J Supercomput (2017) 73:1085–1102
DOI 10.1007/s11227-016-1849-x

Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags

Aakanksha Tewari[1] · **B. B. Gupta**[1]

Published online: 22 August 2016

Abstract Internet of Things (IoT) is an evolving architecture which connects multiple devices to Internet for communication or receiving updates from a cloud or a server. In future, the number of these connected devices will increase immensely making them an indistinguishable part of our daily lives. Although these devices make our lives more comfortable, they also put our personal information at risk. Therefore, security of these devices is also a major concern today. In this paper, we propose an ultra-lightweight mutual authentication protocol which uses only bitwise operation and thus is very efficient in terms of storage and communication cost. In addition, the computation overhead is very low. We have also compared our proposed work with the existing ones which verifies the strength of our protocol, as obtained results are promising. A brief cryptanalysis of our protocol that ensures untraceability is also presented.

Keywords Internet of Things · Authentication · Confidentiality · RFID tags · Anonymity

1 Introduction

The notion of Internet of Things (IoT) is evolving very fast since past few years. It can be described as the 'network of globally connected objects'. These objects or things might be computers, household appliances, vehicles, gadgets or even human beings and animals, etc., which are connected to the Internet making their location and other

✉ B. B. Gupta
gupta.brij@gmail.com

[1] Department of Computer Engineering, National Institute of Technology Kurukshetra, Kurukshetra 136119, Haryana, India

data available on the Internet. IoT helps objects to communicate easily and access services and get updates when required. This new area is built up on various concepts such as, wireless sensor networks, smart grid, smart homes, intelligent transportation, smart cities, radio frequency identifier (RFID), etc. [1–6]. RFID technology does not require any human assistance for its functioning. It was developed at the Auto-ID center. One of the major technologies behind IoT is the identification of the objects and it can be identified using the RFID tag embedded within the devices. RFID tags are capable of communicating with other objects and database server, identification of objects, etc. The RFID technology has contributed a lot towards the development of smart objects which are capable of communication [7,8].

IoT being a new area of development, has very weak security mechanisms, and hence is susceptible to all kinds of attacks. Threats to IoT may be sophisticated or by an insider as they can operate from remote areas or from within the organization. Attacks may also target a certain device or network or a person as in social engineering or phishing [9].

Considering the attack patterns, we present an attack mechanism (shown in Fig. 1) followed by the adversary while planning an IoT attack (or any other attack campaign). There are four steps taken by the attacker while carrying out IoT attack campaign (which can be generalized for any security attack) that are mentioned below:

- *Plan* The first step taken by the adversary is planning of the attack, which is not specific to IoT only. Before carrying out an attack, the first step is to gain as

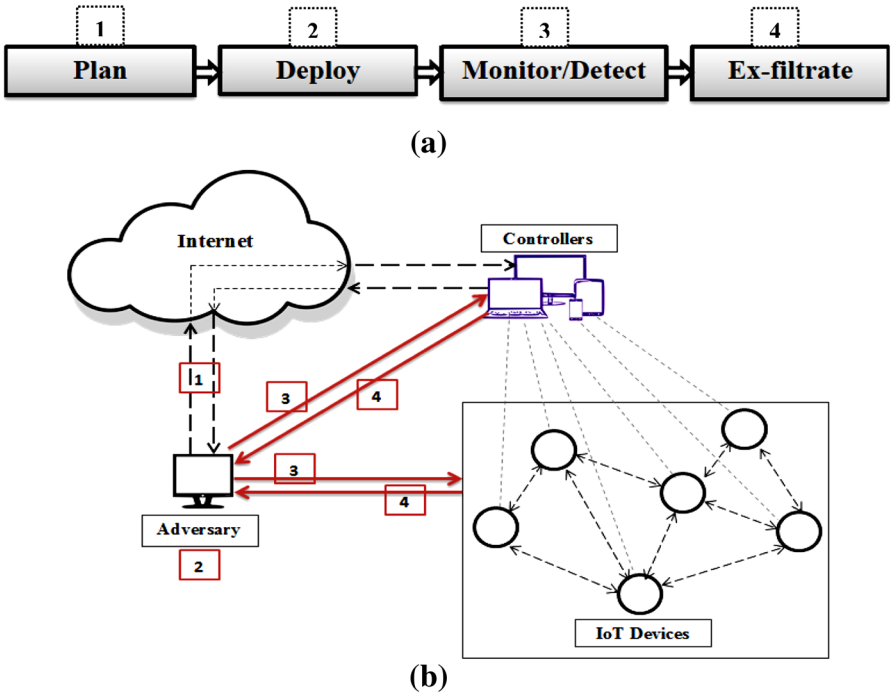

Fig. 1 **a** Stages in an attack. **b** IoT attack mechanism

much information as possible about the network that is to exploited. They can gain information by listening to messages between the IoT devices and server or between the devices.

- *Deploy* After collecting sufficient information, the attacker plans the attack campaign, i.e., deciding the medium of attack, and getting access to the IoT devices to install malicious software and extracting information using a remote system.
- *Monitor/detect* After the malware is successfully installed in the devices of an IoT network, the next step is to monitor the communication and other functionalities of the IoT objects and gather the required confidential information or take network down, etc.
- *Ex-filtrate* After fulfilling their objectives, the adversary's next aim is to ex-filtrate out of the network without being detected as to end their campaign successfully.

