Future Generation Computer Systems 101 (2019) 621–634

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

SecLAP: Secure and lightweight RFID authentication protocol for Medical IoT

Seyed Farhad Aghili [a], Hamid Mala [a,*], Pallavi Kaliyar [b], Mauro Conti [b]

[a] Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib St., Isfahan 81746-73441, Iran
[b] Department of Mathematics, University of Padova, Padova, Italy

HIGHLIGHTS

- We identify several serious security vulnerabilities against LRMI scheme. Our proposed attacks include secret disclosure, reader impersonation, and tag traceability attacks. Moreover, we show that despite the authors claim, the protocol fails to protect the privacy of tag and reader.
- We overcome the vulnerabilities of LRMI by proposing a lightweight authentication scheme (called SecLAP). Besides, a lightweight modular rotate function called $MRot_{(K)}(x,y)$ is proposed to use in SecLAP.
- The security of SecLAP is examined from a formal and informal analysis by considering attacks we found in LRMI scheme and in the other existing common schemes known for IoT networks.
- To show the efficiency of SecLAP for resource constrained tags, we fully implement and evaluate (through simulation results) the $MRot_{(K)}(x,y)$ employed in this scheme.

ARTICLE INFO

Article history:
Received 19 February 2019
Received in revised form 16 June 2019
Accepted 4 July 2019
Available online 9 July 2019

Keywords:
RFID
Internet of Things
FPGA
Secret disclosure attack
Impersonation attack
Anonymity

ABSTRACT

The safety of medical data and equipment plays a vital role in today's world of Medical Internet of Things (MIoT). These IoT devices have many constraints (e.g., memory size, processing capacity, and power consumption) that make it challenging to use cost-effective and energy-efficient security solutions. Recently, researchers have proposed a few Radio-Frequency Identification (RFID) based security solutions for MIoT. The use of RFID technology in securing IoT systems is rapidly increasing because it provides secure and lightweight safety mechanisms for these systems. More recently, authors have proposed a lightweight RFID mutual authentication (LRMI) protocol. The authors argue that LRMI meets the necessary security requirements for RFID systems, and the same applies to MIoT applications as well. In this paper, our contribution has two-folds, firstly we analyze the LRMI protocol's security to demonstrate that it is vulnerable to various attacks such as secret disclosure, reader impersonation, and tag traceability. Also, it is not able to preserve the anonymity of the tag and the reader. Secondly, we propose a new secure and lightweight mutual RFID authentication (SecLAP) protocol, which provides secure communication and preserves privacy in MIoT systems. Our security analysis shows that the SecLAP protocol is robust against de-synchronization, replay, reader/tag impersonation, and traceability attacks, and it ensures forward and backward data communication security. We use Burrows–Abadi–Needham (BAN) logic to validate the security features of SecLAP. Moreover, we compare SecLAP with the state-of-the-art and validate its performance through a Field Programmable Gate Array (FPGA) implementation, which shows that it is lightweight, consumes fewer resources on tags concerning computation functions, and requires less number of flows.

1. Introduction

Nowadays, the Internet of Things (IoT) intends to connect a large number of smart devices through the Internet. From different IoT applications such as home automation, transportation, and health-care industry, the IoT has become an essential part of our day to day lives. All smart devices are equipped with

* Corresponding author.
E-mail addresses: sf.aghili@eng.ui.ac.ir (S.F. Aghili), h.mala@eng.ui.ac.ir (H. Mala), pallavi@math.unipd.it (P. Kaliyar), conti@math.unipd.it (M. Conti).

https://doi.org/10.1016/j.future.2019.07.004
0167-739X/

sensors to sense, collect, and transmit the data from their relevant surroundings. These devices have a unique identification number through which they connect to the Internet and transfer data from one place to another. Due to the integration of IoT in health-care devices, a large number of companies are making considerable investments in the health-care domain. The use of sensors in medical devices provide many benefits such as remote and continuous monitoring of a patient's health, and real-time disease management which lowers the cost of care, and it also improves the quality of life of pediatric and aged populations. The Medical Internet of Things (MIoT) uses the intelligent system, in which devices can obtain patient data and transfer it via a gateway to the secure cloud-based platforms where the data gets stored, aggregated, and analyzed. These systems are beneficial in storing the data of thousands of patients, and it also provides a real-time analysis of the patient's stored information to improve the efficiency of the health-care industry.

Since the patient's information is highly sensitive and vital, the misuse of this data may lead to dangerous situations, and it can even cause a complete system failure. Therefore, it is mandatory to apply proper security mechanisms to secure these smart medical devices when they are sensing and transmitting the data. As these smart medical devices sense, collect and transmit data frequently of different types and sizes, memory constraint is always a stiff challenge for these tiny devices. In today's world, the approach of cryptography for data security is moving from Rivest–Shamir–Adleman (RSA) to Elliptic Curve Cryptography (ECC) [1]. The reason behind it is the lightweight computation nature of the ECC algorithm, which is suitable for the resource constraint smart devices. Nowadays, one of the leading wireless technologies that are used for IoT deployments in the health-care field is Radio-Frequency Identification (RFID).

RFID is a short-range communication technology, which is being highly used as an alternative for barcodes in identifying tagged objects. An RFID system consists of a tag and a reader, and also requires a server when involved in a large capacity of calculation and information. The tag usually contains private and essential information about the products, and the reader tries to establish a connection with the tag all the time to extract the information from it. Free-space information transfer in an RFID system makes it vulnerable against security threats such as eavesdropping. Considering the resource limitations of tags, the advanced crypto primitives such as AES, DES, RSA, and SHA1 cannot be used. Therefore, lightweight cryptography techniques such as lightweight hash functions and logical functions (e.g., XOR and bit rotation) are used in the majority of suggested protocols [2]. Traditionally, an RFID system involves two recommended communication channels:

- The channel between the tag and reader can be in two states, namely, forward channel (reader to tag) and backward channel (tag to reader). The important challenge for these channels is eavesdropping of the messages.
- The channel between the reader and the server formed during their communication with each other is not fully secure.

Fig. 1 depicts a typical architecture of RFID based medical systems, which consists of the following three entities: (i) server, (ii) reader, and (iii) tag. Fig. 1 demonstrates how communication takes place between these three entities. The reader communicates through tags which can be implanted on the patient body, different kinds of medical devices, and medicines. The readers get the information from the tags and then pass it to the servers for record management and other operations. From multiple existing servers, the information moves to the cloud server, which provides real-time access to the information for doctors, patients, and hospital management. This process makes the whole medical system more accessible and usable. Now, if the communication channel between these entities is corrupted or if there is an adversary who can access this information, then the whole concept of the smart medical system becomes purposeless. That is why providing the entity authentication before transmitting any critical information is one of the main concerns of the medical systems. Hence, we solve this problem by proposing the SecLAP authentication protocol.

Recently, the authors in [3] proposed a lightweight RFID protocol for MIoT systems, called LRMI. The authors use a rotation function which is a circular shift on the input string and claimed that their scheme can satisfy the security properties (such as consistency, synchronization, and tag anonymity) necessary for RFID systems and the proposal is suitable for IoT scenarios. In this paper, we show that their protocol has several vulnerabilities regarding security and anonymity.

1.1. Contribution

In this paper we make the following contributions.

- We identify several vulnerabilities against the LRMI scheme [3], including secret disclosure, reader impersonation, and tag traceability attacks. Moreover, we show that despite the author's claim, the protocol fails to protect the privacy of tag and reader.
- To overcome the vulnerabilities of LRMI, we enhance the security of their scheme by proposing a lightweight authentication scheme (called SecLAP). Besides, a lightweight modular rotate function called $MRot_{(K)}(x, y)$ is proposed to use in SecLAP.
- The security of SecLAP is evaluated from a formal and informal analysis by considering attacks we found in LRMI scheme and the other existing common schemes known for IoT networks.
- To show the efficiency of SecLAP for resource-constrained tags, we fully implement and evaluate (through simulation results) the $MRot_{(K)}(x, y)$ employed in this scheme.

1.2. Organization

The rest of the paper is structured as follow. The related work is briefly introduced in Section 2. In Section 3, we first briefly explain about the RFID based medical system architecture and rotation functions, and later the preliminaries and notations that we have used in this paper. We briefly describe about LRMI [3] protocol in Section 4. We analyze the security of LRMI [3] protocol in Section 5 and also discuss the attacks which we found against the LRMI protocol. The description of our proposed protocol (i.e., SecLAP) is presented in Section 6. Formal and informal methods in Section 7 analyze the security of the proposed protocol. Section 8 shows the implementation and simulation of our proposed protocol. Finally, in Section 9, we conclude our work.

2. Previous work

In recent years, many authentication protocols have been proposed for RFID systems. For instance, the HB-family (HB, HB^{+}, HB^{++}, etc.) [4–6] by employing matrix multiplication and XOR, and the MAP-family (EMAP, M2AP, LMP^{+}, etc.) [7–9] based on bitwise operations like AND, XOR, and OR are some of the lightweight authentication protocols proposed in the literature. However, these two models have several limitations concerning weaknesses and vulnerabilities [10–14]. Later, authors in [15] proposed a lightweight solution to mutual authentication for RFID systems by using Physically Unclonable Functions (PUFs) and

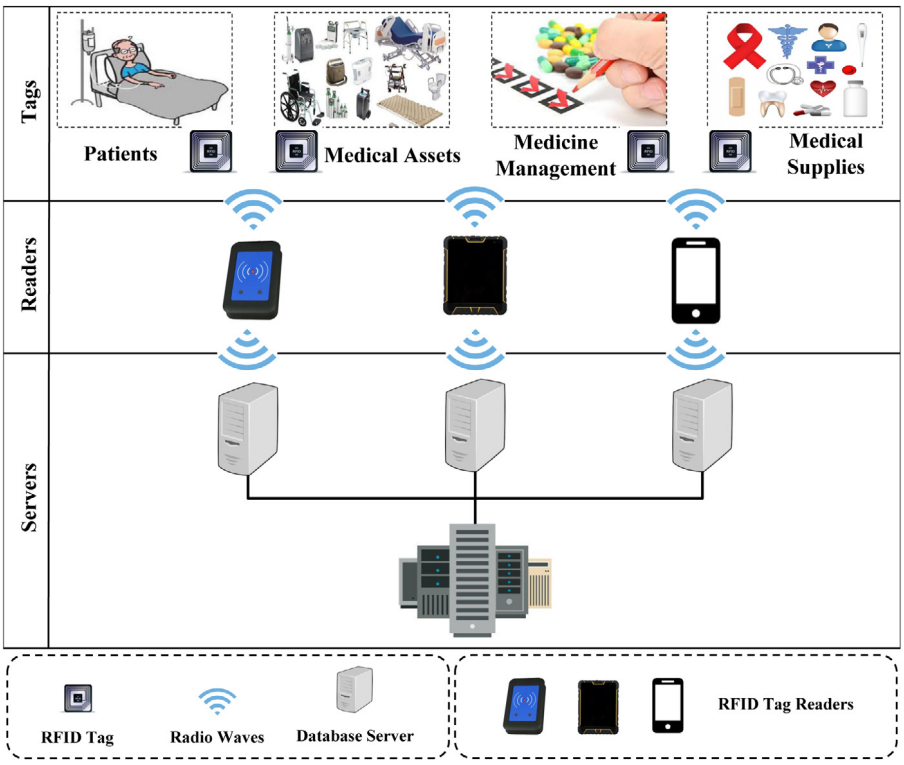

Fig. 1. RFID based Medical System Architecture.

