T_R, ID \oplus R_t), K + R_t)$ with its local values and then verifies whether $M_2' \overset{?}{=} M_2$. If M_2'=M_2, the tag:

- successfully authenticates the back-end server;
- extracts i_{sub} from M_3;
- generates new sub-key as $subkey = Rot(K(i_{sub}), K \oplus R_t \oplus T_R)$;
- and finally updates its IDS, K and T_t as below:
  $IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$;
  K_{new} is generated by replacing $K_{i_{sub}}$.

Table 1. Notations used in this paper.

Notation	Description
RFID	Radio Frequency Identification
IoT	Internet of Things
SD	Secret Disclosure
DA	De-synchronization Attack
IA	Impersonation Attack
TA	Traceability Attack
IDS_{old}	The last time used index number
IDS_{new}	This time successful used of index number
K	The tag's key which is divided to four sub-keys indexed by i_{sub}
K_{old}	The last successful tag's session key
K_{new}	The current tag's session key
$K(i_{sub})$	The last successful sub-key indexed by i_{sub}
i_{sub}	The number which is used for sub-keys index
T_R	The random time stamp generated by the reader
T_t	The last used time stamp
R_t	The random number that is generated by the tag
$X = X_1X_2 \ldots X_L$	The binary representation of X
$Y = Y_1Y_2 \ldots Y_L$	The binary representation of Y
$\lll$	Left rotation operation
$Rot(X, Y)$	The left rotation of X by amount of Y *mod* L where X and Y are of the same length L
$RoR(X, Y)$	The right rotation of X by amount of Y *mod* L where X and Y are of the same length L
L	The length of protocol parameters
X'	The inverse of X
$X' = Reverse(X, Y)$	The inverse operation of X, where for any bit-place in Y that is "1", the corresponding bit in X is inverted
$RR(X, Y)$	This is RR method which has been presented in [24] to do rotation operation as $RR(X, Y) = Rot(X', Y)$
$\mathcal{T}$	An RFID tag
$E_K(.)/D_K(.)$	The Encryption /Decryption function respectively with the key of K

Aghili and Mala in [25], presented a secret disclosure attack and also reader impersonation attack against ULRAS and then presented the improved version of it and claimed their improvement provides security against various kind of attacks. However, their improvement such as its predecessor is still insecure. Aghili and Mala in their improvement removed RR method and instead used $Rot(X, Y)$. They also slightly modified the messages of the protocol. Because of the close similarity to the ULRAS protocol, we ignore the detailed description of the Aghili and Mala protocol and only provide a brief description of it in Figure 2.