The term IoT was coined by Kevin Ashton in 1999, who was the director of the MIT's auto-ID center at that time. In the late 1940s, the first use of Radar technology took place which was the first use of RFID technique and in 1948 the RFID technology was created by Harry Stockman at Auto-ID center. In the year 1959, the IFF long-range RFID system was functioning commercially. The first RFID transponder system was created in 1973 and in 1979 RFID chip were placed in objects. After the origination of IoT in 1999, MIT's Auto-ID center had started working on RFID technology to develop IoT solutions. In the year 2006, first European IoT conference was held. After that, several major developments have been done in the field. Figure 2 presents brief overview of RFID and IoT history [9–12].

The RFID technology [13,14] was initially developed to replace the bar code of the retail objects. After development of IoT, it has become one of its cornerstones and separate Internet protocols for smart objects have also been devised. In the coming years, it is expected that a single addressing scheme will be able to successfully identify the IoT objects. However, in the current scenario, RFID is an elegant and low-cost system able to satisfy all the prerequisites of IoT objects. They are also effective in terms of lifetime, as they depend on the reader for power requirements, thus their

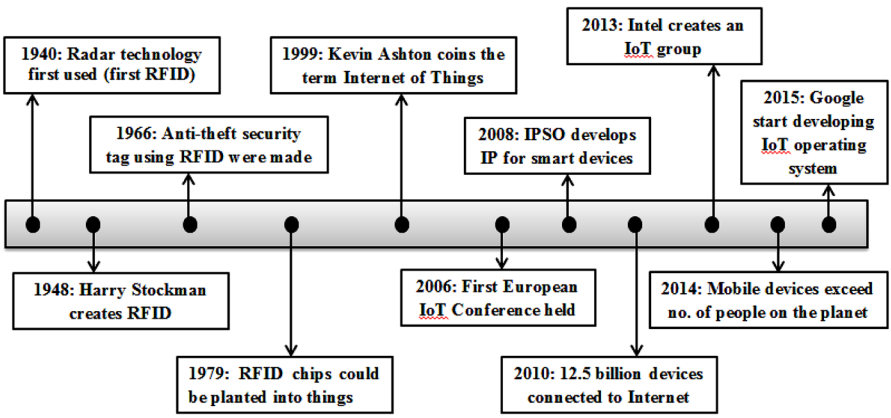

Fig. 2 Brief history of IoT and RFID technology

lifetime is infinity and they do not require any battery back-up which also reduces cost of the technology; moreover, there is no overhead of checking or replacing the batteries. However, issues related to security and privacy also need to be evaluated especially for such low-cost systems, as they have very limited resources they cannot use the traditional security algorithms [15,16]. Therefore, in this paper, we propose an ultra-lightweight mutual authentication protocol to ensure the security and privacy of IoT devices where the RFID tag authenticates the reader and vice versa before exchanging important data. Our protocol uses only two bitwise operations, i.e., XOR and left rotation.

The rest of the paper is organized as follows: Sect. 2 presents related work. In Sect. 3, our proposed approach is discussed in detail. Sections 4 and 5 discuss the efficiency of our work in terms of security and performance. Finally, Sect. 6 concludes our paper and discusses the future work.

2 Related work

RFID's applications are widespread these days, it was observed that there were about 3–4 billion things that had RFID tag in 2013. There is no doubt that RFID has become dynamic part of retail commerce all over the world. The Internet of Things is built up of some important components to ensure proper communication between the devices. RFID play an important role here, it makes the device or the object identifiable and therefore allows it communicate and monitor the data in real time, which enable the user to make better decisions and be aware of the current situation [17–19]. Moreover, the use of RFID has accelerated the development of the Internet of Things [20].

The IoT devices have limited resources and require updation and identification of other objects as well as server. Mutual authentication using RFID tag is the most common way to secure the devices from intrusion and ensure data integrity and confidentiality. However, due to the limited computation and storage capabilities, we need some lightweight mechanisms for this purpose [21].

Authentication protocols using RFID tags are classified into four types [22]: first is full-fledged class of protocols which consists of protocols which use classical cryptographic encryption and decryption protocols and have large computational overhead; the second class comprises of simpler protocols which require techniques such as elliptic curve cryptography, random number generator and hash function, etc., for mutual authentication purpose. The third class consists of lightweight mutual authentication protocols which use comparatively simple functions like random number generators, checksums, etc. The fourth class of protocols is the ultra-lightweight protocols which only use bitwise operations such as OR, AND, XOR, rotation or permutation, etc. These have the lowest overhead in terms of storage and computation.

In SASI [12], each tag has an ID and shares a pseudonym (IDS) and key value with the back end database server. The length of each of them is 96-bits. SASI ensures strong authentication and integrity. It uses bitwise XOR ($\oplus$), bitwise OR ($\vee$), bitwise AND ($\wedge$), addition mod $2n(+)$ and left rotate ($Rot(x, y)$) operation which left rotates the value of x with y bits. Complex operations such as hash functions are not used by this protocol. However, this protocol is susceptible to disclosure attacks and does not ensure untraceability.

Henrici et al. [23] presented a mutual authentication technique which makes use of one-way hash functions for secure communications and after every successful protocol run the Tag ID is updated by both back end database server and the tag, to ensure tag anonymity and location privacy; however, it failed to provide backward un-traceability.

Molnar et al. [24] presented a protocol based on the hash tree approach so that computation cost is evenly distributed among all the devices (or nodes). They also proposed a new library model in order to ensure stronger security at the back end server. Although the library architecture is very efficient, the protocol does not ensure tag anonymity.

In Weis et al. [25] proposed protocol based on the hash lock approach which a Meta ID for the authentication, the Meta ID is the hash of some random number. On receiving the Meta ID, if the database finds match, then it sends a key value and authentication is successful. Rhee et al. [26] proposed a challenge response scheme that uses random numbers and one-way hash for authentication, the secret is updated after each protocol run, unlike [27] it provides security from spoofing and replay attacks but this protocol does not ensure forward secrecy. Juels et al. [28] proposed a human computer interaction based protocol which used the parity concept and symmetric key authentication, but this approach is susceptible to noise on parity values.