Linear Feedback Shift Registers (LFSRs), which are lightweight functions. However, the authors in [16] showed that the protocol in [15] is not resistant against message injection attack, and it has several vulnerabilities.

To overcome the RFID authentication problems, authors in [17] employed Chebyshev chaotic maps. However, the authors in [18] proposed several attacks like de-synchronization attack and secret disclosure attack against [17] protocol. In 2014, authors in [19] proposed an improvement to overcome these weaknesses. However, authors in [20] showed that the protocol proposed in [19] is vulnerable to tracking, tag impersonation, and de-synchronization attacks. In [21], authors proposed a new authentication protocol for RFID-based IoT systems and claimed that their protocol is secure. However, authors in [22] showed that the protocol in [21] is vulnerable to reader compromised attack. Moreover, they also showed that by executing this attack, the attacker could also perform reader impersonation, de-synchronization, and replay attacks. In [23], the authors proposed an ultra-lightweight RFID mutual authentication protocol for IoT in a secure manner. However, the authors in [24] illustrate that the protocol in [23] cannot satisfy all the security issues for RFID-based IoT systems and an attacker can compromise the reader and then execute the denial of service (DoS), reader and tag impersonation and de-synchronization attacks.

Considering the most recent researches related to RFID technology in the health-care environment, in 2014, authors in [25] proposed a secure ECC based RFID authentication protocol combined with ID-verifier transfer protocol. Authors [25] claim that their protocol has been proven invulnerable against various attacks. In the same year, the authors of [26] showed that the integrated protocol presented in [25] is vulnerable to the important compromise problem, where the adversary can reveal the secret information like the identity of a legitimate user by compromising the tag. Then, the authors in [26] proposed an improved RFID authentication protocol based on the previous approach presented in [25]. The authors in [26] claimed that their improved protocol is immune to the compromise problem and could withstand various attacks.

In [27], authors present an RFID authentication scheme based on the hash function and shared secret keys between the tag and the server. This scheme is called Telecare Medicine Information Systems (TMISs). Authors in [28] discovered that the protocol in [27] is vulnerable to multiple attacks like the destitute of mutual authentication attack, and the scheme is also not scalable. Authors in [28] provided their solution to solve the attacks mentioned above and tried to fix the problem. In the same year, authors in [29] showed that the solution presented in [28] also has weaknesses like the de-synchronization and reader traceability attacks. All these three schemes [27–29] are not suitable because they all use time-stamp, which is not supported by the Electronic Product Code (EPC) Class-1 Generation-2 standard (EPC technology is in use because of its low price). Same way authors in [30] proposed an efficient mutual authentication RFID protocol based on ECC and a one-way hash function. Then the authors in [31] proved that the protocol proposed in [30] is vulnerable to tag information privacy and forward–backward tracking problem, and they proposed an improved RFID authentication protocol based on the same approach as in [31]. Authors in [32] also found that the protocol presented in [30] is prone to impersonation attack, and with this finding, they proposed an improved protocol to avoid its drawbacks.

In 2015, authors in [33] proposed a lightweight ECC-based authentication protocol with an ID-verifier transfer for RFID systems. This protocol is mainly proposed for the mobile-health applications, and the authors claimed that their proposed protocol is efficient and secure. Later, in [34] authors showed that the protocol proposed in [33] is vulnerable to active tracking attack, which can be applied using bilinear pairing properties. After this, researches have proposed some schemes for health-care applications based on ECC [35–37], but due to the ponderous calculations on resource-constrained tags, the need of hash-based

authentication schemes due to their lightweight computation cost is defined [38].

In 2017, authors in [39] proposed a secure mobile RFID authentication protocol based on elliptic curve signature. The proposed protocol provides the safety of the patients' medical records and medical staff. This protocol also includes message recovery. Authors in [39] show that their proposed protocol solved the security weaknesses of some previously proposed protocols. With this proposal, they provide mobility, privacy, scalability, and it can be adapted for a multi-server environment. A new hash-based lightweight RFID mutual authentication protocol has been proposed in [38]. The protocol is proposed for health-care applications using cloud assistance. Authors in [38] proved that their protocol provides anonymity for both the tag and the reader and provides the forward–backward untraceability.

Recently in 2018, authors in [3] proposed a lightweight mutual authentication protocol. They claimed that their protocol owned the security properties necessary for RFID systems and is suitable for universal RFID applications such as IoT. In this paper, we show that the proposed protocol in [3] has serious vulnerabilities such as secret disclosure, reader impersonation, and tag traceability attacks. Moreover, their scheme is not able to preserve the anonymity of the tag and the reader.

The authentication and key agreement protocol, which is proposed for MIoT systems should meet a set of security and privacy requirements. Below, we list these requirements that we see in our work but the previous work presented in this section do not support all of them.

- *Mutual authentication*: In MIoT systems, each legal entity should prove its legitimacy before agreeing to the session key and transferring important data.
- *Entity untraceability*: An attacker or unauthorized entity should not be able to trace the target entity.
- *Message integrity*: Each entity has to sure of the integrity and freshness of the received messages from other entities.
- *Robustness against replay attack*: An adversary should not be able to relay the eavesdropped messages of the protocol to achieve malicious purposes such as entity impersonation.
- *Immunity against secret disclosure attack*: Messages of the protocol must be calculated in such a way that an adversary should not be able to obtain any vital information such as secret key form them.
- *Entity privacy preserving*: Privacy of each entity should not be faced with the leakage. So, running the protocol should not provide the opportunity for the adversary to obtain any information related to the entities.

As mentioned above, the predecessor schemes are not able to preserve all security and privacy concerns. So, in this paper, we aim to propose an authentication and key agreement protocol, which is not only secure but also efficient enough to be employed in MIoT systems.

3. Materials and methods

In [3], the authors proposed $cro(x, y)$ rotation function and employed this function as a building block in their scheme. In this section, first we describe the previously mentioned rotation function $cro(x, y)$. We show that this function has the complexity equal to XOR function and by knowing values of x (or y) and $cro(x, y)$, one can easily compute y (or x). Thus, to overcome this drawback, we introduce a more secure keyed rotation function called $MRot_{(K)}(x, y)$ to be used as a building block in our proposed protocol. Finally, we list notations used in this paper with the help of Table 2.

3.1. Rotation functions

In this subsection we briefly describe the *cross* function ($cro(x, y)$) proposed by the authors of LRMI [3] and our proposed modular rotate function ($MRot_{(K)}(x, y)$).

Definition 1. In $cro(x, y)$ function suppose that x and y are two N-bit strings, and $\sim x$ denotes the *not* operation on bit string x. Then the *cross* operation is defined as below:

- The odd bits of string $\sim x \parallel y$ are *XORed* by the even bits of string $\sim y \parallel x$, and the result is regarded as the odd bits of the final result.
- The even bits of string $\sim x \parallel y$ are *XORed* by the odd bits of string $\sim y \parallel x$, and the result is regarded as the even bits of the final result denoted by $cro(x, y)$.

Observation 1: The $cro(x, y)$ function can be implemented by two bit permutations over x and y, and two N-bit XORs. So, by knowing values of x (or y) and $cro(x, y)$, one can easily compute y (or x).

Proof: Let $x = (x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1)$ and $y = (y_8, y_7, y_6, y_5, y_4, y_3, y_2, y_1)$ be 8-bit strings. The functionality of $cro(x, y)$, which has been elaborated in Fig. 2 and Table 1, shows that $cro(x, y)$ can be written as $cro(x, y) = (\pi_1(x) \oplus \pi_2(y), \pi_2(x) \oplus \pi_1(y))$, where $\pi_1(x)$ and $\pi_2(x)$ are the following permutations.

$$\pi_1(x_8, x_7, \ldots, x_2, x_1) = (x_1, x_2, \ldots, x_7, x_8) \tag{1}$$

$$\pi_2(x_8, x_7, \ldots, x_2, x_1) = (x_2, x_1, x_4, x_3, x_6, x_5, x_8, x_7) \tag{2}$$

It is easy to see that for n-bit strings x and y, the relation $cro(x, y) = (\pi_1(x) \oplus \pi_2(y), \pi_2(x) \oplus \pi_1(y))$ is hold, where the general form of π_1 and π_2 are as relations (3) and (4).

$$\pi_1(x_n, x_{n-1}, \ldots, x_2, x_1) = (x_1, x_2, \ldots, x_{n-1}, x_n) \tag{3}$$

$$\pi_2(x_n, x_{n-1}, \ldots, x_2, x_1) = (x_2, x_1, \ldots, x_n, x_{n-1}) \tag{4}$$

Definition 2. Suppose that x and y are two n-bit strings and the key $K = k_2 \parallel k_1$ is a $2n$-bit string. Then, the modular rotate function $MRot_{(K)}(x, y)$ is defined as below.

- The odd bits of x are concatenated with even bits of y, and the result is *XORed* by the sub-key k_2. The circular shift on this string by $(y \oplus k_1)$ *mod* n bits to the left is regarded as the odd bits of the final result.
- The even bits of x are concatenated with odd bits of y, and the result is *XORed* by the sub-key k_1. The circular shift on this string by $(y \oplus k_2)$ *mod* n bits to the left is regarded as the even bits of the final result (as shown in Fig. 3).

Observation 2: As illustrated in Fig. 4, given Z, the entity who knows the values of y and K can quickly obtain the value of x.

Property 1: For any given $Z \in \{0, 1\}^{2n}$ there exist, on average, 2^{2n} tuples (x, y, k_1, k_2) that $MRot_{(K)}(x, y) = Z$.

Proof: Given $Z \in \{0, 1\}^{2n}$, Algorithm 1 exhaustively searches the whole space of tuples (j, i, p, q) such that $Z = MRot_{(K)}(x, y)$, where $j = y \oplus k_2 \ mod\ n$, $i = y \oplus k_1 \ mod\ n$, $p = k_2$ and $q = k_1$.