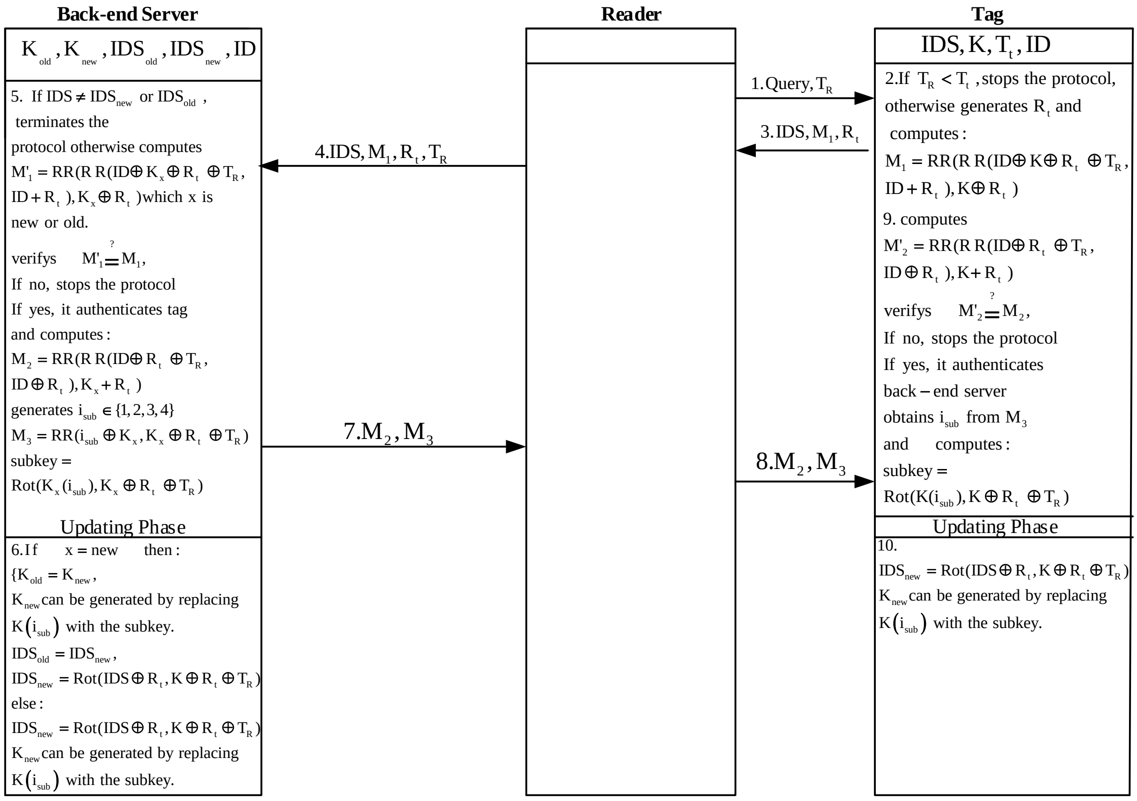

Back-end Server
$K_{old}, K_{new}, IDS_{old}, IDS_{new}, ID$
5. If $IDS \neq IDS_{new}$ or IDS_{old}, terminates the protocol otherwise computes
$M'_1 = RR(RR(ID \oplus K_x \oplus R_t \oplus T_R, ID + R_t), K_x \oplus R_t)$ which x is new or old.
verifys $M'_1 \overset{?}{=} M_1$,
If no, stops the protocol
If yes, it authenticates tag and computes:
$M_2 = RR(RR(ID \oplus R_t \oplus T_R, ID \oplus R_t), K_x + R_t)$
generates $i_{sub} \in \{1, 2, 3, 4\}$
$M_3 = RR(i_{sub} \oplus K_x, K_x \oplus R_t \oplus T_R)$
subkey $= Rot(K_x(i_{sub}), K_x \oplus R_t \oplus T_R)$
Updating Phase
6. If x = new then:
$\{K_{old} = K_{new}$,
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.
$IDS_{old} = IDS_{new}$,
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$
else:
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.
Reader
1.Query, T_R
3.IDS, M_1, R_t
4.IDS, M_1, R_t, T_R
7.M_2, M_3
8.M_2, M_3
Tag
IDS, K, T_t, ID
2.If $T_R < T_t$, stops the protocol, otherwise generates R_t and computes:
$M_1 = RR(RR(ID \oplus K \oplus R_t \oplus T_R, ID + R_t), K \oplus R_t)$
9. computes
$M'_2 = RR(RR(ID \oplus R_t \oplus T_R, ID \oplus R_t), K + R_t)$
verifys $M'_2 \overset{?}{=} M_2$,
If no, stops the protocol
If yes, it authenticates back – end server
obtains i_{sub} from M_3 and computes:
subkey $= Rot(K(i_{sub}), K \oplus R_t \oplus T_R)$
Updating Phase
10.
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.

Figure 1. The ULRAS protocol [24].

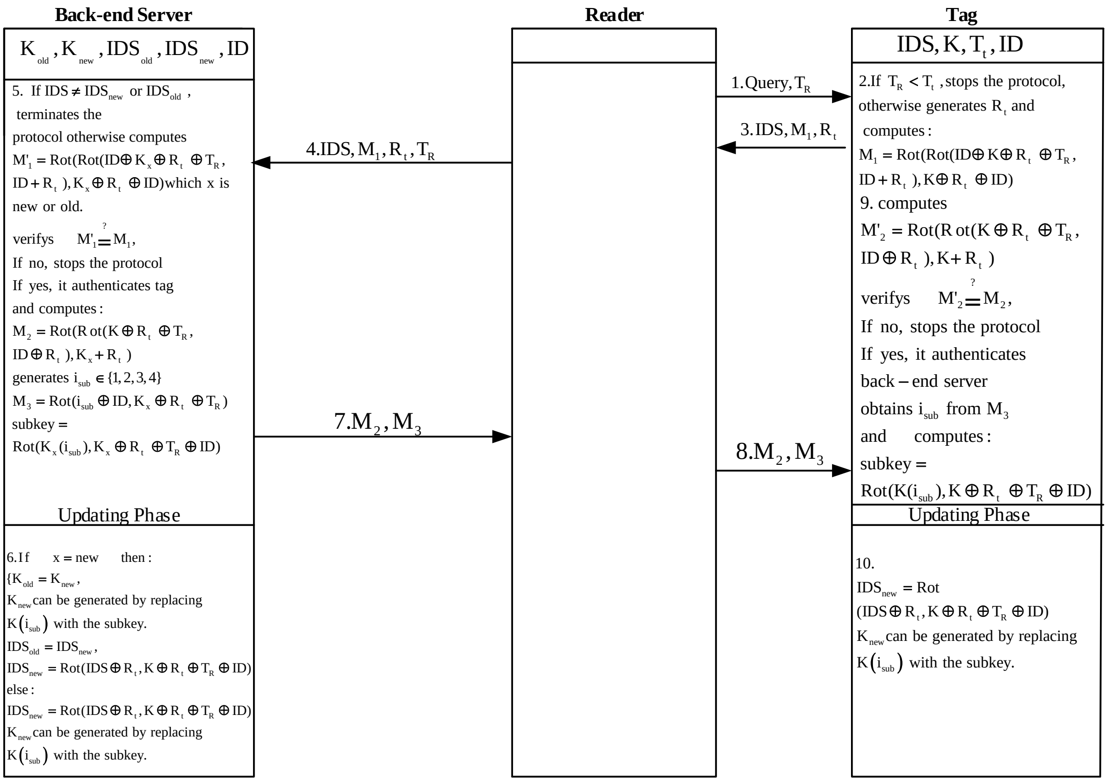

Back-end Server
$K_{old}, K_{new}, IDS_{old}, IDS_{new}, ID$
5. If $IDS \neq IDS_{new}$ or IDS_{old}, terminates the protocol otherwise computes
$M'_1 = Rot(Rot(ID \oplus K_x \oplus R_t \oplus T_R, ID + R_t), K_x \oplus R_t \oplus ID)$ which x is new or old.
verifys $M'_1 \overset{?}{=} M_1$,
If no, stops the protocol
If yes, it authenticates tag and computes:
$M_2 = Rot(Rot(K \oplus R_t \oplus T_R, ID \oplus R_t), K_x + R_t)$
generates $i_{sub} \in \{1, 2, 3, 4\}$
$M_3 = Rot(i_{sub} \oplus ID, K_x \oplus R_t \oplus T_R)$
subkey $= Rot(K_x(i_{sub}), K_x \oplus R_t \oplus T_R \oplus ID)$
Updating Phase
6. If x = new then:
$\{K_{old} = K_{new}$,
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.
$IDS_{old} = IDS_{new}$,
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R \oplus ID)$
else:
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R \oplus ID)$
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.
Reader
1.Query, T_R
3.IDS, M_1, R_t
4.IDS, M_1, R_t, T_R
7.M_2, M_3
8.M_2, M_3
Tag
IDS, K, T_t, ID
2.If $T_R < T_t$, stops the protocol, otherwise generates R_t and computes:
$M_1 = Rot(Rot(ID \oplus K \oplus R_t \oplus T_R, ID + R_t), K \oplus R_t \oplus ID)$
9. computes
$M'_2 = Rot(Rot(K \oplus R_t \oplus T_R, ID \oplus R_t), K + R_t)$
verifys $M'_2 \overset{?}{=} M_2$,
If no, stops the protocol
If yes, it authenticates back – end server
obtains i_{sub} from M_3 and computes:
subkey $= Rot(K(i_{sub}), K \oplus R_t \oplus T_R \oplus ID)$
Updating Phase
10.
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R \oplus ID)$
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.