Peris-Lopez et al. [29] proposed LMAP protocol which used simple bitwise operations XOR ($\oplus$), bitwise OR ($\vee$), bitwise AND ($\wedge$), and addition mod 2m (+). This protocol ensures mutual authentication and security from various attacks without the use of complex operations like hashing, etc. This scheme uses an index pseudonym (IDS) which is 96-bit in length. Here the IDS is the index of the row where all the tag-related data are stored. Each tag has key which is divided into four parts of 96-bits each.

M2AP [30] protocol which is very similar to LMAP, is also a lightweight mutual authentication protocol for RFID tags, where the index pseudonym updation procedure is different from LMAP while key updating operations remain the same. Both LMAP and M2AP ensure anonymity and mutual authentication and provide security against various attacks such as replay attacks and man-in-the-middle attacks, etc. However, both of these protocols are susceptible to de-synchronization and full-disclosure attacks.

Another protocol EMAP [31] which is based on challenge–response mechanism is an authentication scheme for passive tags. Most of the complex computations in this protocol are performed by the reader and tags perform lightweight operations such as hash, etc. It only requires one storage unit for the tag in addition to the ID for storing authentication related data. This protocol also ensures confidentiality, integrity and un-traceability.

Peris-Lopez et al. proposed the Gossamer protocol [31], which addresses the weaknesses of SASI [22] such as de-synchronization and disclosure attacks. It uses dual rotations and mixbits operation which is a lightweight function (combination of bitwise right shift and addition operations), although this protocol has low throughput.

The problem of ensuring security and privacy in RFID tags is of primary concern. The channel between the reader and the RFID tag is generally susceptible to various attacks as the traditional cryptographic schemes cannot be applied to these channels due to limited resources and computation power. Thus, in order to ensure security,

RFID tags can implement simpler protocols for mutual authentication between the reader and the RFID tag. The protocol should be designed in such a way that it not only provides authentication but also integrity, confidentiality and tag anonymity in addition to defending against various attacks. Therefore, the motivation behind our work is to ensure a secure communication between the reader and tag with minimal cost and computations.

In this paper, we present an ultra-lightweight protocol for mutual authentication between the reader and the RFID tag, which uses only basic operations such as bitwise XOR and left-rotation (Rot(X,Y) where X is rotated by the hamming weight of Y). Our protocol ensures data confidentiality, integrity and tag anonymity and resistance to tracking; it is also secure from various attacks such as man-in-the-middle attacks, replay attacks, disclosure attacks, etc.

3 Proposed solution

In this section, we present our proposed protocol. The protocol involves the following entities: the tag, the reader and the back end database server. Moreover, our protocol uses only bitwise operations, i.e., bitwise XOR and rotation operations which makes it an ultra-lightweight solution for mutual authentication between reader and the RFID tag. The random number generation is performed by the server so it does not affect the performance of our protocol.

3.1 Preliminaries

Bitwise operations: As discussed earlier, our protocol uses two bitwise operations XOR ($\oplus$) and left rotation operation (*Rot (A,B)*) [33,34]. The bitwise XOR operation ($\oplus$) stands for bitwise addition modulo 2. *Rot (A,B)* rotates A left by *wt(B)* (mod 96) bits, where the bit length of A is 96 bits and *wt(B)* is the hamming weight of B which the number of 1's in B. The output we get from *Rot(A,B)* might be A with the probability 1/96, the probability distribution of *Rot(A,B)* is uniform. For example, consider two 8-bit strings:

Let, $A = 10111010$ and $B = 10000110$, and *Rot (A,B)* = left rotation of A by *wt(B) (mod 8)*, where, *wt (B) = 3, wt(B) (mod 8) = 3 mod 8 = 5.*

Therefore, *Rot(A,B) = Rot(A,5) = 01010111.*

3.2 Mutual authentication protocol

Our proposed protocol assumes the channel between the reader and back end database server to be secure, but the channel between the reader and the RFID tag is assumed to be susceptible to various attacks. Each tag stores two sets 96-bit tag ID and a pseudonym and a key value, i.e, {*IDS, K*} are shared by both back end database server and the tag, each of them are 96 bits. The database server and the tag also store the old values {IDS_{old}, K_{old}} from the previous authentication session. Thus, if at the end the values of {*IDS, K*} are not updated, the tag can be verified on the basis of old tag

value [35]. Since the resources and computational capabilities of the IoT devices is limited, we proposed an ultra-lightweight mutual authentication protocol. The steps followed by our protocol are given below:

1. The protocol execution is initialized with a *"hello"* message sent by the reader to the device's RFID tag.
2. On receiving the *"hello"* message from the reader, the device tag sends its pseudonym value $\{IDS_{new}, IDS_{old}\}$ to the reader.
3. Upon receiving the *IDS* values from the device tag, the reader matches it with the values in the back end database server. If the values are not matched then the protocol terminates as the tag is not valid. Two cases arise while matching the values sent by the tag:

   Case I: both $\{IDS_{new}, IDS_{old}\}$ sent by the tag match the corresponding IDS values $\{IDS_{new}, IDS_{old}\}$ stored in the database. Here we set:

   $$back\ end\ server\ database's(IDS_{old}) = back\ end\ server\ database's(IDS_{new});$$
   $$back\ end\ server\ database's(K_{old}) = back\ end\ server\ database's(K_{new});$$