Let S be the space of these tuples. By considering the four nested *for* loops in Algorithm 1, one can easily see that $|S| = n^2 \times 2^{2n}$. On the other hand, the sieving probability defined by the *if* loop in Algorithm 1 is $\frac{1}{n^2}$. It means that

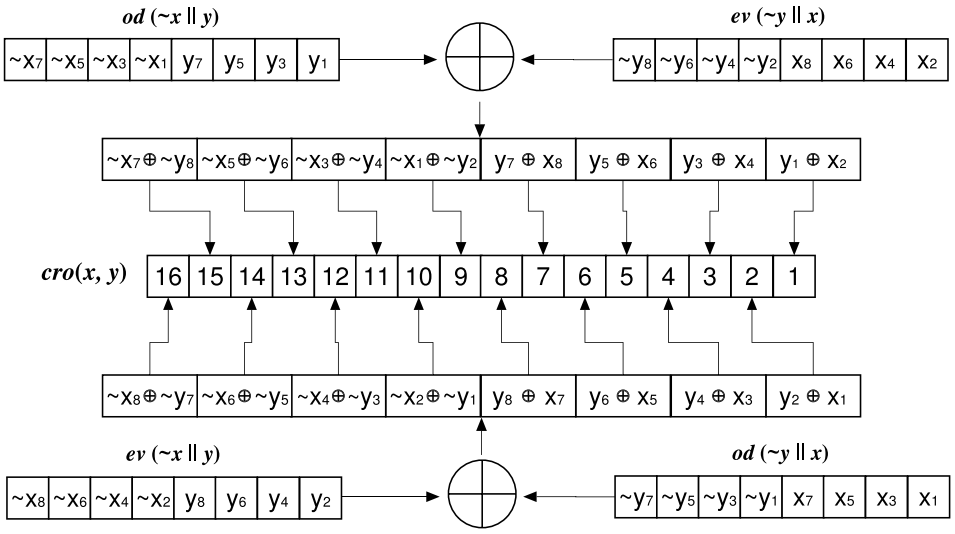

Fig. 2. The $cro(x, y)$ function.

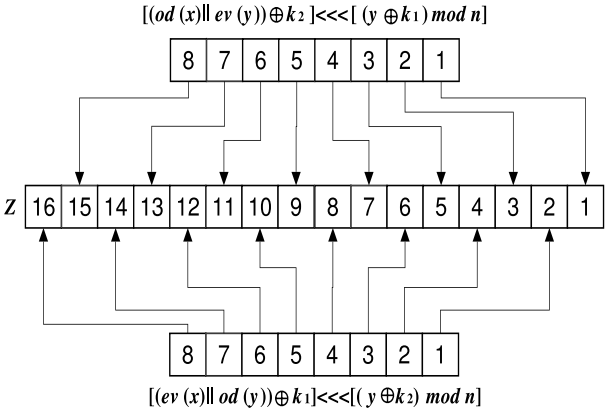

Fig. 3. The $Z = MRot_{(K)}(x, y)$ function.

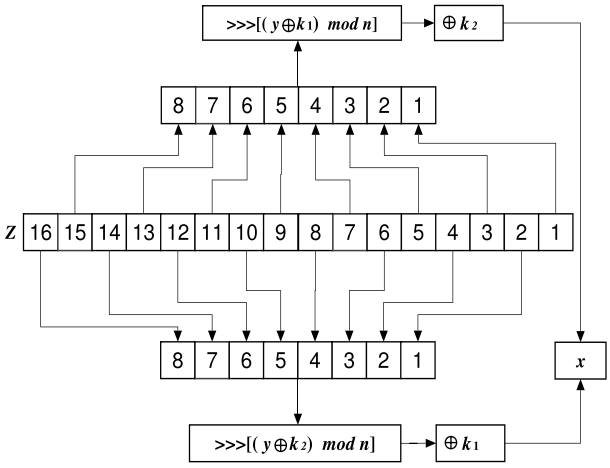

Fig. 4. Extracting x from the $Z = MRot_{(K)}(x, y)$ function.

the output size of Algorithm 1, say the number of tuples (x, y, k_1, k_2), is $|S| \times \frac{1}{n^2} = 2^{2n}$.

Corollary 1: For any given x and $y \in \{0, 1\}^n$, and $Z \in \{0, 1\}^{2n}$, there exists, on average, one value K such that $MRot_{(K)}(x, y) = Z$. In other words, for any output of the $MRot$ function, every pair (x, y) can generate this output. So, the adversary is not able to distinguish between these pairs.

4. LRMI authentication protocol

Recently, authors in [3] proposed a lightweight mutual authentication protocol for RFID systems, and they claimed that their protocol could be used for the MIoT applications. When an RFID system is applied in the IoT networks, a significant challenge that must be taken into account by the protocol designer is the potentially insecure channel between the server and the reader. In LRMI [3] protocol, the three components of the protocol pre-share the tuple $(K_i, cro(\cdot), Rot(\cdot), PRNG(\cdot))$. In their scheme, the Index Data Table includes index value and index content, which are unique (as shown in Table 3). In every session, the value of the key is updated, so the index value is fresh for each session. Moreover, after every successful session, the status of *Mark* changes to "10" from "00".

The LRMI [3] protocol shown in Fig. 5 runs the following steps.

Table 1
The 16 bits of $cro(x, y)$ function in terms of 8 bits of x and 8 bits of y.

cro output bits	*cro* output bits in terms of x and y
$cro(x, y)_1$	$\sim x_8 \oplus \sim y_7 = x_8 \oplus y_7$
$cro(x, y)_2$	$\sim x_7 \oplus \sim y_8 = x_7 \oplus y_8$
$cro(x, y)_3$	$\sim x_6 \oplus \sim y_5 = x_6 \oplus y_5$
$cro(x, y)_4$	$\sim x_5 \oplus \sim y_6 = x_5 \oplus y_6$
$cro(x, y)_5$	$\sim x_4 \oplus \sim y_3 = x_4 \oplus y_3$
$cro(x, y)_6$	$\sim x_3 \oplus \sim y_4 = x_3 \oplus y_4$
$cro(x, y)_7$	$\sim x_2 \oplus \sim y_1 = x_2 \oplus y_1$
$cro(x, y)_8$	$\sim x_1 \oplus \sim y_2 = x_1 \oplus y_2$
$cro(x, y)_9$	$x_7 \oplus y_8$
$cro(x, y)_{10}$	$x_8 \oplus y_7$
$cro(x, y)_{11}$	$x_5 \oplus y_6$
$cro(x, y)_{12}$	$x_6 \oplus y_5$
$cro(x, y)_{13}$	$x_3 \oplus y_4$
$cro(x, y)_{14}$	$x_4 \oplus y_3$
$cro(x, y)_{15}$	$x_1 \oplus y_2$
$cro(x, y)_{16}$	$x_2 \oplus y_1$

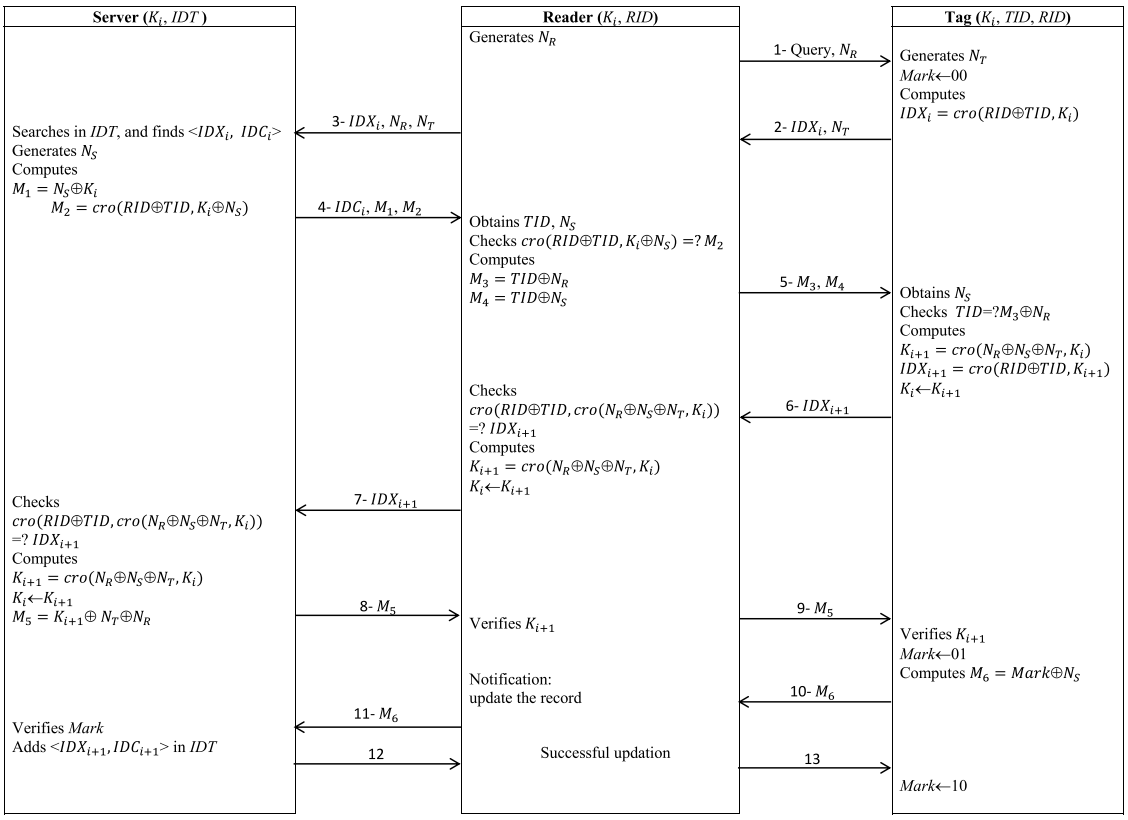

Fig. 5. LRMI [3] authentication protocol.

Algorithm 1: Brute force attack against the $MRot_{(K)}(x, y)$ function.

Input: $Z = MRot_{(K)}(x, y)$
Output: $x, y, K = k_2 \| k_1$
for $j = 0 : n - 1$
  $d_1 \leftarrow od(Z) \ggg j$
  for $i = 0 : n - 1$
    $d_2 \leftarrow ev(Z) \ggg i$
    for $k_2 = 0 : 2^n - 1$
      $d_3 \leftarrow d_2 \oplus k_2$
      for $k_1 = 0 : 2^n - 1$
        $d_4 \leftarrow d_1 \oplus k_1$
        if [(extracted y from $(d_3, d_4)) \oplus k_1$] $mod\ n = i$ &
          [(extracted y from $(d_3, d_4)) \oplus k_2$] $mod\ n = j$ then
            Return (k_1, k_2, x, y)
        end if
      end
    end
  end
end

1. The reader starts the protocol by sending a random number N_R to the tag.
2. Once the tag received this message, generates a random number N_T and sets $Mark = 00$. It then computes $IDX_i = cro(RID \oplus TID, K_i)$ and transmits $IDX_i \| N_T$ to the reader.
3. After receiving the message, the reader obtains N_T and forwards $IDX_i \| N_R \| N_T$ to the server.
4. Once the server received the message, it obtains N_R and N_T and then employs $IDX_i = cro(RID \oplus TID, K_i)$ to find the corresponding index content IDC_i in the IDT. If it can

Table 2
Notations.

Notation	Description
TID	Private ID of the tag
RID	Private ID of the reader
N_R, N_T, N_S	Random numbers generated by the reader, tag and server respectively
K_i	The *i-th* session key
k_2, k_1	The sub-keys in which $K_i = k_2 \| k_1$
$ev(Z)$	The even bits of the string Z
$od(Z)$	The odd bits of the string Z
IDX_i	The *i-th* session index value
IDC_i	The *i-th* session index content
$PRNG(\cdot)$	The Pseudo Random Number Generator function
$\pi_1(\cdot), \pi_2(\cdot)$	Permutation functions
$cro(x, y)$	The *cross* function
$Rot(x, y)$	The function of $W(y)$-bit circular left rotation on x, where $W(y)$ is the hamming weight of a string y
$MRot_{(K)}(x, y)$	The modular rotate function
$\oplus$	Exclusive OR operation
$Mark$	The status of the last session
$\|$	Concatenation operation
$a \lll x$	Circular shift of a by x bits to the left
$a \ggg x$	Circular shift of a by x bits to the right

Table 3
Index data table LRMI scheme.