Figure 2. The Aghili and Mala improvement protocol from ULRAS [25].

3. Security Analysis of ULRAS Protocol

The main observation which we used in our attacks against ULRAS protocol is that the used reverse function in the protocol, i.e., $X' = Reverse(X, Y)$, equals to $X \oplus Y$, as shown by a truth table in Table 2. So, with this equality, we can express $RR(X, Y)$ as $RR(X, Y) = Rot(X', Y) = Rot(Reverse(X, Y), Y) = Rot(X \oplus Y, Y) = (X \oplus Y) \lll (Y \ mod \ L)$, where L is the bit-length of X and Y.

Given that $RR(X, Y) = (X \oplus Y) \lll (Y \ mod \ L)$, in this section, we present our security analysis for ULRAS protocol.

Table 2. The truth table to show the equality of $X' = Reverse(X, Y)$ with $X \oplus Y$.

X	Y	$X' = Reverse(X, Y)$	$X \oplus Y$
0000	1011	1011	1011
0001	1011	1010	1010
0010	1011	1001	1001
0011	1011	1000	1000
0100	1011	1111	1111
0101	1011	1110	1110
0110	1011	1101	1101
0111	1011	1100	1100
1000	1011	0011	0011
1001	1011	0010	0010
1010	1011	0001	0001
1011	1011	0000	0000
1100	1011	0111	0111
1101	1011	0110	0110
1110	1011	0101	0101
1111	1011	0100	0100

3.1. De-Synchronization Attack

A de-synchronization attack is a type of attack for which the adversary tries to do operations that lead to a shared value between protocols' parties to be updated to different values. Therefore, in this case, protocols' parties may not authenticate each other any more and therefore the adversary, by using this attack, can destroy the availability property of security protocols. A security protocol which does not have any of three main security properties, i.e., confidentiality, integrity or availability (or in brief CIA triangle) is not secure and it is not recommended to be used in any sensitive application.

The ULRAS protocol's designers have claimed that, since the reader keeps a history of old shared IDS and K, an adversary cannot de-synchronize the tag and the reader. However, in this section, we present an efficient attack to de-synchronize the tag and the reader. In our attack, the adversary employs the fact that the tag and the reader partially update the key in the last step of the protocol. Hence, if the adversary forces them to update different parts of K, the tag and the reader will be de-synchronized. To do the attack, in a session of the protocol between the legitimate reader and the target tag $\mathcal{T}$, the adversary does as follows:

1. The reader sends T_R and Query to the tag.
2. The tag verifies whether $T_R \overset{?}{>} T_t$, generates R_t, calculates $M_1 = RR(RR(ID \oplus K \oplus R_t \oplus T_R, ID + R_t), K \oplus R_t)$, and sends IDS, M_1 and R_t to the reader.
3. The reader sends IDS, M_1, R_t and T_R to the back-end database.
4. The back-end database verifies the received M_1, authenticates the tag, generates $i_{sub} \in \{1, 2, 3, 4\}$ and computes $M_2 = RR(RR(ID \oplus R_t \oplus T_R, ID \oplus R_t), K_X + R_t)$ and $M_3 = RR(i_{sub} \oplus K_X, K_X \oplus R_t \oplus T_R)$ and sends them to the reader. It then generates $subkey = Rot(K_X(i_{sub}), K_X \oplus R_t \oplus T_R)$ and updates the tag's parameters as below:

$IDS_{old} = IDS_{new}$;
$K_{old} = K_{new}$;
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$;
K_{new} generated by replacing $K_{i_{sub}}$;

5. The adversary, who has eavesdropped T_R, R_t, M_2 and M_3, manipulates M_3 as follows:
    - Assuming $x = K_X \oplus R_t \oplus T_R \ mod\ L$ and given that $M_3 = (i_{sub} \oplus R_t \oplus T_R) \lll x$, because $M_3 = RR(i_{sub} \oplus K_X, K_X \oplus R_t \oplus T_R) = (i_{sub} \oplus K_X \oplus K_X \oplus R_t \oplus T_R) \lll (K_X \oplus R_t \oplus T_R) = (i_{sub} \oplus R_t \oplus T_R) \lll (K_X \oplus R_t \oplus T_R) = (i_{sub} \oplus R_t \oplus T_R) \lll x$, the adversary can determine i_{sub} and also x by knowing M_3 as below:
        - Given that the adversary already has eavesdropped R_t and T_R, she can calculate $R_t \oplus T_R$. On the other hand, i_{sub} has only three bits. Hence, given $R_t \oplus T_R$ and $(i_{sub} \oplus R_t \oplus T_R) \lll x$, it would be easy to determine the values of x and i_{sub}, exclude that the value of $(i_{sub} \oplus R_t \oplus T_R) \lll x$ is rotation invariant which has no high probability and we omit it here for simplicity.
    - Adversary selects $i'_{sub} \in \{1, 2, 3, 4\}/\{i_{sub}\}$ and calculates $M'_3 = M_3 \oplus ((i_{sub} \oplus i'_{sub}) \lll x)$.
6. The adversary sends M_2 and M'_3 to the tag.
7. Upon receipt of the messages, the tag calculates $M'_2 = RR(RR(ID \oplus R_t \oplus T_R, ID \oplus R_t), K + R_t)$ with its local values and then verifies whether $M'_2 \overset{?}{=} M_2$, which it is because the adversary has not changed M_2. Hence, the tag:
    - successfully authenticates the back-end server;
    - gets i'_{sub}, where $i'_{sub} \neq i_{sub}$.
    - generates a new sub-key as $subkey = Rot(K(i'_{sub}), K \oplus R_t \oplus T_R)$;
    - and finally updates its IDS, K and T_t as below:
      $IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$;
      K_{new} generated by replacing $K_{i'_{sub}}$;