   Case II: tag's (IDS_{old}) = back end server database's (IDS_{new}), which means in the last session, the updation process was not successful at the server side. Here we set:

   $$back\ end\ server\ database's(IDS_{new}) = back\ end\ server\ database's(IDS_{old}) = tag's(IDS_{new});$$
   $$back\ end\ server\ database's(K_{new}) = back\ end\ server\ database's(K_{old}) = tag's(K_{new});$$

   After this check, we use the values of IDS_{new} and K_{new} and refer to them as *IDS* and K for simplicity in all the further communications. Thus, the synchronization between the tag and the server remains unbroken even if at the end of authentication step, the value of S is intercepted by the attacker and does not reach the reader. While the authentication protocol is on run and the server does not receive the value S, it terminates until the session expires.
   If a match is found in the database, the PRNG now generates two 96-bit random numbers m and n. Then, it calculates P and Q which mask the random numbers using *IDS* and key values, respectively. Then it calculates R for the purpose of device tag authentication:

   $$P = IDS \oplus m \oplus n;\ Q = K \oplus n;$$
   $$R = Rot(Rot(K \oplus n, IDS), K \oplus m);$$

   The values (P, Q, R) are then sent to the device tag.
4. On receiving the values (P, Q, R), the device tag obtains m by XOR value of P and *IDS* and n by the XOR value of Q and K. Then, using these values of m and n, it calculates R'.

$$n = Q \oplus K;$$
$$m = P \oplus n \oplus IDS;$$
$$R' = Rot(Rot(K \oplus n, IDS), K \oplus m).$$

If this $R' = R$, then the device tag next calculates S which is for the reader authentication by the device tag as follows:

$$S = Rot(Rot(IDS \oplus n, K), R' \oplus m).$$

The value of S is then sent to the reader. If the values of R and R' do not match, then, the protocol is immediately terminated.

5. Upon receiving the value S, the reader now calculates the value S' as:

$$S' = Rot(Rot(IDS \oplus n, K), R \oplus m).$$

If $S' = S$, then mutual authentication process is successful.

The old values $\{IDS_{old}, K_{old}\}$ at the tag are changed as:

$$IDS_{old} = IDS_{new}; \text{ and}$$
$$K_{old} = K_{new}.$$

After the completion of this process, reader and device tag both will update their *IDS* and key value as:

$$IDS_{new} = Rot(IDS \oplus n, K \oplus n), IDS \oplus m);$$
$$K_{new} = Rot(R \oplus n, IDSm).$$

Both the device tag and back end database replace the previous values of (IDS_{new}, K_{new}) with the new values; therefore, if the attacker obtains *IDS* value for one execution of the protocol, it will not be valid for another protocol run. Figure 3 depicts the flow diagram of our protocol.

4 Security analysis

In this section, we prove the security of our approach by providing cryptanalysis of our algorithm using the general untraceable privacy model, to ensure that the tags in our approach cannot be traced. The model we describe is based on the Juels–Weis model for RFID protocols [36]. In [37], Phan has presented a cryptanalysis of the SASI protocol based on this model, which proves that SASI is vulnerable to tracking attacks. For our protocol we have also used the Juels–Weis model as in [37], to perform the cryptanalysis of our protocol.

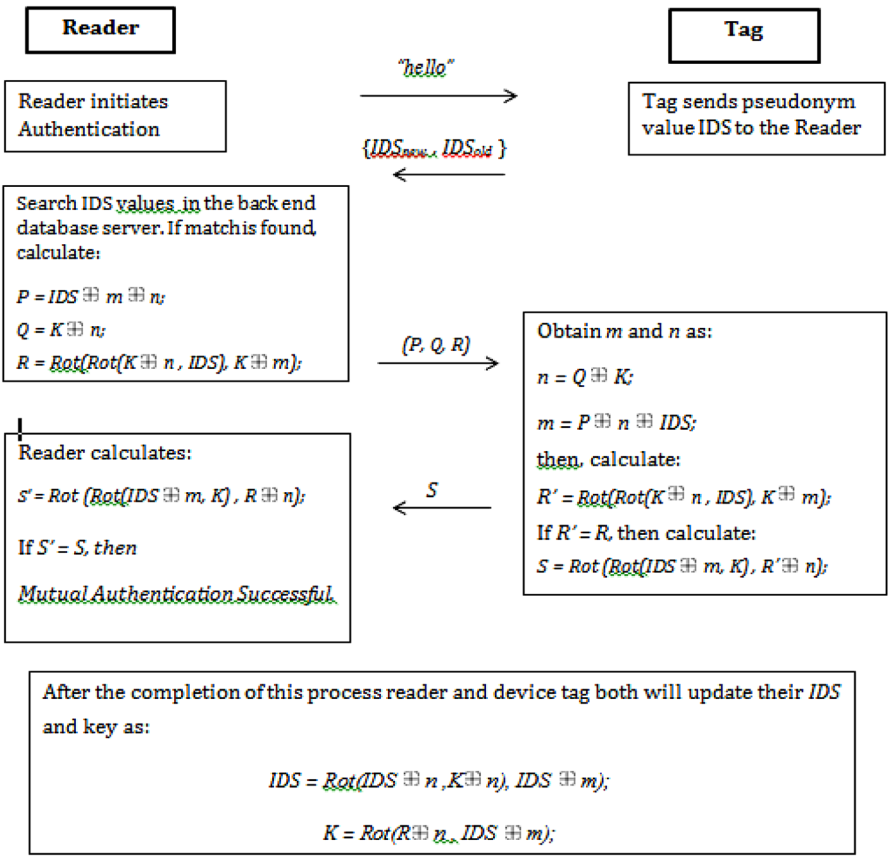

Fig. 3 Flow diagram of the proposed protocol

4.1 The Juels–Weis challenge–response model

This model comprises a system with one reader R and 'n' RFID tags, every tag has its own secret key and one pseudonym which are reset after successful execution of each session, and the following messages are used:

- To assign new secret key to the tag the *SetKey* message is used, when the tag receives *SetKey* message it discards the old key value and a new arbitrary secret is allotted to the tag.
- The *TagInit* message initializes the session key to a new value, discarding the current session details and issuing a new session key.
- The *ReaderInit* message is used by R to initialize a new session.