Index value (IDX_i)	Index content (IDC_i)
$cro(RID \oplus TID, K_1)$	$Rot(K_1 \oplus TID, K_1 \oplus RID)$
$cro(RID \oplus TID, K_2)$	$Rot(K_2 \oplus TID, K_2 \oplus RID)$
…	…
$cro(RID \oplus TID, K_i)$	$Rot(K_i \oplus TID, K_i \oplus RID)$
$cro(RID \oplus TID, K_{i+1})$	$Rot(K_{i+1} \oplus TID, K_{i+1} \oplus RID)$

find a match, it indicates that the last session has been done correctly and the current session is executable. Then the server generates a random number N_S and computes $M_1 = N_S \oplus K_i$ and $cro(RID \oplus TID, K_i \oplus N_S)$. It then sends $IDC_i \parallel M_1 \parallel M_2$ to the reader. Otherwise, the authentication fails, and the protocol will be terminated.

5. Once the reader received the tuple $(IDC_i \parallel M_1 \parallel M_2)$, according to the hamming weight $W(K_i \oplus TID)$ of the rotation function and $K_i \oplus K_i \oplus TID$ it obtains TID. It then obtains N_S and verifies the value of $cro(RID \oplus TID, K_i \oplus N_S)$ by comparing with the received value M_2. If so, it computes $M_3 = TID \oplus N_R$ and $M_4 = TID \oplus N_S$ and sends them to the tag.
6. After receiving these messages, the tag obtains N_S and if $TID = M_3 \oplus N_R$ holds, it authenticates the server and the reader. Then the tag updates K_i as $K_{i+1} = cro(N_R \oplus N_S \oplus N_T, K_i)$ and sends it to the reader involved in the message $IDX_{i+1} = cro(RID \oplus TID, K_{i+1})$. Otherwise the authentication fails.
7. Upon receiving the message IDX_{i+1}, if $cro(RID \oplus TID, cro(N_R \oplus N_S \oplus N_T, K_i)) = IDX_{i+1}$ holds, the reader updates K_i by the same equation $K_{i+1} = cro(N_R \oplus N_S \oplus N_T, K_i)$ and sends it to the server by the message $IDX_{i+1} = cro(RID \oplus TID, K_{i+1})$. Otherwise the protocol will be terminated.
8. Once the server received this message, it does the same checking operation, and if it holds, the server updates K_i as $K_{i+1} = cro(N_R \oplus N_S \oplus N_T, K_i)$. It then computes the message $M_5 = K_{i+1} \oplus N_T \oplus N_R$ and sends it to the reader. Otherwise, the connection fails.
9. Upon receiving the message M_5, if $K_{i+1} = M_5 \oplus N_T \oplus N_R$ holds, the reader verifies K_{i+1} and sends the message M_5 to the tag for the same verification process. Otherwise, the protocol will be terminated.
10. Once the tag received message M_5, it does the same checking operation. If the tag accepts the validity of K_{i+1}, it sets $Mark = 01$, indicating the synchronization of K_i is completed. Then the tag computes $M_6 = Mark \oplus N_S$ and sends it to the server through the reader. Note that in the original work [3], $Mark$ has been defined as a 2-bit string while parameters like N_S are typically larger strings, so their XOR does not make any sense. However, without loss of generality, we assume $Mark$ is some trivial extension of this 2-bit string.
11. After receiving the message M_6, the server obtains the value of $Mark$ and if it is equal to 01, it concludes that the synchronization of K_i is completed. Then the server adds a new record $IDX_{i+1} = cro(RID \oplus TID, K_{i+1})$, $IDC_{i+1} = Rot(K_{i+1} \oplus TID, K_{i+1} \oplus RID)$ to IDT, after which the notification that the record completes the update is sent to the tag through the reader.
12. Now, the tag sets $Mark = 10$, indicating that the authentication protocol is completed.

5. Security analysis of the LRMI protocol

Although this paper is related to the analysis of RFID mutual authentication in the MIoT networks, it is necessary to discuss all kinds of adversary models briefly. An adversary who plays a primary role in cryptanalysis of security protocols can be categorized into the following two groups [40]: (i) passive adversaries, and (ii) active adversaries.

Passive adversary has only access to the transactions between all the entities of the protocol who transfer their messages via insecure channel. As a result, this kind of attacker can eavesdrop, intercept, and replay messages with no ability to change or generate them. While, on the other hand, an active adversary can impersonate one of the protocol parties (i.e, tag, reader, or server) by using suitable devices (e.g., a rough reader for reader impersonation in the proximity of a legitimate tag) and then communicate with the other protocol's party, and in this line, s/he can modify the transferred messages [40]. Putting the impersonating device in the proximity of readers or tags is the main complexity of such attacks [41]. Moreover, modifying or blocking transferred messages can be conducted by using man-in-the-middle devices [42].

Below, we present several attacks against LRMI [3] protocol. Mainly, we show that this protocol is vulnerable to secret disclosure, reader impersonation, and tag traceability attacks. Moreover, we show that despite the designers claim, the protocol fails to protect the privacy of the tag and the reader.

5.1. Secret disclosure attack

In this attack, the goal is to recover the secret shared between the entities of the authentication protocol. In LRMI protocol, the passive adversary eavesdrops the messages of the protocol and tries to obtain the updated shared secret used in the next run of the protocol. S/he starts the attack by eavesdropping the messages of steps 1, 2 and 9, which are respectively N_R, N_T, and $K_{i+1} \oplus N_T \oplus N_R$. It then executes the attack by obtaining the new session key K_{i+1} from the equation $K_{i+1} = (K_{i+1} \oplus N_T \oplus N_R) \oplus N_R \oplus N_T$.

5.2. Attack on the privacy of the tag and the reader

In the LRMI protocol, the authors claim that their scheme can successfully preserve the anonymity of the tag and the reader. In this sub-section, we show how a passive attacker can obtain both tag and reader identities TID and RID, respectively.

- Attack on the tag anonymity: The attacker can eavesdrop the messages of steps 1 and 5 which are respectively N_R and $TID \oplus N_R$, and it jeopardizes the anonymity of the target tag by obtaining the identification TID of the tag from the equation $TID = (TID \oplus N_R) \oplus N_R$.
- Attack on the reader anonymity: In this attack, the passive adversary wants to obtain the identification of the reader RID. This attack is based on Observation 1, which is mentioned in Section 3. The scenario of the attack is that the adversary eavesdrops the message of Step 6 which is $cro(RID \oplus TID, K_{i+1})$, then uses K_{i+1}, obtained in the above mentioned secret disclosure attack, and the $cro(x, y)$ property mentioned in Observation 1 to obtain $RID \oplus TID$. Finally, it uses TID, obtained in the attack as mentioned above on the tag anonymity to compute RID. Hence, the adversary can jeopardize the anonymity of the reader.

5.3. Reader impersonation attack

The concept of this attack is that the active adversary tries to run a new successful session with the target tag as a legitimate reader. Assuming the situation that the attacker has already made previous attacks and obtained the tag's identification TID and the tag's current key K_i. The reader impersonation attack against LRMI [3] protocol is described as follows.

1. The active adversary starts the protocol by sending a random number N_{A1} to the tag;
2. The tag generates a random number N_T and sets $Mark = 00$. It then transmits $cro(RID \oplus TID, K_i) \parallel N_T$ to the adversary.
3. Once the adversary received the message, it generates another random number N_{A2} and computes $TID \oplus N_{A1}$ and $TID \oplus N_{A2}$ and sends them to the tag.

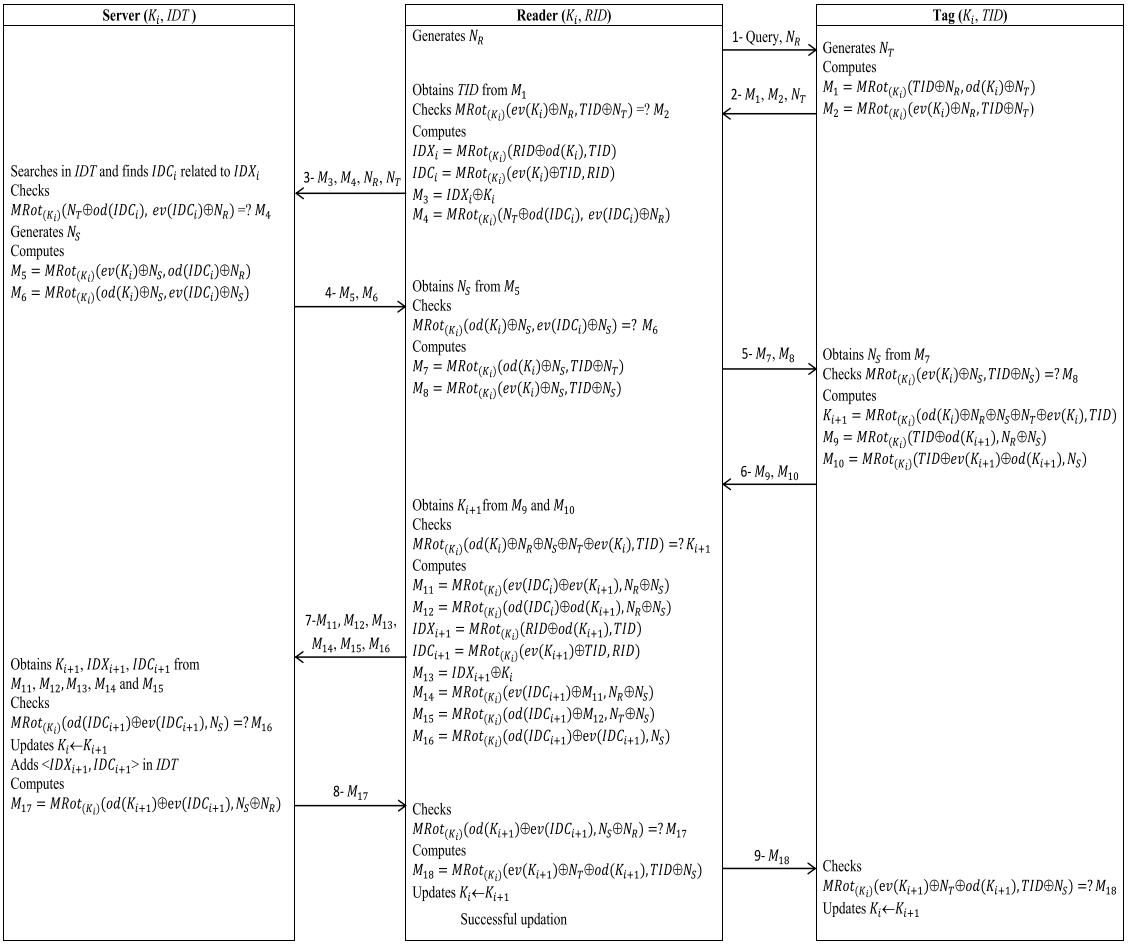

Fig. 6. SecLAP: Our improved authentication protocol.