In the above attack, the tag updates $K_{i'_{sub}}$ and $i'_{sub} = i_{sub} \oplus \Delta \neq i_{sub}$ while the reader updated $K_{i_{sub}}$. In this attack, if $R_t \oplus T_R$ is not rotation invariant, the adversary's success probability to de-synchronize the tag and the reader would be '1' and its complexity is only one run of protocol and doing some offline computation and sending some messages. It should be noted in the given attack that the tag authenticates the reader and updates its parameters. Hence, keeping a record of old parameters by the back-end server does not prevent this attack and so the ULRAS protocol is not a secure protocol for use.

3.2. Traceability Attack

Traceability attacks often occur when a constant information binded with protocols' parties leak through the exchanged messages over protocol. Now, in this section, we present a traceability attack against the ULRAS protocol which once again shows that this protocol is not secure.

In the de-synchronization attack which was presented in Section 3.1, the adversary can determine x. Given that $x = K_X \oplus R_t \oplus T_R \ mod\ L$ and the adversary knows $R_t \oplus T_R$, x leaks $\log_2$L bits information from K_X, if $L = 2^n$, where n is an integer. In this case, the above de-synchronization attack can be used as a traceability attack on a target tag $\mathcal{T}$, as long as the first quarter of K_X has not been updated. To do this traceability attack, a passive adversary eavesdrops T_R, R_t and M_3 and determines x. Assuming that $i_{sub} \neq 1$ the tag $\mathcal{T}$ will not update the first quarter of K_X, which x depends on. Hence, in the next run of the ULRAS protocol, given a tag $\mathcal{T}'$, the adversary can eavesdrop a session between $\mathcal{T}'$ and the reader $\mathcal{R}$ to determine $\log_2$L bits of the first quarter of K_X and to decide whether $\mathcal{T}' \overset{?}{=} \mathcal{T}$. Here, $\mathcal{T}$ is the target tag which previously adversary eavesdropped its authentication session with the reader and saved its protocol's exchanged messages and $\mathcal{T}'$ is a new tag which adversary wants to know whether it is the target tag. The algorithm of the above attack is also shown in Algorithm 1. The adversary's success probability to trace the tag is '1' and its complexity is only two runs of the protocol and some offline computations.

Algorithm 1: The algorithm of proposed traceability attack against ULRAS protocol

```
Data: $T_R, R_t, M_3 = RR(i_{sub} \oplus K_X, K_X \oplus R_t \oplus T_R) == (i_{sub} \oplus R_t \oplus T_R) \lll x, i_{sub}, i'_{sub} \neq 1$
Result: decides whether $\mathcal{T}' \overset{?}{=} \mathcal{T}$ where $\mathcal{T}$ is an adversary's target tag.
1. Eavesdrops a session between reader and $\mathcal{T}$ and stores
   $T_R, R_t, M_3 = RR(i_{sub} \oplus K_X, K_X \oplus R_t \oplus T_R) == (i_{sub} \oplus R_t \oplus T_R) \lll x$;
2. Obtains $x = K_X \oplus R_T \oplus T_R \; mod \; L$ and $i_{sub}$ by using $M_3$, $T_R$ and $R_t$ and this fact
   $i_{sub} \in \{2, 3, 4\}$;
3. Retrieves $log_2 L$ bits information from $K_X$ by using $x$;
4. Eavesdrops a session between $\mathcal{T}'$ and the reader;
5. Obtains $x' = K'_X \oplus R'_T \oplus T'_R \; mod \; L$ and $i'_{sub}$ by using $M'_3$, $T'_R$ and $R'_t$ and this fact
   $i'_{sub} \in \{2, 3, 4\}$;
6. Retrieves $log_2 L$ bits information from $K'_X$ by using $x'$;
7. Compares the retrieved bits of $K'_X$ with $K_X$ to decide whether $\mathcal{T}' \overset{?}{=} \mathcal{T}$.
```

3.3. Security Analysis of Aghili and Mala Improvement to ULRAS

There are several important points to note about Aghili and Mala's [25] improvement to ULRAS:

- The use of a rotation operation several times is like using one rotation i.e., $M_2 = Rot(Rot(K \oplus R_t \oplus T_R, ID \oplus R_t), K_X + R_t)$ in the Aghili and Mala improvement equals with $M_2 = Rot(K \oplus R_t \oplus T_R, i)$ where i is a value between 0 to L. The same point applies to M_1 message.
- Based on this fact given $M = Rot(X, Y) \; mod \; L$ and X, if we rotate right M for $i = 0, \ldots, L$ and comparing the result with X, one can determine Y, the adversary with eavesdropping two sessions of protocol messages without completion of protocol sessions which leads to not updating secret values, can conduct secret disclosure attack which reveals ID and K. Precisely, given $M_2 = Rot(K \oplus R_t \oplus T_R, i)$ and $M'_2 = Rot(K \oplus R'_t \oplus T'_R, j)$, R_t, R'_t, T_R, T'_R, M_1 and M'_1, the adversary for $i, j = 0, \ldots, L$ verifies whether $RoR(M_2, i) \oplus R_T \oplus T_R \overset{?}{==} RoR(M'_2, j) \oplus R'_T \oplus T'_R$ to retrieve K as $RoR(M_2, i) \oplus R_T \oplus T_R$. Similarly, for $i, j = 0, \ldots, L$ the adversary verifies whether $RoR(M_1, i) \oplus R_T \oplus T_R \overset{?}{==} RoR(M'_1, j) \oplus R'_T \oplus T'_R$ to retrieve $ID \oplus K$ as $RoR(M_1, i) \oplus R_T \oplus T_R$. Given that K has already been acquired, the adversary can get ID and can verify the correctness of the obtained values by using other protocol's messages.
- Since all the secret values of the protocol are revealed, it is easy to do a variety of attacks including impersonation attacks, traceability attacks, de-synchronization attacks, etc.