A is assumed to be an adversary which has the capability to generate any of the above messages. Once the tag is active it is entitled to respond to any number of challenge–response, i.e., (c_i, r_i) messages, which is based on the information from previous sessions parameters and challenges-response messages. The tag stores a log which details previous sessions and challenge–response message pairs.

Whenever the reader R receives the message (sid, r_i), it first evaluates a certain function based on its current status and all open as well closed sessions. On the basis of this function R outputs "accept" or "reject". A can corrupt any existing tag and issue any new tag as it able to send any number of SetKey messages. Thus, if a tag receives SetKey message from A, it is said to be corrupted.

In this approach, it is assumed that the adversary A listens and controls all the ongoing communications and among the reader and the tags. At the end of every protocol session, the reader or tag (T Tags) outputs Accept message if it finds the other to be legitimate.

A has the ability to issue the following queries:

- *Execute(R, T, i)*: here A eavesdrops on an actual execution of the protocol between reader (R) and tag (T) in session i. It is a passive attack.
- $Send(P_1, P_2, i, m)$: This query is the generalization of the challenge–response technique defined in Juels–Weis model along with the TagInit and ReaderInit messages. Here the adversary can impersonate a party P_1 which could be the reader or a tag during session 'i' and sends m messages to some other party P_2.
- *Corrupt(T, K)*: This query is same as the SetKey query, here the adversary changes the secret key of the tag to K. This is more powerful attack as compared to Send query as the adversary has access to the tag.
- $Test(T_1, T_2, i)$: When the Test query is issued, in the session 'i' depending upon b ∈ {0,1} an id ID_b is chosen from $\{ID_1, ID_2\}$ and A has to guess the bit 'b' correctly to succeed.

In the game that is to be played between the party and the adversary, the goal of the adversary is to identify the correct tag, and both of them must be fresh, i.e., it has not issued any corrupt queries. The following phases are considered in the game [38,39]

- *Phase I* (learning phase): During this phase, A is able to send any number of Execute, Send and Corrupt queries to learn about the tags and the reader.
- *Phase II* (challenge phase): Here A chooses a new session and sends a Test message to the session. The session must be fresh and selects a random bit ∈ {0, 1} depending on which it is given a tag to guess. A then continues making the queries, ensuring that the tags that are chosen for guessing remain fresh.
- *Phase III* (guessing phase): The game terminates when A outputs a bit value b, A wins if it successfully guesses the tag ID and is able to distinguish between T_0 and T_1. The success of A is quantitatively represented as advantage of A and denoted as:

$$\begin{aligned} \mathrm{Adv}_{\mathrm{A}}^{\mathrm{UNT}}(\mathrm{k}) &= |\Pr[A \text{ wins}] - \Pr[\text{random coin flip}]| \\ &= |\Pr[b' = b] - 1/2| \end{aligned}$$

In particular, consider an adversary A performing the following steps:

Learning phase: An execute query is issued so that A can listen to a protocol session between R and a tag T_1 to obtain the authentication variables R and S.

Challenge phase: A then chooses fresh tags T_1, T_2 with identifiers ID_1, ID_2 ($ID_1 \equiv 0 \, mod \, 2$ and $ID_2 \equiv 1 \, mod \, 2$). A then sends a Test query. Subsequently, adversary

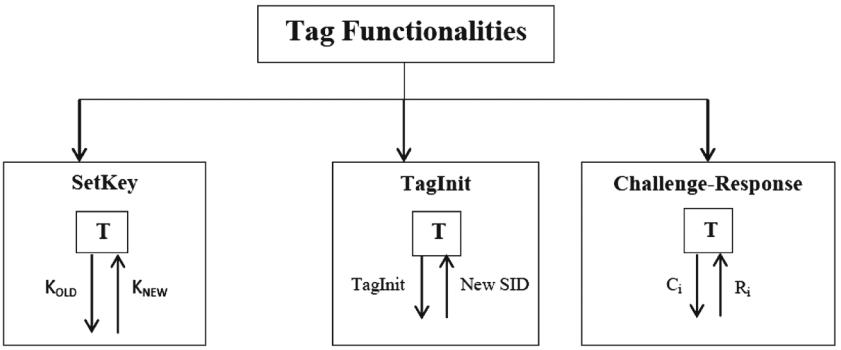

Fig. 4 Tag functionalities in the Juels–Weis model [36]

A is given a test challenge identifier $ID_b = \{ID_1, ID_2\}$. Note that by construction, $b = ID_{bLSB}$.