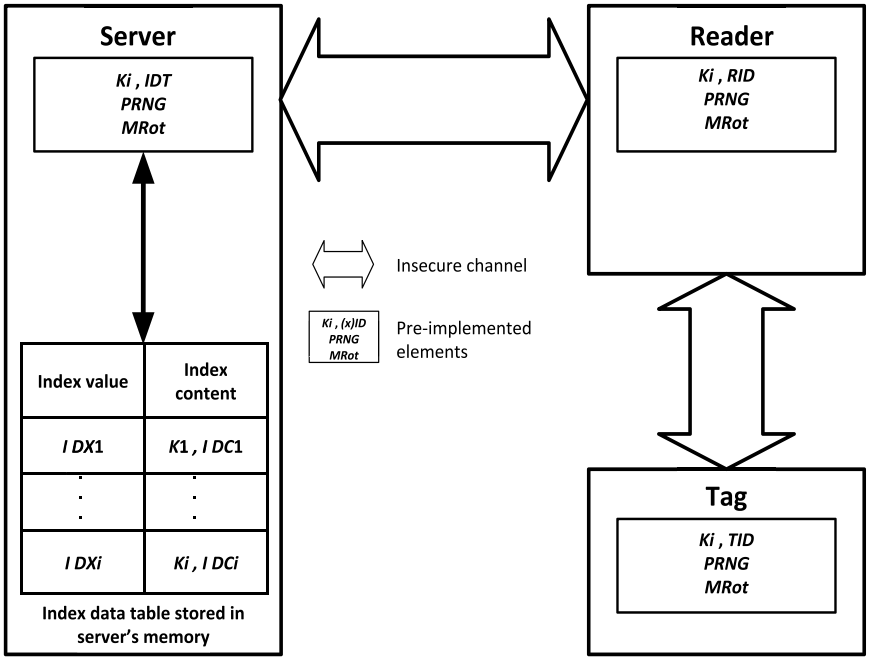

Fig. 7. SecLAP: Block diagram.

Table 4
Index data table of the SecLAP protocol.

Index value (IDX_i)	Index content (K_i, IDC_i)
$MRot_{(K_1)}(RID \oplus od(K_1), TID)$	K_1, $MRot_{(K_1)}(ev(K_1) \oplus TID, RID)$
$MRot_{(K_2)}(RID \oplus od(K_2), TID)$	K_2, $MRot_{(K_2)}(ev(K_2) \oplus TID, RID)$
...	...
$MRot_{(K_i)}(RID \oplus od(K_i), TID)$	K_i, $MRot_{(K_i)}(ev(K_i) \oplus TID, RID)$
$MRot_{(K_{i+1})}(RID \oplus od(K_{i+1}), TID)$	K_{i+1}, $MRot_{(K_{i+1})}(ev(K_{i+1}) \oplus TID, RID)$

4. After receiving this message, the tag obtains N_{A2} and verifies $TID = TID \oplus N_{A1} \oplus N_{A1}$ and authenticates the adversary. Then it updates K_i as $K_{i+1} = cro(N_{A1} \oplus N_{A2} \oplus N_T, K_i)$ and sends it to the adversary in the blind form of $cro(RID \oplus TID, K_{i+1})$.
5. The adversary uses N_{A1}, N_{A2}, N_T and K_i, and computes $K_{i+1} = cro(N_{A1} \oplus N_{A2} \oplus N_T, K_i)$. Then, it sends the message $K_{i+1} \oplus N_T \oplus N_{A1}$ to the tag for the verification process.
6. Upon receiving the message $K_{i+1} \oplus N_T \oplus N_{A1}$, the tag checks if $K_{i+1} = (K_{i+1} \oplus N_T \oplus N_{A1}) \oplus N_T \oplus N_{A1}$ holds. So, the tag verifies K_{i+1} and sets $Mark = 01$, indicating the synchronization of K is completed. Then the tag computes $Mark \oplus N_{A2}$ and sends it to the adversary.
7. After receiving the message, the adversary informs the tag that the update is successful.
8. Now, the tag sets $Mark = 10$, indicating the authentication protocol is completed.

5.4. Tag traceability attack

To trace a target tag it is enough to link two sessions of the protocol in which the same tag is involved. Below, we use Ouafi and Phan privacy model [43] to describe a traceability attack on LRMI [3] protocol. In LRMI protocol, the passive adversary sends an *Execute query* $(Reader, Tag, n)$ and intercepts message of Step 6 $IDX_{n+1,0}^{Tag_0} = cro(RID_{n,0}^{Tag_0} \oplus TID_{n,0}^{Tag_0}, K_{n+1,0}^{Tag_0})$. Now, the adversary selects two tags Tag_0 and Tag_1 and sends a *Test query* $(Tag_1, Tag_0, n+1)$ and receives $IDX_b^{Tag_b} \in \{IDX_0^{Tag_0}, IDX_1^{Tag_1}\}$ corresponding to $Tag_b \in \{Tag_0, Tag_1\}$ for a random bit $b \in \{0, 1\}$. Then, the adversary sends an *Execute query* $(Reader, Tag, n+1)$ and eavesdrops the message of Step 2 of the current session which is $IDX_{n+1,b}^{Tag_b} = cro(RID_{n+1,b}^{Tag_b} \oplus TID_{n+1,b}^{Tag_b}, K_{n+1,b}^{Tag_b})$.

At this point, if the adversary finds $IDX_{n+1}^{Tag_0} = IDX_{n+1}^{Tag_b}$, s/he can judge that $b = 0$. As K_{n+1}, RID_n and TID_n in message Step 6 of the *n-th* session and K_{n+1}, RID_{n+1} and TID_{n+1} in message Step 2 of the $(n+1)$-*th* session are the same, it means that $IDX_n^{Tag_b} = IDX_{n+1}^{Tag_0}$, then $Tag_b = Tag_0$ and $b = 0$. Thus, the attacker can use this link to distinguish and track tags.

6. SecLAP authentication protocol

In this section, we proposed an enhanced authentication protocol called SecLAP to overcome the security concerns of the LRMI [3] protocol. In SecLAP protocol, the flows between tag and reader are reduced significantly, and the server does not store the identity of the tag and the reader (*RID* and *TID*). Moreover, in our SecLAP protocol, the freshness of all messages are preserved by employing the random numbers generated by legal entities. Also, applying a secure and lightweight $MRot(\cdot)$ function is the salient and essential point of the SecLAP protocol.

Fig. 7 provides the block diagram which describe the structure of our proposed scheme. As mentioned in this diagram, the three components of the protocol (server, reader and tag) pre-share the tuple $(K_i = (k_2 \parallel k_1)_i, Rot(\cdot), PRNG(\cdot))$. In addition, the Index Data Table (IDT) including index value (IDX) and index content (IDC) which are unique (as shown in Table 4) is stored in server's memory. In the proposed IDT, the index content involves K_i and $IDC_i = MRot_{(K_i)}(ev(K_i) \oplus TID, RID)$. In every session, the value of key, index value and index content are updated. Besides, at the moment IDT keeps two tuples of index value and index content (e.g., (IDX_i, K_i, IDC_i) and $(IDX_{i+1}, K_{i+1}, IDC_{i+1})$ after running the $i+1$-*th* session).

Our proposed SecLAP protocol, as shown in Fig. 6, runs the following steps.

1. The reader starts the protocol by sending a random number N_R along with the *Query* to the tag.
2. Once the tag received this message, generates a random number N_T and computes $M_1 = MRot_{(K_i)}(TID \oplus N_R, od(K_i) \oplus N_T)$ and $M_2 = MRot_{(K_i)}(ev(K_i) \oplus TID \oplus N_R, TID \oplus N_T)$. It then transmits $M_1 \parallel M_2 \parallel N_T$ to the reader.
3. After receiving the message, the reader uses the stored K_i and obtains TID from the message M_1. It also calculates $MRot_{(K_i)}(ev(K_i) \oplus TID \oplus N_R, TID \oplus N_T)$ and checks if M_2 is valid. Then it computes $IDX_i = MRot_{(K_i)}(RID \oplus od(K_i), TID)$, $IDC_i = MRot_{(K_i)}(ev(K_i) \oplus TID, RID)$, $M_3 = IDX_i \oplus K_i$ and $M_4 = MRot_{(K_i)}(N_T \oplus od(IDC_i), ev(IDC_i) \oplus N_R)$ and forwards $M_3 \parallel M_4 \parallel N_R \parallel N_T$ to the server. Otherwise the authentication fails and the protocol is terminated.
4. Once the server receives the message, it obtains IDX_i and searches in IDT to find the corresponding K_i and IDC_i in the IDT. Then the server checks if M_4 is valid. If so, it generates a random number N_S and computes $M_5 = MRot_{(K_i)}(ev(K_i) \oplus N_S, od(IDC_i) \oplus N_R)$ and $M_6 = MRot_{(K_i)}(od(K_i) \oplus N_S, ev(IDC_i) \oplus N_S)$. It then sends $M_5 \parallel M_6$ to the reader. Otherwise the protocol is terminated.
5. Once the reader received the tuple $(M_5 \parallel M_6)$, it obtains N_S from M_5 and verifies the value of computed $MRot_{(K_i)}(od(K_i) \oplus N_S, ev(IDC_i) \oplus N_S)$ by comparing with the received value M_6. If so, it computes $M_7 = MRot_{(K_i)}(od(K_i) \oplus N_S, TID \oplus N_T)$ and $M_8 = MRot_{(K_i)}(ev(K_i) \oplus N_S, TID \oplus N_S)$ and sends M_7 along with M_8 to the tag.
6. After receiving these messages, the tag obtains N_S from M_7 and if the validity of M_8 holds, it authenticates the server and the reader. Then the tag computes $K_{i+1} = MRot_{(K_i)}(od(K_i) \oplus N_R \oplus N_S \oplus N_T \oplus ev(K_i), TID)$ and sends it to the reader involved in the messages $M_9 = MRot_{(K_i)}(TID \oplus od(K_{i+1}), N_R \oplus N_S)$ and $M_{10} = MRot_{(K_i)}(TID \oplus ev(K_{i+1}) \oplus od(K_{i+1}), N_S)$. Otherwise the authentication fails.
7. Upon receiving the message $M_9 \parallel M_{10}$, the reader obtains K_{i+1} and if $MRot_{(K_i)}(od(K_i) \oplus N_R \oplus N_S \oplus N_T \oplus ev(K_i), TID) = K_{i+1}$ holds, the reader computes $M_{11} = MRot_{(K_i)}(ev(IDC_i) \oplus ev(K_{i+1}), N_R \oplus N_S)$, $M_{12} = MRot_{(K_i)}(od(IDC_i) \oplus ed(K_{i+1}), N_R \oplus N_S)$,$IDX_{i+1} = MRot_{(K_i)}(RID \oplus od(K_{i+1}), TID)$, $IDC_{i+1} = MRot_{(K_i)}(ev(K_{i+1}) \oplus TID, RID)$,$M_{13} = IDX_{i+1} \oplus K_i$,$M_{14} = MRot_{(K_i)}(ev(IDC_{i+1}) \oplus M_{11}, N_R \oplus N_S)$, $M_{15} = MRot_{(K_i)}(od(IDC_{i+1}) \oplus M_{12}, N_T \oplus N_S)$ and $M_{16} = MRot_{(K_i)}(od(IDC_{i+1}) \oplus ev(IDC_{i+1}), N_S)$ and sends the tuple $(M_{11} \parallel M_{12} \parallel M_{13} \parallel M_{14} \parallel M_{15} \parallel M_{16})$ to the server. Otherwise it terminates the protocol.
8. Once the server received this message, it obtains K_{i+1}, IDX_{i+1} and IDC_{i+1} from M_{11}, M_{12}, M_{13}, M_{14} and M_{15} respectively. If the validity of M_{16} holds, the server updates K_i as K_{i+1}. It then adds the tuple $(IDX_{i+1}, K_{i+1}, IDC_{i+1})$ to the IDT and computes the message $M_{17} = MRot_{(K_i)}(od(K_{i+1}) \oplus ev(IDC_{i+1}), N_S \oplus N_R)$ and sends it to the reader. Otherwise the connection fails and the protocol is terminated.
9. Upon receiving the message M_{17}, if $MRot_{(K_i)}(od(K_{i+1}) \oplus ev(IDC_{i+1}), N_S \oplus N_R) = M_{17}$ holds, the reader updates K_i as K_{i+1} and computes $M_{18} = MRot_{(K_i)}(ev(K_{i+1}) \oplus N_T \oplus od(K_{i+1}), TID \oplus N_S)$. It then, sends the message M_{18} to the tag for the verification process. Otherwise the protocol is terminated.