4. UEAP-Our Proposed Protocol

As shown above, the design of RFID security protocols using the rotation operation does not lead to desired security. Therefore, it seems it is not possible to achieve a secure protocol without the use of cryptographic primitives. There are also lightweight cryptographic primitives such as lightweight block ciphers e.g., Skinny [27], SIMON and SPECK [28] that are suggested to be used to design a secure protocol instead of rotation function, although they are more costly. Using a lightweight block cipher, the disadvantages of the ULRAS authentication protocol are resolved, it is also depicted in Figure 3. We call our improved protocol UEAP, which is the acronym for Ultra-lightweight Encryption based Authentication Protocol:

1. The reader starts the protocol by generating and sending a random time stamp T_R and $Query$ to the tag.
2. The tag, once it receives the message, verifies whether $T_R \overset{?}{>} T_t$. If $T_R > T_t$, the tag:

   - generates a random number R_t;
   - calculates M_1 as $E_K(ID \| R_t \| T_R)$;
   - and sends IDS, M_1 and R_t to the reader.

3. Upon reception the message, the reader sends IDS, M_1, R_t and T_R to the back-end database.
4. Once the back-end database received the message, verifies whether the received IDS matches with IDS_{new} or IDS_{old}. If the back-end database does not find any match, stops the protocol; otherwise, the database:

   - calculates $M_1' = E_{K_X}(ID \| R_t \| T_R)$ which X is *new* or *old*. Then it verifies whether $M_1' \overset{?}{=} M_1$. If $M_1' \neq M_1$, the back-end database stops the protocol; otherwise, it does as follows:
     - authenticates the tag;
     - generates $i_{sub} \in \{1, 2, 3, 4\}$ and computes M_2 and M_3 as below:
$M_2 = E_{K_X \oplus R_t}(ID \| T_R \| K_X)$;
$M_3 = E_{K_X \oplus T_R}((K_X \oplus i_{sub}) \| R_t \| T_R)$;
     - generates sub-key as below:
$subkey = Rot(K_X(i_{sub}), K_X \oplus R_t \oplus T_R)$;
     - updates its values as below:
$IDS_{old} = IDS_{new}$;
$K_{old} = K_{new}$;
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$;
K_{new} is generated by replacing $K_{i_{sub}}$;
     - and sends M_2 and M_3 through the reader to the tag.

5. Upon receipt of the messages, the tag calculates $M_2' = E_{K_X \oplus R_t}(ID \| T_R \| K_X)$ by its local values and then verifies whether $M_2' \overset{?}{=} M_2$. If M_2'=M_2, the tag:

   - successfully authenticates the back-end server;
   - extracts i_{sub} from M_3;
   - generates new sub-key as $subkey = Rot(K(i_{sub}), K \oplus R_t \oplus T_R)$;
   - and finally updates its IDS, K and T_t as below:
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$;
K_{new} is generated by replacing $K_{i_{sub}}$.