Guessing: The adversary A outputs a guess $b' \equiv S_{LSB} \oplus R_{LSB}$, which can be derived as:

$$\begin{aligned} Adv_A(k) &= |Pr[A\ wins\ ] - 1/2| \\ &= |Pr[b' = b] - -1/2| = |Pr[S \oplus R = b] - 1/2| \end{aligned}$$

Considering only the LSBs we have:

$$\begin{aligned} &= |Pr\left[S_{LSB} \oplus R_{LSB} = b\right] - 1/2| \\ &= |Pr\left[K_{LSB} \oplus n_{LSB} \oplus IDS_{LSB} \oplus m_{LSB}\right] - 1/2| << \in (k) \end{aligned}$$

Therefore, our advantage is negligible as compared to $\in (k)$. As we have:

$$\begin{aligned} R_{LSB} &= Rot\left(Rot\left(K_{LSB} \oplus n_{LSB}, IDS_{LSB}\right), K_{LSB} \oplus m_{LSB}\right) \\ S_{LSB} &= Rot\left(Rot\left(IDS_{LSB} \oplus m_{LSB}, K_{LSB}\right), R_{LSB} \oplus n_{LSB}\right) \end{aligned}$$

We have calculated the values of S_{LSB} and R_{LSB} using the truth tables and due to the use of rotation operation modulo 96, the values of R and S are always uncertain. Therefore, key and *IDS* values cannot be guessed by the adversary during the session. Tag and reader functionalities in the Juels–Weis Model are shown in Figs. 4 and 5, respectively.

4.2 Vulnerabilities due to modular operations

Here we perform the analysis of our protocol based on the modular operations, Hernandez–Castro et al. [40], have used this approach for the cryptanalysis of SASI. As we have also used rotation operations explicitly in calculating most of security

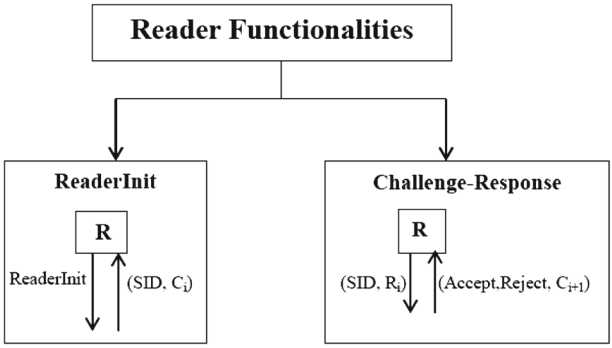

Fig. 5 Reader functionalities in the Juels–Weis model [36]

parameters, another way to analyze the security of our protocol is to check the conditions where modular operation do not work, i.e., the value (wt(B) (mod n) = 0). In such case, we get the same variable A without rotation. Therefore, we have:

$$R = Rot(Rot(K \oplus n, IDS), K \oplus m);$$
$$R = Rot(Rot(K \oplus n, IDS), 0).$$

Here, the probability of $(K \oplus m)\, mod n = 0$ is $1/n$ (where $n = 96$). In this case, we have:

$$R = Rot(K \oplus n, IDS).$$

Considering this case, our protocol will be safe as another rotation is to be performed and it will not give away any variable here. Therefore, if we also consider the case along with $K \oplus m$, $IDS \ mod \ n = 0$, we have:

$$R = Rot(K \oplus n, 0);$$
$$R = K \oplus n.$$

The probability of IDS mod n being zero will be $1/n$. The total probability of R being $K \oplus n$ will be $(1/n^2)$ which is very low and ensure high security for our scheme, and even if such case arises, the value $K \oplus n$ would not give away the key or any random variable as:

$$R \oplus P = (K \oplus n) \oplus (IDS \oplus m \oplus n);$$
$$= K \oplus IDS \oplus m;$$

and

$$R \oplus Q = (K \oplus n) \oplus (K \oplus n) = 0.$$

Similarly, $R \oplus S$ will also make no sense, thus in this case, our protocol is secure.

If we consider a similar case with the variable S, with the probability of $(1/n^2)$, we have:

$$S = IDS \oplus n.$$

Here, if we try to extract other parameters by the help of S, we get:

$$S \oplus P = (IDS \oplus n) \oplus (IDS \oplus m \oplus n) = m;$$

and

$$S \oplus Q = (IDS \oplus n) \oplus (K \oplus n) = K \oplus IDS.$$

Here also, $S \oplus R$ will make no sense, although the adversary has the value of random variable m, but if we examine our protocol carefully, the adversary will not be able to crack the values of any other variables with only 'm'. Hence, our protocol is not vulnerable due to modular operation used during the rotations.

4.3 Security against other attacks

In this section, we evaluate the strength of our protocol on the basis of the basic requirements of any mutual authentication protocol, i.e., confidentiality, integrity and tracking, etc., and the attacks from which it needs to be secured.

Confidentiality The messages that are sent over the insecure channel between device tag and the reader make use of the shared key, *IDS* and the random numbers. It is hard to obtain the random numbers and the device tag ID, because it is not send over the insecure channel. It is also hard to guess the Key value and *IDS* from P and Q as they are masked with the random numbers. The other variables R and S also have the key and *IDS* values masked and then left rotated on a certain hamming weight, so that data confidentiality is ensured.

Data integrity The variables R and S which are used to ensure mutual authentication also ensure the data integrity. Suppose the attacker modifies the values of P and Q, then the value of R' will be invalid and the protocol will terminate. It is hard for the attacker to modify the values such that the value of R' is correctly calculated. Thus, our protocol ensures data integrity.

Anonymity and tracking As shown in Sect. 4.1, the value of the key and the pseudonym are updated after every successful protocol run and the random number are also different in each run. Since the device tag ID is not send in any messages, the attacker cannot acquire the tag ID, ensuring device tag anonymity. Thus, the attacker is also able to track the device tag using the key and pseudonym value and variable transmitted during the protocol run, as they are different every time.

Forward secrecy It refers to securing the previous communications between the device tag and the adversary if the device tag is compromised. Suppose the attacker is somehow able to acquire the tag ID. The adversary is still not able to determine the previous conversations as it involves random numbers, pseudonym and a key value. Suppose there is a case where the attacker obtains the *IDS* and key value, the adversary is still unable to get the previous data as after protocol run the values of *IDS* and key are different and the communication also involve random numbers which mask these values.