10. Once the tag received the message, it checks the validity of M_{18}. If the tag validates this value, it updates K_i as $K_{i+1} = MRot_{(K_i)}(od(K_i) \oplus N_R \oplus N_S \oplus N_T \oplus ev(K_i), TID)$ and indicates that the authentication protocol is completed.

7. Security analysis of the SecLAP protocol

In this section, the security analysis of the SecLAP protocol is evaluated, which has a security level higher than the LRMI [3] protocol.

7.1. Informal security analysis

In this section, we informally analyze the security of the SecLAP authentication protocol against known attacks as below.

1. *Resistance to de-synchronization attack:* In the SecLAP, the server stores the previous record of *IDT* included the tuple (IDX_i, IDC_i) in the $i+1$-*th* session. So, if an attacker tries to block the message M_{18} in order to perform de-synchronization attack, when the tag starts the protocol with K_i and IDX_i, because the server stores these values in the $i+1$-*th* session, it accepts the tags message and omits the values which depend on K_{i+1} and IDX_{i+1}.
2. *Resistance to replay attack:* The tag, the reader and the server use the freshly generated random numbers N_R, N_T, N_S, and a random sequence in their messages. Therefore, an adversary cannot be successful in his/her attack by replaying messages from previous sessions. Hence, the improved protocol provides resistance to the replay attack.
3. *Resistance to reader impersonation attack:* By using a random nonce N_T generated by the tag in the messages M_7 and M_{18} computed by the reader and also the verification process in the tag side, the adversary cannot use the eavesdropped previous messages M_7 and M_{18} and resend them in the other new session. Also, the adversary cannot be able to compute these message without knowing the secret key K_i and the tag's identification *TID*. So, the adversary cannot send an acceptable response to the tag. Therefore, the proposed SecLAP protocol is robust against reader impersonation attack.
4. *Resistance to tag impersonation attack:* In the improved protocol, the tag uses the new random number N_R generated by the reader in all the messages M_1, M_2 and M_9. So, due to the verification processes in the reader side, the adversary cannot use previous eavesdropped messages M_1, M_2, and M_9. Moreover, the adversary cannot compute M_1, M_2 and M_9 without having any knowledge on K_i and N_S.
5. *Resistance to traceability attack:* In this attack, an adversary tries to predict/find the relation between the tag responses by sending the constant messages to the tag. So, if the responses are constant, they might be predictable by an adversary. Consequently, the adversary can trace the target tag successfully. In our proposed SecLAP protocol, all of the messages sent from the tag are involved by the fresh random numbers N_T generated by the tag. So, the adversary cannot find any similarity between all tags responses with the acceptable probability when the tag employs this random number to compute the messages.
6. *Forward and backward security:* In forward and backward security, an adversary should not be able to find any message that can help her/him to obtain the current and previously confidential information. In our proposed protocol SecLAP, all of the messages are computed by irreversible function $(MRot_{(K)}(\cdot))$ without knowing the value of the current key. Therefore, our proposed protocol SecLAP achieves forward and backward security.

Table 5
Security comparison.

Authentication protocol	RD	RR	RRI	RTI	RT	FBS	RS
Cheng et al. [17]	No	Yes	Yes	Yes	Yes	Yes	No
Benssalah et al. [19]	No	Yes	Yes	No	No	Yes	Yes
Zhu et al. [21]	No	No	No	Yes	Yes	Yes	Yes
Fan et al. [23]	No	Yes	No	No	Yes	Yes	Yes
LRMI [3]	Yes	Yes	No	Yes	No	Yes	No
SecLAP (Our solution)	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

RD: resistance against de-synchronization attack
RR: resistance against replay attack
RRI: resistance against reader impersonation
RTI: resistance against tag impersonation
RT: resistance against traceability attack
FBS: forward and backward security
RS: resistance against secret disclosure attack

Table 6
BAN-logic notations.

Notation	Description
$P \mid\equiv X$	P believes X
$P \triangleleft X$	P receives X
$P \mid\sim X$	P sends X
$P \Rightarrow X$	P has jurisdiction over X
$\sharp(X)$	X is fresh
$\{X\}_k$	X is encrypted by the secret k
$(X)_k$	X is hashed by the secret k
$P \overset{k}{\leftrightarrow} Q$	P and Q have a shared secret k
$\frac{P}{Q}$	If P then Q

7. *Privacy:* The RFID reader and RFID tag anonymity should be preserved to ensure privacy. It is essential to say that, as in our proposal the tag identity *TID* and the reader identity *RID* are protected by $(MRot_{(K)}(\cdot))$, due to which an adversary is not able to determine the identity of the tag and the reader. So, our proposed protocol SecLAP preserves the tag and the reader anonymity.

   In Table 5, the security comparison among our proposed protocol SecLAP and some other similar existing authentication protocols is presented. In this table, the symbol 'yes' denotes that the authentication protocol prevents an attack and the symbol 'No' represents that the authentication protocol does not resist the attack.

7.2. Formal security analysis

We use BAN-logic [44] to conduct a formal security analysis of our SecLAP authentication protocol. The notations used in BAN-logic proof are listed in Table 6. In this section, we denoted the server, the reader, and the tag by S, R, and T, respectively. The essential rules that we employ in our analysis are as below.

R1 (Shared key rule): $\frac{P \mid\equiv P \overset{k}{\longleftrightarrow} Q, P \triangleleft [X]_k}{P \mid\equiv Q \mid\sim X}$, it means that when P believes that he/she shared the key K with Q, and then it received the message $[X]_k$, as a result, it can believe that Q has sent X to him/her.

R2 (Belief rule). $\frac{P \mid\equiv Q \mid\sim (X,Y)}{P \mid\equiv Q \mid\sim X}$, it means that when P believes Q sent the message set (X, Y), then P believes X is also sent from Q.

The steps of our formal security analysis are as follows.

Step 1. Messages transmitted in the one session of the protocol: All messages that are involved in the proposed protocol are listed as below.

PM1: N_R, this random nonce is transferred from the reader to the tag in Step 1 as the first message of the protocol.

PM2: M_1, M_2, N_T, these values are transferred from the tag to the reader in Step 2.

PM3: M_3, M_4, N_R, N_T, the reader sends these four values to the server in Step3.

PM4: M_5, M_6, this message which is transferred in Step 4 is from the server to the reader.

PM5: M_7, M_8, this message is transferred from the reader to the tag in Step 5.

PM6: M_9, M_{10}, this message is the response of the tag to the reader in Step 6.

PM7: $M_{11}, M_{12}, M_{13}, M_{14}, M_{15}, M_{16}$, these six values are transferred from the reader to the server in Step 7.

PM8: M_{17}, this value is transferred from the server to the reader in Step 8.

PM9: M_{18}, the reader transfers this value, which is the last message of the protocol to the tag.

Step 2. Idealizing the messages of the protocol: In this step, the idealized form of the messages of the protocol are listed as IM1, …, IM10, which are based on the BAN-logic notations.

IM1 $(T \rightarrow R)$**:** $R \lhd \{TID\}_{K_i}$, this represents that the reader receives the value of the *TID* encoded by the key K_i.

IM2 $(R \rightarrow S)$**:** $S \lhd \{IDX_i\}_{K_i}$, this idealized message indicates that the server receives the value of the IDX_i encrypted using the key K_i.

IM3 $(R \rightarrow S)$**:** $S \lhd \{IDC_i\}_{K_i}$, this implies that the server also receives the value of the IDC_i encrypted by the key K_i.

IM4 $(S \rightarrow R)$**:** $R \lhd \{N_S\}_{K_i, IDC_i}$, this ideally represents that the reader receives the nonce N_S encrypted by both K_i and IDC_i.

IM5 $(R \rightarrow T)$**:** $T \lhd \{N_S\}_{K_i, TID}$, this idealized message indicates that the tag receives the nonce N_S encrypted by both K_i and *TID*.

IM6 $(T \rightarrow R)$**:** $R \lhd \{K_{i+1}\}_{K_i, TID, N_S}$, this depicts that the reader receives the value of the K_{i+1} encrypted by the values K_i, *TID* and N_S.

IM7 $(R \rightarrow S)$**:** $S \lhd \{K_{i+1}, IDC_{i+1}\}_{K_i, IDC_i}$, this depicts that the server receives the tuple $\{K_{i+1}, IDC_{i+1}\}$ encrypted by both K_i and IDC_i.

IM8 $(R \rightarrow S)$**:** $S \lhd \{IDX_{i+1}\}_{K_i}$, this idealized message shows that the server receives the nonce IDX_{i+1} encrypted by the key K_i.

IM9 $(S \rightarrow R)$**:** $R \lhd \{IDC_{i+1}, K_{i+1}\}_{K_i, N_S}$, this ideally represents that the reader receives the tuple $\{IDC_{i+1}, K_{i+1}\}$ encrypted by both K_i and N_S.

IM10 $(R \rightarrow T)$**:** $T \lhd \{K_{i+1}\}_{K_i, TID, N_S}$, this idealized message indicates that the tag receives the updated key K_{i+1} encrypted by the values K_i, *TID* and N_S.

Step 3. Explicit assumptions: The explicit assumptions on the proposed protocol are as below.

A1: $R \mid\equiv \sharp(N_R)$, it indicates that the reader believes that the random nonce N_R is fresh.

A2: $T \mid\equiv \sharp(N_T)$, it shows that the tag believes on the freshness of the random nonce N_T.