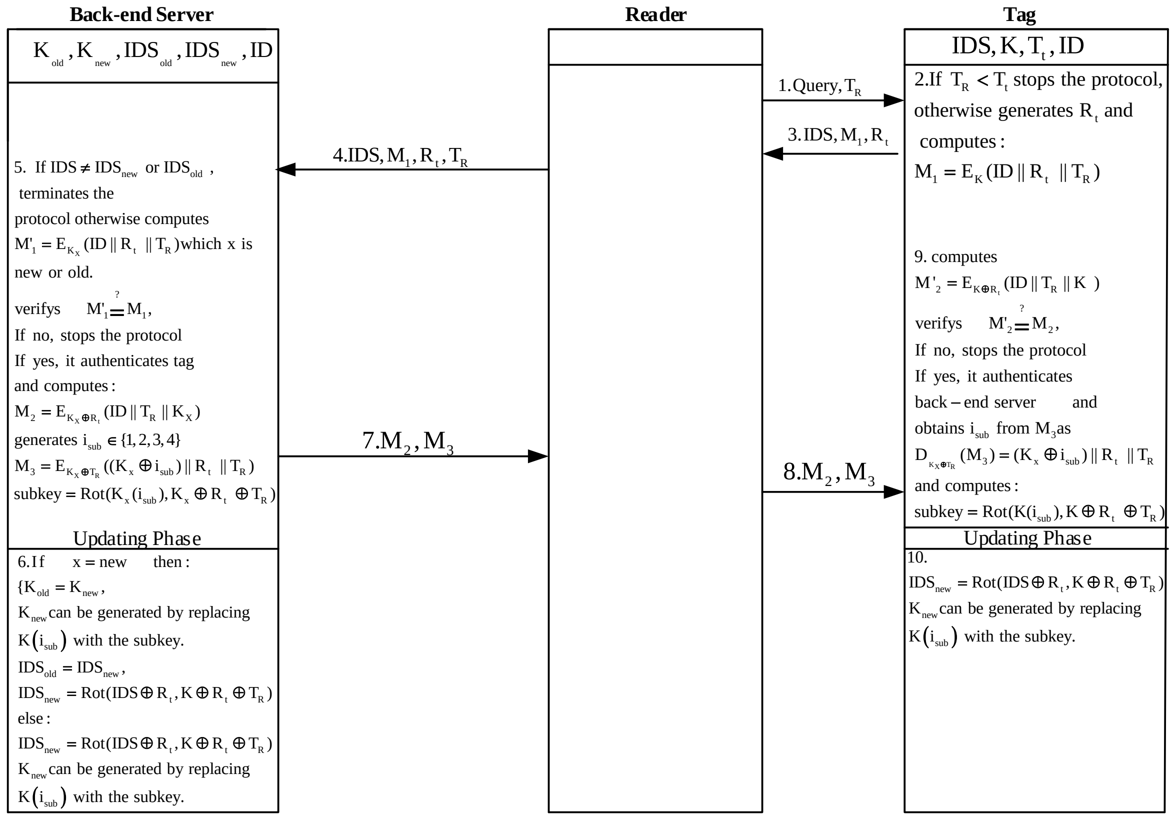

Back-end Server
Reader
Tag
$K_{old}, K_{new}, IDS_{old}, IDS_{new}, ID$
IDS, K, T_t, ID
1.Query, T_R
2.If $T_R < T_t$ stops the protocol, otherwise generates R_t and computes: $M_1 = E_K(ID \| R_t \| T_R)$
3.IDS, M_1, R_t
4.IDS, M_1, R_t, T_R
5. If $IDS \neq IDS_{new}$ or IDS_{old}, terminates the protocol otherwise computes $M'_1 = E_{K_X}(ID \| R_t \| T_R)$ which x is new or old.
verifys $M'_1 \overset{?}{=} M_1$,
If no, stops the protocol
If yes, it authenticates tag and computes:
$M_2 = E_{K_X \oplus R_t}(ID \| T_R \| K_X)$
generates $i_{sub} \in \{1,2,3,4\}$
$M_3 = E_{K_X \oplus T_R}((K_x \oplus i_{sub}) \| R_t \| T_R)$
$subkey = Rot(K_x(i_{sub}), K_x \oplus R_t \oplus T_R)$
7.M_2, M_3
8.M_2, M_3
9. computes $M'_2 = E_{K \oplus R_t}(ID \| T_R \| K)$
verifys $M'_2 \overset{?}{=} M_2$,
If no, stops the protocol
If yes, it authenticates back – end server and obtains i_{sub} from M_3 as
$D_{K_X \oplus T_R}(M_3) = (K_x \oplus i_{sub}) \| R_t \| T_R$
and computes:
$subkey = Rot(K(i_{sub}), K \oplus R_t \oplus T_R)$
Updating Phase
6.If x = new then:
$\{K_{old} = K_{new}$,
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.
$IDS_{old} = IDS_{new}$,
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$
else:
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.
Updating Phase
10.
$IDS_{new} = Rot(IDS \oplus R_t, K \oplus R_t \oplus T_R)$
K_{new} can be generated by replacing $K(i_{sub})$ with the subkey.

Figure 3. The UEAP protocol.

4.1. *Security Evaluation of UEAP*

In this section, we first informally prove that the UEAP protocol can resist against the attacks proposed in this paper and the other known active and passive attacks. Next, we show that the Scyther tool could not find any attack in UEAP.

4.1.1. Informal Security Proof

Resistance against de-synchronization attack: Given that in the UEAP protocol all messages are encrypted, the adversary cannot modify the transferred messages in such a way that the protocol parties exist from synchronization. Any modification in any transferred encrypted message is identified by the tag or the reader and it will terminate the protocol.

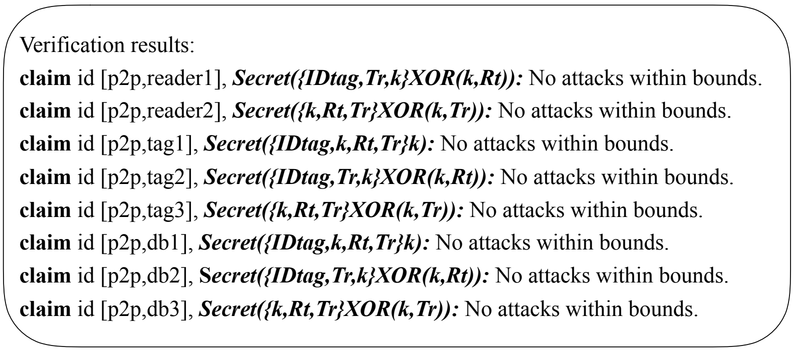
Resistance against traceability attack: The vulnerability of ULRAS protocol was due to the fact that the adversary could retrieve the value of $K_x \oplus R_t \oplus T_R \ mod \ L$. Because of using encryption function in calculating of messages in the UEAP protocol, the adversary cannot determine $K_x \oplus R_t \oplus T_R \ mod \ L$, and so the UEAP protocol is secure against the traceability attack presented in this manuscript.

Resistance against replay and impersonation attacks: All protocols' parties participate in the randomization of the messages exchanged in the UEAP protocol, and also all the messages exchanged are encrypted. Hence, the adversary cannot use a message later or fake a message on his behalf. Therefore, the UEAP protocol resists all types of replay and impersonation attacks.