Security against man-in-the-middle attacks Our scheme is secure against the man-in-the-middle attack as the values of P, Q, R and S if intercepted and changed will cause the value of R' and S' to be different. Consequently, the authentication will be unsuccessful. Moreover, the attacker is also not successful in getting key and pseudonym value as they are masked with random numbers.

Security against replay attacks Our scheme is secure against the replay attacks, as each message involves random numbers which are different; therefore, the protocol will terminate, if the attacker tries to replay older messages. The replays of messages will not have any effect on the device tag.

Security against de-synchronization attacks De-synchronization attacks disturb the synchronization between the device tag and the reader. The attacker may try to desynchronize the device tag and the reader by enabling them to use different values of m and n. The messages can be modified so that different values of random number one obtained by both parties, but such modification also causes the values of R to change in an unpredicted fashion, thus, ensure synchronization between the reader and the device tag. As we save, both new as well as the old values if *IDS* and key at both tag and the reader, the attacker cannot use fake *IDS* value to break the synchronization between them.

Security against disclosure attacks The attacker cannot obtain any information; even if they have P, Q, R and S, even if the adversary has *IDS* they are able to obtain any of the random number or the key value from P and Q. Similarly, the attackers cannot extract any values from R and S with the help of P and Q. Thus, our protocol is safe from disclosure attacks.

5 Performance analysis

In this section, we evaluate the performance of our proposed scheme on the basis of its computation, storage and communication cost for each device tag. As our approach only requires XOR ($\oplus$) and left rotation (Rot(.,.)), thus it has very low computation cost.

Each tag requires to store five 96-bit values which are: *IDS, K, m, n* and device tag ID. Thus, the total storage requirements is $5L$ bits where $L = 96$ (i.e., $5 \times 96 = 480$ bits) which is lesser than SASI. Each tag while undergoing mutual authentication sends

Table 1 Comparison of our protocol with existing ultra-lightweight protocols

	LMAP	M2AP	EMAP	SASI	RAPP	Gossamer protocol	Our protocol
Storage cost	6L	6L	6L	7L	5L	7L	7L
Communication cost	4L	5L	5L	4L	2L	5L	3L
Operation used	+, ⊕, OR	+, ⊕, AND, OR	⊕, AND, OR	+, ⊕, AND, OR, Rot(A,B)	⊕, AND, OR, Rot(A,B), Per(A,B)	+, ⊕, AND, Rot(A,B), MixBits	⊕, Rot(A,B)
Security from de-synchronization attacks	No	No	No	No	No	Yes	Yes
Security from disclosure attacks	No	No	No	No	Yes	Yes	Yes
Tracking	No	No	No	No	Yes	No	Yes

two messages each having one 96-bit value (i.e., *IDS* and *S*); thus, the communication cost for each device tag is $2L = 2 \times 96 = 192$ bits.

Table 1 shows the comparison of our work with existing ones. Our protocol performs significantly well as compared to SASI [22], LMAP [29], M2AP [30], EMAP[31] and RAPP [41] protocols in terms of security against attacks and storage and communication size. In addition, our protocol requires fewer operation as compared to others and thus without the use of AND and OR operations our protocol is safe from tango attacks.

6 Conclusion and future work

In this paper, we have discussed a new ultra-lightweight mutual authentication protocol for IoT device tags. Our scheme transmits data securely over an insecure channel ensuring security from various attacks such as DDOS, Replay and Tracking, etc., as discussed in the previous sections. We have also compared our approach with the existing approaches and we obtained promising results. In addition, our protocol uses only two bitwise operations for authentication, which makes it very efficient for IoT devices with limited resources and computation capabilities as these operations can be implemented with passive tags. The storage and communication cost is also very low as compared to other ultra-lightweight mutual authentication approaches. Our protocol is also secure from tango attacks as it does not use AND and OR operations for mutual authentication protocol. We also aim to study the security of our protocol against denial of service attacks at both tag and reader side. In future, we also aim to perform further in-depth cryptanalysis of our proposed approach.

Acknowledgements This research work is being funded by Department of Electronic and Information technology (DeitY), Ministry of Communications and IT, Government of India.

References

1. Ashton K (2009) That "Internet of Things" thing. RFiD J. http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf. Last accessed August 2016
2. Stergiou C, Psannis KE (2016) Recent advances delivered by mobile cloud computing and internet of things for big data applications: a survey. Int J Netw Manag. doi:10.1002/nem.1930
3. Shengdong X, Yuxiang W (2014) Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. Wirel Pers Commun 78(1):231–246
4. Guo P, Wang J, Li B, Lee S (2014) A variable threshold-value authentication architecture for wireless mesh networks. J Internet Technol 15(6):929–936
5. Psannis KE (2016) HEVC in wireless environments. J Real-Time Image Process 12(2):509–516
6. Psannis K (2009) Efficient redundant frames encoding algorithm for streaming video over error prone wireless channels. IEICE ELEX J 6(21):1497–1502
7. Buckley J (ed) (2006) The internet of things: from RFID to the next-generation pervasive networked systems. Auerbach Publications, New York
8. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54:2787–2805
9. Cisco (2016) "IoT Threat Environment". Available at: http://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/C11-735871.pdf. Last accessed July
10. Zorzi M, Gluhak A, Lange S, Bassi A (2010) From today's intranet of things to a future internet of things: a wireless- and mobility-related view. IEEE Wirel Commun 17:43–51