A3: $S \mid\equiv \sharp(N_S)$, this explicit assumption indicates that the server also believes that the random nonce N_S is fresh.

A4: $T \mid\equiv T \overset{K_i}{\longleftrightarrow} R$, it presents that the tag believes that it has the shared secret key K_i with the reader.

A5: $R \mid\equiv R \overset{K_i}{\longleftrightarrow} T$, it implies that the reader also believes that it has the shared secret key K_i with the tag.

A6: $R \mid\equiv R \overset{K_i}{\longleftrightarrow} S$, this explicit assumption presents that the reader believes that it has the shared secret key K_i with the server.

A7: $S \mid\equiv S \overset{K_i}{\longleftrightarrow} R$, it predicts that the server also believes that it has the shared secret key K_i with the reader.

A8: $R \Rightarrow IDC_i$, it represents that the reader believes that it has jurisdiction over IDC_i.

A9: $T \Rightarrow TID$, this explicit assumption indicates the tag believes that it has jurisdiction over *TID*.

Step 4. Security goals of the protocol: The security goals of the protocol (G1, …, G12) which are expected to be verified after analyzing the protocol by BAN-logic are listed as below.

G1: $R \mid\equiv T \mid\sim TID$, it implies that the reader must believe that the tag has sent the *TID*.

G2: $S \mid\equiv R \mid\sim IDX_i$, this goal shows that the server must believe that the reader has sent the IDX_i.

G3: $S \mid\equiv R \mid\sim IDC_i$, it ensures that the server believes that the reader has sent the IDC_i.

G4: $R \mid\equiv S \mid\sim N_S$, it states that the reader convinced that the server has sent the N_S.

G5: $T \mid\equiv R \mid\sim N_S$, this goal shows that the tag have no doubt that the reader has sent the N_S.

G6: $R \mid\equiv T \mid\sim K_{i+1}$, it ensures that the reader believes that the tag has sent the K_{i+1}.

G7: $S \mid\equiv R \mid\sim K_{i+1}$, it shows that the server approbated that the reader has sent the K_{i+1}.

G8: $S \mid\equiv R \mid\sim IDC_{i+1}$, it states that the server must believe that the reader has sent the IDC_{i+1}.

G9: $S \mid\equiv R \mid\sim IDX_{i+1}$, this goal shows that the server have no doubt that the reader has sent the IDX_{i+1}.

G10: $R \mid\equiv S \mid\sim K_{i+1}$, it states that the reader is convinced that the server has sent the K_{i+1}.

G11: $R \mid\equiv S \mid\sim IDC_{i+1}$, this goal shows that the reader must believe that the server has sent the IDC_{i+1}.

G12: $T \mid\equiv R \mid\sim K_{i+1}$, it states that the tag is convinced that the reader has sent the K_{i+1}.

Step 5. Deriving the security goals of the protocol: Finally, we apply logical rules of the BAN-logic to show the realization of the goals mentioned above. We use the idealized messages and the initial premises presented in the previous steps. The goals are achieved as below.

According to PM2, IM1, A1, A5, and R1:
Result1: $R \mid\equiv T \mid\sim TID$ (satisfy **G**1);
Given the PM3, IM2, A7, and R1:
Result2: $S \mid\equiv R \mid\sim IDX_i$ (satisfy **G**2);
Given the PM3, IM3, A7, and R1:
Result3: $R \mid\equiv S \mid\sim N_S$(satisfy **G**3);
In accordance with PM4, IM4, A1, A6, A8, and R1:
Result4: $R \mid\equiv S \mid\sim N_s$ (satisfy **G**4);
According to PM5, IM5, A2, A4, A9, and R1:
Result5: $T \mid\equiv R \mid\sim (N_S)$ (satisfy **G**5);
According to PM6, IM6, A1, A5, Result1, Result4, and R1:
Result6: $R \mid\equiv T \mid\sim K_{i+1}$ (satisfy **G**6);
Given the PM7, IM7, A3, A7, Result3, and R1:
Result7: $S \mid\equiv R \mid\sim (K_{i+1}, IDC_{i+1})$
Taking into account Result7, and R2:
Result8: $S \mid\equiv R \mid\sim K_{i+1}$ (satisfy **G**7);
Result9: $S \mid\equiv R \mid\sim IDC_{i+1}$ (satisfy **G**8);
According to PM7, IM8, A3, A7, Result3, and R1:
Result10: $S \mid\equiv R \mid\sim IDX_{i+1}$ (satisfy **G**9);
Given the PM8, IM9, A1, A6, Result4, and R1:
Result11: $R \mid\equiv S \mid\sim (K_{i+1}, IDC_{i+1})$
Taking into account Result11 and R2:
Result12: $R \mid\equiv S \mid\sim K_{i+1}$ (satisfy **G**10);
Result13: $R \mid\equiv S \mid\sim IDC_{i+1}$ (satisfy **G**13);
Given the PM9, IM10, A2, A4, A9, Result5 and R1:
Result14: $T \mid\equiv R \mid\sim (K_{i+1})$ (satisfy **G**12);

As proved in the above steps, it can quickly be concluded that the protocol can verify all preset goals. So, we can say that our proposed authentication protocol SecLAP is secure.

8. Design and simulation of the *MRot* function proposed for SecLAP

In SecLAP protocol, the $MRot_{(K)}(x, y)$ function used as a building block in all transferred messages. Thus, we use the register transfer level (RTL) design to implement this function and to show its efficiency. The implementation is conducted on a Xilinx Kintex-7 (XC7k480t) FPGA. We use VIVADO 2018.1 to carry

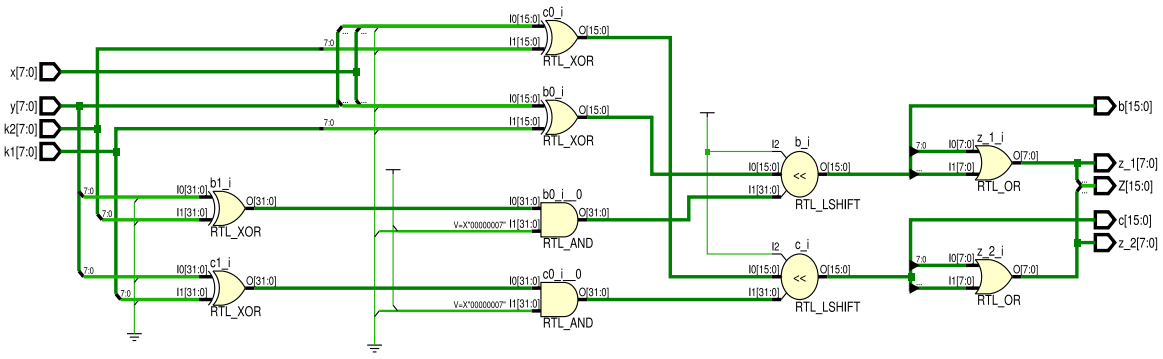

Fig. 8. Logic diagram of the synthesized $MRot_{(K)}(x, y)$ function.

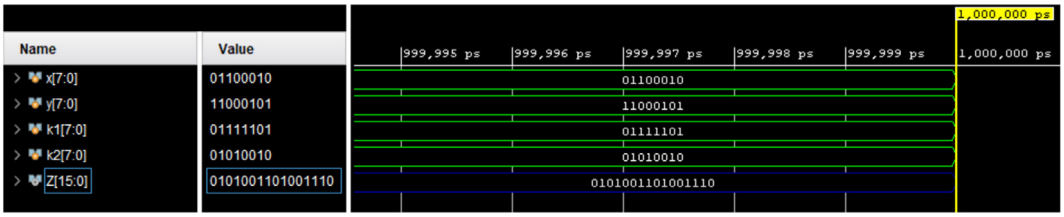

Fig. 9. Simulation results of the $MRot_{(K)}(x, y)$ function.

Table 7
Implementation comparison.

Function	Area (LUT)	TP (Mbps)	TP/Area (Mbps/LUT)
SIMON 96/96	435	1041	2.39
SPECK 96/96	452	1622	3.59
PRESENT-80	311	1084	3.49
LED-80	358	111	0.31
TWINE-80	306	1170	3.82
MRot	65	1746	**26.86**

out the simulation, synthesis, and RTL schematic of the proposed $MRot_{(K)}(x, y)$ function. Moreover, we compare our work with other existing proposed functions used for employing RFID authentication protocols [45].

An RTL analysis of the proposed $MRot_{(K)}(x, y)$ function is illustrated in Fig. 8. This figure describes the flow of signals between logic gates and shift registers. In this diagram, 8-bit x, 8-bit y, and 16-bit secret key ($K = k_2 \parallel k_1$) are inputs, and 16-bit Z is the output. It is noteworthy that, our proposed $MRot_{(K)}(x, y)$ function consists of only two 32-bit XOR, two 16-bit XOR, two 32-bit AND, and two 8-bit OR logic functions, and two 48-bit shift registers that illustrates the efficiency of the proposed function. In Table 7, we have shown the implementation comparison of our proposed $MRot_{(K)}(x, y)$ function with previously proposed other existing functions [45]. The parameters considered for the comparisons are area (number of LUTs), throughput (TP) (Mbps) and the throughput-to-area (TA/Area) ratio (Mbps/LUT). As shown in Table 7 the exact device utilization of the simulation after synthesis of the proposed $MRot_{(K)}(x, y)$ is 65 look-up-tables (LUTs) which is significantly less than the encryption algorithms like the PRESENT-80 which requires 311 LUTs [45]. Moreover, our proposed $MRot_{(K)}(x, y)$ function has the highest TP/Area which shows that $MRot_{(K)}(x, y)$ function is more lightweight than the others. The simulation window in Fig. 9 shows the result of $MRot_{(K)}(x, y)$ function in which the input values are $x = 01100010$, $y = 11000101$ and $K = k_2 \parallel k_1 = 0101001001111101$. As a result, the output value of the $MRot_{(K)}(x, y)$ is 0101001101001110.

In our proposed protocol SecLAP, for tag side, we use only $MRot_{(K)}(x, y)$ function, which makes the whole protocol execution very lightweight. Moreover, the number of flows we used for our proposed SecLAP protocol is 9 whereas for the LRMI protocol the number of flows used is 13, which also proves that the communication cost for our proposal is less than the LRMI protocol.

9. Conclusion

In the MIoT system, RFID-based sensors are the primary intelligent wireless devices employed to transfer the vital data obtained from a patient to a secure cloud-based platform. This system communicates the information of thousands of patients, and it also provides a real-time analysis of this information. Hence employing MIoT in the health-care industry requires security for the safe communication of information between the devices. In this paper, we showed that the recently proposed LRMI [3] protocol for lightweight RFID systems in the context of MIoT is not secure. We found and proved that the LRMI protocol could not provide all the essential security requirements, and it is vulnerable to secret disclosure, reader impersonation, and tag traceability attacks. Moreover, we showed that in LRMI protocol, the anonymity of the tag does not hold. Then, we proposed an authentication protocol called SecLAP, which is robust against the attacks that we identified in LRMI [3] protocol by employing our proposed lightweight function $MRot(\cdot)$. Regarding the discussion presented in Section 3, we proved that our proposed $MRot(\cdot)$ function is secure, and we used the BAN-logic method to validate the security features of our proposed SecLAP protocol. For implementation, we used an FPGA to simulate our proposed function and showed that it is easy to implement it over hardware. In particular, we conducted our proposed $MRot(\cdot)$ function on a Xilinx Kintex-7 (XC7k480t) FPGA using VIVADO 2018.1 software suite. The results indicate that the exact device utilization of the simulation after synthesis is 65 LUTs, and $MRot(\cdot)$ has a high TP/Area which shows that it is significantly lightweight and has low-cost requirements for being employed in RFID-based IoT systems. We also compared our proposed $MRot(\cdot)$ function with other existing functions and showed that it is better in terms of required area and throughput. In the future, we are planning to implement our proposed scheme by considering all the sequences

of the scheme and also employ it in the real world application of MIoT.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.future.2019.07.004.