4.1.2. Formal Security Proof

Scyther [29] is an automatic tool for security analysis of security protocols which can be used to check the security problems of protocols. In Scyther tool, entire possible behaviors of a protocol are predicted and let us know the possible attacks on the protocol and also let us know whether the security claims of the protocol are provided or not. Security claims are essential components of the security protocols. To evaluate the security of the protocol by the Scyther tool, first, we must write the protocol description in spdl language. Then, the Scyther tool verifies whether the defined security claims of the protocol are satisfied or not, and also the Scyther has this ability to define appropriate security claims of protocol automatically and then verifies them. The Scyther tool also let us interpret the principles and properties of security in the language of security claims, and then we can check whether these claims were either satisfied or violated. Precisely, the Scyther tool checks security claims of secrecy and authentication. The secrecy, which means keeping a certain data secret and confidential, and authentication should exist between communication parties [29].

In this section, we analyze the UEAP protocol with the Scyther tool. The output results of the Scythe tool for the UEAP protocol are presented in Figure 4. As it can be seen, this analysis with the Scyther tool showed that the UEAP protocol is resistant to defined threats.



```
Verification results:
claim id [p2p,reader1], Secret({IDtag,Tr,k}XOR(k,Rt)): No attacks within bounds.
claim id [p2p,reader2], Secret({k,Rt,Tr}XOR(k,Tr)): No attacks within bounds.
claim id [p2p,tag1], Secret({IDtag,k,Rt,Tr}k): No attacks within bounds.
claim id [p2p,tag2], Secret({IDtag,Tr,k}XOR(k,Rt)): No attacks within bounds.
claim id [p2p,tag3], Secret({k,Rt,Tr}XOR(k,Tr)): No attacks within bounds.
claim id [p2p,db1], Secret({IDtag,k,Rt,Tr}k): No attacks within bounds.
claim id [p2p,db2], Secret({IDtag,Tr,k}XOR(k,Rt)): No attacks within bounds.
claim id [p2p,db3], Secret({k,Rt,Tr}XOR(k,Tr)): No attacks within bounds.
```

Figure 4. The result of UEAP protocol 's security analysis with Scyther.

4.2. Comparison

In this section, we compare the UEAP protocol with some recent rotation based authentication protocols from the security and also computational costs point of views. As can be seen in Table 3, all rotation-based protocols are vulnerable against one or more attacks while UEAP protocol which uses a lightweight encryption function is secure. However, as it is shown in Table 4, it costs to implement an encryption/decryption function, although this is a cost we should pay to achieve a promising security.

Table 3. Security comparison of the UEAP protocol with other protocols, where SD, DA, IA, TA, ✓ and × denote Secret Disclosure Attack, De-synchronization Attack, Impersonation Attack, Traceability Attack, Secure and Vulnerable respectively.

Protocol	**SD**	**DA**	**IA**	**TA**
SASI [12]	× [18,30]	× [19]	× [18,30]	× [16]
Gossamer [13]	✓	× [20]	× [20]	× [20]
ULRMAPC [15]	× [23]	✓	×[23]	×[23]
Tewari and Gupta [14]	× [21,22]	×[21,22]	× [21,22]	×[21,22]
ULRAS [24]	×(in this paper,[25])	✓	×[25]	×(in this paper)
Aghili and Mala [25]	×(in this paper)	×(in this paper)	×(in this paper)	×(in this paper)
UEAP	✓	✓	✓	✓

Table 4. Computational cost comparison of the UEAP protocol with other protocols, where L denotes the length of each parameter in protocols

Protocol	♯ of $\oplus$	♯ of $Rot(X, Y)$	♯ of $E_K(X)/D_K(X)$	♯ of Transferred Bits
SASI [12]	20 L	4	-	6 L
Gossamer [13]	12 L	36	-	6 L
Tewari and Gupta [14]	24 L	12	-	7 L
ULRMAPC [15]	34 L	14	-	11 L
ULRAS [24]	30 L	14	-	13 L
Aghili and Mala [25]	36 L	14	-	13 L
UEAP	16 L	4	6	13 L

5. Conclusions

In this paper, we analyzed the security of a rotation-based ultra-lightweight authentication protocol which has been recently proposed for mobile applications. We presented an efficient de-synchronization attack against this protocol and extended it to a traceability attack when the parameter length is an integer power of 2. Although it is possible to present several other attacks against the protocol, we just mentioned our most efficient attacks in this paper, which is enough to contradict the designers' claims on the security of this protocol. We also extend the attack against its improved version which has been introduced by Aghili and Mala.

Moreover, we presented a new lightweight RFID authentication protocol named UEAP using lightweight encryption functions and also its security proof which showed that the proposed protocol is safe against all types of active and passive attacks.

This paper once again showed that the design of a secure protocol based on rotation operation may not be possible, and hence the use of lightweight cryptographic primitives in the design of the security protocols is inevitable.

Author Contributions: M.S. (Masoumeh Safkhani) and N.B. created the main contributions and wrote the paper; M.S. (Mahyar Shariat) did the verification of the proposed protocol through the Scyther tool.

Funding: This work was supported by Shahid Rajaee Teacher Training University.

Acknowledgments: We would like to thank the anonymuse reviewers for their suggestions on improving the presentation of the paper and also their technical suggestions.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Arjona, L.; Landaluce, H.; Perallos, A.; Onieva, E. Timing-Aware RFID Anti-Collision Protocol to Increase the Tag Identification Rate. *IEEE Access* **2018**, *6*, 33529–33541. [CrossRef]
2. Saadi, H.; Touhami, R.; Yagoub, M.C.E. TDMA-SDMA-based RFID algorithm for fast detection and efficient collision avoidance. *Int. J. Commun. Syst.* **2018**, *31*, e3392. [CrossRef]
3. Liu, B.; Su, X. An Anti-Collision Algorithm for RFID Based on an Array and Encoding Scheme. *Information* **2018**, *9*, 63. [CrossRef]
4. Arjona, L.; Landaluce, H.; Perallos, A. Energy-Aware RFID Anti-Collision Protocol. *Sensors* **2018**, *18*, 1904. [CrossRef] [PubMed]
5. Memon, M.Q.; He, J.; Yasir, M.A.; Memon, A. Improving Efficiency of Passive RFID Tag Anti-Collision Protocol Using Dynamic Frame Adjustment and Optimal Splitting. *Sensors* **2018**, *18*, 1185. [CrossRef] [PubMed]
6. Tan, X.; Wang, H.; Fu, L.; Wang, J.; Min, H.; Engels, D.W. Collision Detection and Signal Recovery for UHF RFID Systems. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 239–250. [CrossRef]
7. Zhang, L.; Xiang, W.; Tang, X.; Li, Q.; Yan, Q. A Time- and Energy-Aware Collision Tree Protocol for Efficient Large-Scale RFID Tag Identification. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2406–2417. [CrossRef]