11. Ning HS, Wang ZO (2011) Future internet of things architecture: like mankind neural system or social organization framework? IEEE Commun Lett 15:461–463
12. Psannis KE, Xinogalos S, Sifaleras A (2014) Convergence of internet of things and mobile cloud computing. Syst Sci Control Eng Open Access J 2(1):476–483
13. Near Field Comminications History (2016) "Timeline of RFID technology". Available at: http://www.nfcnearfieldcommunication.org/timeline.html, Last accessed July
14. Postscapes (2016) "History of internet of things". Available at: http://postscapes.com/internet-of-things-history. Last accessed July
15. Roman R, Najera P, Lopez J (2011) Securing the internet of things. Computer 44(9):51–58
16. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (IoT): a vision, architectural elements, and future directions. Futur Gener Comput Syst 29(7):1645–1660
17. Welbourne E, Battle L, Cole G et al (2009) Building the Internet of things Using RFID: The RFID ecosytem experience. IEEE Computing Society. Available at: http://homes.cs.washington.edu/~magda/papers/welbourne-ieeeic09.pdf. Last accessed July 2016
18. Khoo B (2011) "RFID as an enabler of the internet of things: issues of security and privacy". In: Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing, pp. 709–712
19. Chris Edwards, (2016) " RFID tags along with the Internet of Things", Engineering and Technology magazine vol 9, Issue 8. Available at: http://eandt.theiet.org/magazine/2014/08/tagging-along.cfm, Last accessed July
20. Thrasher J (2016) "A primer on the internet of things and RFID". Available at: http://blog.atlasrfidstore.com/internet-of-things-and-rfid. Last accessed July
21. Bolic M, Simplot-Ryl D, Stojmenovic I (2010) RFID systems: research trends and challenges. Wiley, New York
22. Chien H-Y (2007) SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Trans Dependable Secur Comput 4(4):337–340
23. Henrici A, Muller P (2004) "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers". In: International Workshop on Pervasive Computing and Communication Security PerSec, Orlando, Florida, USA, pp 149–153 (ISBN: 0-7695-2106-1)
24. Molnar D, Wagner D (2004) "Privacy and security in library RFID: Issues, practices, and architectures". In: Conference on Computer and Communications Security—ACM CCS, Washington, DC, USA, pp 210–219 (ISBN:1-58113-961-6)
25. Weis SA, Sarma SE, Rivest RL, Engels DW (2004) Security and privacy aspects of low-cost radio frequency identification systems. Secur Pervasive Comput LNCS 2802:201–212
26. Rhee K, Kwak J, Kim S, Won D (2005) Challenge-response based RFID authentication protocol for distributed database environment. Int Conf Secur Pervasive Comput SPC 2005:70–84
27. Jules A (2006) RFID security and privacy: a research survey. IEEE J Sel Areas Commun 24(2):381–394
28. Juels A, Weis S (2005) Authenticating pervasive devices with human protocols. CRYPTO'05., vol 3126 of LNCS, IACR. Springer, Heidelberg, pp 293–308
29. Peris-Lopez P, Hernandez-Castro JC, Estevez-Tapiador J, Ribagorda A (2006) "LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags". Printed handout of Workshop on RFID Security -RFIDSec 06 July
30. Peris-Lopez P, Hernandez-Castro JC, Estevez-Tapiador J, Ribagorda A (2006) "M2AP: a minimalist mutual-authentication protocol for low-cost RFID tags". Lecture Notes in Computer Science, pp 912–923. Springer, Berlin
31. Peris-Lopez P, Hernandez-Castro JC, Estevez-Tapiador JM, Ribagorda A (2006) "EMAP: an efficient mutual authentication protocol for low-cost RFID Tags". OTM Federated Conferences and Workshop: IS Workshop, IS'06, 4277 Lecture Notes in Computer Science, pp 352–361. Springer, Berlin
32. Peris-Lopez P, Hernandez-Castro JC, Tapiador JME, Ribagorda A (2008) "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol,". In: Proceedings of International Workshop on Information Security Applications, pp 56–68
33. Kelsey J, Schneier B, Wagner D, Hall C (1998) Cryptanalytic attacks on pseudorandom number generators, Fast Software Encryption, LNCS, vol 1372, Springer, Berlin. pp 168–188 (ISBN: 978-3-540-69710-71998)
34. Erguler I, Unsal C, Anarim E, Saldamli G (2012) Security analysis of an ultra-lightweight RFID authentication protocol-SLMAP*. Secur Comm Netw 5:287–291

35. Tagra D, Rahman M, Sampalli S (2010) "Technique for preventing DoS attacks on RFID systems". In: Proceedings of 18th International Conference on Software Telecommunication and Computer Networks (SoftCOM'10), IEEE Computer Society
36. Juels A, Weis SA (2007) "Defining strong privacy for RFID". In: Proceedings of Fifth Ann IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '07), pp 342–347. http://eprint.iacr.org/2006/137
37. Phan R (2008) Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI. IEEE Trans Dependable Secur Comput 6(4):316–320
38. Ouafi K, Phan RC-W (2008) "Traceable privacy of recent provably-secure RFID protocols". Proceedings of Sixth Int'l Conf. Applied Cryptography and Network Security (ACNS '08), pp 479–489
39. Ouafi K, Phan R.C.-W (2008) "Privacy of recent RFID authentication protocols". In: Proceedings of Fourth Information Security Practice and Experience Conference (ISPEC '08), pp 263–277,
40. Hernandez-Castro JC, Tapiador JME, Peris-Lopez P, Quisquater J-J (2008) Cryptanalysis of the sasi ultralightweight rfid authentication protocol with modular rotations. Technical Report arXiv:0811.4257
41. Tian Y, Chen G, Li J (2012) A new ultralightweight RFID authentication protocol with permutation. IEEE Commun Lett 16(5):702–705