References

[1] A. Satoh, K. Takano, A scalable dual-field elliptic curve cryptographic processor, IEEE Trans. Comput. 52 (4) (2003) 449–460.

[2] S. Karthikeyan, M. Nesterenko, RFID Security without extensive cryptography, in: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, 2005, pp. 63–67.

[3] K. Fan, W. Jiang, H. Li, Y. Yang, Lightweight RFID protocol for medical privacy protection in IoT, IEEE Trans. Ind. Inf. 14 (4) (2018) 1656–1665.

[4] N.J. Hopper, M. Blum, Secure human identification protocols, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2001, pp. 52–66.

[5] A. Juels, S.A. Weis, Authenticating pervasive devices with human protocols, in: Annual International Cryptology Conference, Springer, 2005, pp. 293–308.

[6] J. Bringer, H. Chabanne, E. Dottax, HB^{++}: a lightweight authentication protocol secure against some attacks, in: Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU'06), IEEE, 2006, pp. 28–33.

[7] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, 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", Springer, 2006, pp. 352–361.

[8] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, A. Ribagorda, M^{2}AP: A minimalist mutual-authentication protocol for low-cost RFID Tags, in: International Conference on Ubiquitous Intelligence and Computing, Springer, 2006, pp. 912–923.

[9] T. Li, Employing lightweight primitives on low-cost RFID tags for authentication, in: Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th, IEEE, 2008, pp. 1–5.

[10] H. Gilbert, M. Robshaw, H. Silvert, An active attack against HB+ - A provably secure lightweight authentication protocol, Tech. rep., Cryptology ePrint Archive, Report 2005/237, 2005, available at http://eprint.iacr.org/2005/237.pdf.

[11] K. Ouafi, R. Overbeck, S. Vaudenay, On the security of hb# against a man-in-the-middle attack, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2008, pp. 108–124.

[12] S. Islam, Security analysis of LMAP using AVISPA, Int. J. Secur. Netw. 9 (1) (2014) 30–39.

[13] M. Safkhani, N. Bagheri, M. Naderi, S.K. Sanadhya, Security analysis of lmap^{++}, an RFID authentication protocol, in: Internet Technology and Secured Transactions (ICITST), 2011 International Conference for, IEEE, 2011, pp. 689–694.

[14] F. Zeng, H. Mu, X. Wen, An improved LMAP++ protocol combined with low-cost and privacy protection, in: Advanced Technologies, Embedded and Multimedia for Human-Centric Computing, Springer, 2014, pp. 847–853.

[15] L. Kulseng, Z. Yu, Y. Wei, Y. Guan, Lightweight mutual authentication and ownership transfer for RFID systems, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–5.

[16] S. Kardas, M. Akgün, M.S. Kiraz, H. Demirci, Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems, in: Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec), 2011 Workshop on, IEEE, 2011, pp. 20–25.

[17] Z.-Y. Cheng, Y. Liu, C.-C. Chang, S.-C. Chang, Authenticated RFID security mechanism based on chaotic maps, Secur. Commun. Netw. 6 (2) (2013) 247–256.

[18] M. Akgün, T. Uekae, M.U. Caglayan, Vulnerabilities of RFID security protocol based on chaotic maps, in: 2014 IEEE 22nd International Conference on Network Protocols, IEEE, 2014, pp. 648–653.

[19] M. Benssalah, M. Djeddou, K. Drouiche, Security enhancement of the authenticated RFID security mechanism based on chaotic maps, Secur. Commun. Netw. 7 (12) (2014) 2356–2372.

[20] M. Akgün, A.O. Bayrak, M.U. Çalayan, Attacks and improvements to chaotic map-based RFID authentication protocol, Secur. Commun. Netw. 8 (18) (2015) 4028–4040.

[21] W. Zhu, J. Yu, T. Wang, A security and privacy model for mobile RFID systems in the internet of things, in: Communication Technology (ICCT), 2012 IEEE 14th International Conference on, IEEE, 2012, pp. 726–732.

[22] I. Erguler, A potential weakness in RFID-based internet-of-things systems, Pervasive Mob. Comput. 20 (2015) 115–126.

[23] K. Fan, Y. Gong, C. Liang, H. Li, Y. Yang, Lightweight and ultralightweight RFID mutual authentication protocol with cache in the reader for IoT in 5G, Secur. Commun. Netw. 9 (16) (2015) 3095–3104.

[24] S.F. Aghili, M. Ashouri-Talouki, H. Mala, DoS, impersonation and de-synchronization attacks against an ultra-lightweight RFID mutual authentication protocol for IoT, J. Supercomput. (2017) 1–17.

[25] Y.-P. Liao2014, C.-M. Hsiao, A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol, Ad Hoc Netw. 18 (2014) 133–146.

[26] Z. Zhao, A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem, J. Med. Syst. 38 (5) (2014) 1–7.

[27] K. Srivastava, A.K. Awasthi, S.D. Kaul, R.C. Mittal, A hash based mutual RFID tag authentication protocol in telecare medicine information system, J. Med. Syst. 39 (1) (2014) 153.

[28] C.-T. Li, C.-Y. Weng, C.-C. Lee, A secure RFID tag authentication protocol with privacy preserving in telecare medicine information system, J. Med. Syst. 39 (8) (2015) 1–8.

[29] S.S.S. GhaemMaghami, M. Mirmohseni, A. Haghbin, A privacy preserving improvement for SRTA in telecare systems, CoRR abs/1510.04197 (2015).

[30] J.-S. Chou, An efficient mutual authentication RFID scheme based on elliptic curve cryptography, J. Supercomput. 70 (1) (2014) 75–94.

[31] Z. Zhang, Q. Qi, An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography, J. Med. Syst. 38 (5) (2014) 1–7.

[32] M.S. Farash, An improved password-based authentication scheme for session initiation protocol using smart cards without verification table, Int. J. Commun. Syst. 30 (1) (2014) e2879.

[33] D. He, N. Kumar, N. Chilamkurti, J.-H. Lee, Lightweight ECC based RFID authentication integrated with an ID verifier transfer protocol, J. Med. Syst. 38 (10) (2014) 1–6.

[34] C.-I. Lee, H.-Y. Chien, An elliptic curve cryptography-based RFID authentication securing E-Health system, Int. J. Distrib. Sens. Netw. 11 (12) (2015) 642425.

[35] N. Kumar, K. Kaur, S. C. Misra, R. Iqbal, An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud, Peer-to-Peer Netw. Appl. 9 (2015).

[36] C. Jin, C. Xu, X. Zhang, F. Li, A secure ECC-based RFID mutual authentication protocol to enhance patient medication safety, J. Med. Syst. 40 (1) (2016) 1–6.

[37] C. Jin, C. Xu, X. Zhang, J. Zhao, A secure RFID mutual authentication protocol for healthcare environments using elliptic curve cryptography, J. Med. Syst. 39 (3) (2015) 1–8.

[38] F. Wu, L. Xu, S. Kumari, X. Li, A.K. Das, J. Shen, A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications, J. Ambient Intell. Humaniz. Comput. 9 (4) (2018) 919–930.

[39] M. Benssalah, M. Djeddou, K. Drouiche, A provably secure RFID authentication protocol based on elliptic curve signature with message recovery suitable for m-health environments, Trans. Emerg. Telecommun. Technol. 28 (11) (2017) e3166.

[40] T.-L. Lim, T. Li, T. Gu, Secure RFID identification and authentication with triggered hash chain variants, in: 2008 14th IEEE International Conference on Parallel and Distributed Systems, IEEE, 2008, pp. 583–590.

[41] T. van Deursen, 50 ways to break RFID privacy, in: Privacy and Identity Management for Life IFIP Advances in Information and Communication Technology, vol. 352, 2011, pp. 192–205.

[42] V. Lyubashevsky, D. Masny, Man-in-the-middle secure authentication schemes from LPN and weak PRFs, in: R. Canetti, J.A. Garay (Eds.), CRYPTO (2), in: Lecture Notes in Computer Science, 8043, Springer, 2013, pp. 308–325.

[43] K. Ouafi, R.C.-W. Phan, Privacy of recent rfid authentication protocols, in: International Conference on Information Security Practice and Experience, Springer, 2008, pp. 263–277.

[44] M. Burrows, M. Abadi, R. Needham, A logic of authentication, ACM Trans. Comput. Syst. (TOCS) 8 (1) (1990) 18–36.

[45] W. Diehl, F. Farahmand, P. Yalla, J.-P. Kaps, K. Gaj, Comparison of hardware and software implementations of selected lightweight block ciphers, in: Field Programmable Logic and Applications (FPL), 27th International Conference on, IEEE, 2017, pp. 1–4.

Seyed Farhad Aghili received his M.S. degree in Electrical Engineering from Shahid Rajaee Teacher Training University (SRTTU) in 2013. He is currently a Ph.D. candidate at the Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan. His current research interest includes RFID and IoT systems security.

Hamid Mala received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 2003, 2006 and 2011, respectively. He joined University of Isfahan (UI) in September 2011 as an Assistant Professor in the Department of Information Technology Engineering. Currently, he is with the Faculty of Computer Engineering at UI. His Research interests include design and cryptanalysis of block ciphers, digital signatures, cryptographic protocols and secure multiparty computation.

Pallavi Kaliyar is currently a Ph.D. student in school of Brain Mind and Computer Science at the University of Padova, Italy with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). Here, she is part of the SPRITZ Security and Privacy Research Group under the supervision of Prof. Mauro Conti. She received her Master of Technology in Computer Science and Engineering in 2012 and Bachelor of Engineering in Computer Science and Engineering in 2008. She is conducting research on fields including security and communication reliability related to the Internet of Things. For additional information: https://sites.google.com/site/pallavikaliyar/.

Mauro Conti is Full Professor at the University of Padua, Italy, and Affiliate Professor at the University of Washington, Seattle, USA. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU (2008, 2016), UCLA (2010), UCI (2012, 2013, 2014, 2017), TU Darmstadt (2013), UF (2015), and FIU (2015, 2016). He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco and Intel. His main research interest is in the area of security and privacy. In this area, he published more than 200 papers in topmost international peer-reviewed journals and conference. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Information Forensics and Security, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, and General Chair for SecureComm 2012 and ACM SACMAT 2013. He is Senior Member of the IEEE. For additional information: http://www.math.unipd.it/~conti/.