8. Rezaie, H.; Golsorkhtabaramiri, M. A fair reader collision avoidance protocol for RFID dense reader environments. *Wirel. Netw.* **2018**, *24*, 1953–1964. [CrossRef]
9. Su, J.; Sheng, Z.; Xie, L. A Collision-Tolerant-Based Anti-Collision Algorithm for Large Scale RFID System. *IEEE Commun. Lett.* **2017**, *21*, 1517–1520. [CrossRef]
10. Liu, B.H.; Nguyen, N.T.; Pham, V.T.; Yeh, Y.H. A maximum-weight-independent-set-based algorithm for reader-coverage collision avoidance arrangement in rfid networks. *IEEE Sens. J.* **2016**, *16*, 1342–1350. [CrossRef]
11. Nguyen, N.T.; Liu, B.H.; Pham, V.T. A dynamic-range-based algorithm for reader-tag collision avoidance deployment in rfid networks. In Proceedings of the 2016 International Conference on Electronics, Information, and Communications (ICEIC), Danang, Vietnam, 27–30 January 2016; pp. 1–4.
12. Chien, H.Y. Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity. *IEEE Trans. Dependable Secur. Comput.* **2007**, *4*, 337–340. [CrossRef]
13. Peris-Lopez, P.; Hernandez-Castro, J.C.; Tapiador, J.M.E.; Ribagorda, A. Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protoco. In *Information Security Applications*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 56–68.
14. Tewari, A.; Gupta, B.B. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. *J. Supercomput.* **2017**, *73*, 1085–1102. [CrossRef]
15. Fan, K.; Gong, Y.; Liang, C.; Li, H.; Yang, Y. Lightweight and ultralightweight RFID mutual authentication protocol with cache in the reader for IoT in 5G. *Secur. Commun. Netw.* **2016**, *9*, 3095–3104. [CrossRef]
16. Phan, R.C.W. Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI. *IEEE Trans. Dependable Secur. Comput.* **2009**, *6*, 316–320. [CrossRef]
17. Cao, T.; Bertino, E.; Lei, H. Security analysis of the SASI protocol. *IEEE Trans. Dependable secur. Comput.* **2009**, *6*, 73–77.
18. Hernandez-Castro, J.C.; Tapiador, J.M.E.; Peris-Lopez, P.; Quisquater, J.J. Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations. *arXiv* **2008**, arXiv:0811.4257.
19. Sun, H.M.; Ting, W.C.; Wang, K.H. On the security of Chien's ultralightweight RFID authentication protocol. *IEEE Trans. Dependable Secur. Comput.* **2011**, *8*, 315–317. [CrossRef]
20. Bilal, Z.; Masood, A.; Kausar, F. Security analysis of ultra-lightweight cryptographic protocol for low-cost RFID tags: Gossamer protocol. In Proceedings of the 2009 International Conference on Network-Based Information Systems, Indianapolis, IN, USA, 19–21 August 2009; pp. 260–267.
21. Safkhani, M.; Bagheri, N. Passive secret disclosure attack on an ultralightweight authentication protocol for Internet of Things. *J. Supercomput.* **2017**, *73*, 3579–3585. [CrossRef]
22. Wang, K.H.; Chen, C.M.; Fang, W.; Wu, T.Y. On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags. *J. Supercomput.* **2018**, *74*, 65–70. [CrossRef]
23. Aghili, S.F.; Ashouri-Talouki, M.; Mala, H. DoS, impersonation and de-synchronization attacks against an ultra-lightweight RFID mutual authentication protocol for IoT. *J. Supercomput.* **2018**, *74*, 509–525. [CrossRef]
24. Fan, K.; Ge, N.; Gong, Y.; Li, H.; Su, R.; Yang, Y. An ultra-lightweight RFID authentication scheme for mobile commerce. *Peer-to-Peer Netw. Appl.* **2016**, *10*, 1–9. [CrossRef]
25. Aghili, S.F.; Mala, H. Security Analysis of an Ultra-lightweight RFID Authentication Protocol for M-commerce. *IACR Cryptol. ePr. Archiv.* **2017**, *2017*, 547.
26. Beaulieu, R.; Shors, D.; Smith, J.; Treatman-Clark, S.; Weeks, B.; Wingers, L. The SIMON and SPECK lightweight block ciphers. In Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, 7–11 June 2015; p. 175.
27. Beierle, C.; Jean, J.; Kölbl, S.; Leander, G.; Moradi, A.; Peyrin, T.; Sasaki, Y.; Sasdrich, P.; Sim, S.M. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *Advances in Cryptology—CRYPTO 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 123–153.
28. Beaulieu, R.; Treatman-Clark, S.; Shors, D.; Weeks, B.; Smith, J.; Wingers, L. The SIMON and SPECK lightweight block ciphers. In Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.

29. Cremers, C.J.F. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In *Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 414–418.
30. Avoine, G.; Carpent, X.; Martin, B. Strong authentication and strong integrity (SASI) is not that strong. In *Radio Frequency Identification: Security and Privacy Issues*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 50–64.