Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Sciences

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

Cloud-based lightweight secure RFID mutual authentication protocol in IoT

Kai Fan[a,1,*], Qi Luo[a], Kuan Zhang[b,1], Yintang Yang[c,1]

[a] State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China
[b] Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, NE 68588, USA
[c] Key Laboratory of Ministry of Education for Wide Band-Gap Semiconductor Materials and Devices, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 15 June 2018
Revised 20 July 2019
Accepted 2 August 2019
Available online xxx

Keywords:
Internet of things
Lightweight
Security and privacy
RFID
Mutual authentication

ABSTRACT

Radio Frequency Identification (RFID) is a supporting technology for the Internet of things (IoT). RFID enables all physical devices to be connected to IoT. When RFID is widely used and developing rapidly, its security and privacy issues cannot be ignored. The wireless broadcast channel between the tag and the reader may be subject to many security attacks, such as interception, modification, and replay. Messages from unverified tags or readers are also untrustworthy. A secure and stable RFID authentication scheme is critical to IoT. This paper puts forward an efficient and reliable cloud-based RFID authentication scheme. In order to reduce the RFID tag's overhead, the proposed authentication scheme explores the rotation and enhanced permutation to encrypt data. The proposed protocol not only resists the above common attacks and protects the privacy of the tag, but also adds the cloud server to the RFID system. Performance simulation shows that permutation and rotation are efficient. Security analysis shows that our protocol can resist various attacks, such as tracking, replay, and desynchronization attack. Mutual authentication and backward security are also achieved. Finally, we apply BAN logic to prove the security of the protocol.

1. Introduction

1.1. IoT and RFID

Cloud computing, big data and artificial intelligence are benefiting the Internet industry and providing strong technical support to the Internet of Things (IoT). IoT is a new revolution in science and technology with the goal of connecting all objects together. In the IoT era, the way of transmitting the information has been changed dramatically. Through various sensing devices and automatic identification technologies, IoT connects any items with the Internet for information exchange and communication in accordance with the agreement, to achieve intelligent analysis and processing of physical world [6].

As an important part of IoT, recognition technology deserves our attention. There are many traditional automatic identification technologies, such as barcode recognition technology, optical character recognition technology, biometric identification technology, identification technology of magnetic card and contact IC card identification technology. However, they still

* Corresponding author.
E-mail addresses: kfan@mail.xidian.edu.cn (K. Fan), kuan.zhang@unl.edu (K. Zhang), ytyang@xidian.edu.cn (Y. Yang).
[1] Member, IEEE

https://doi.org/10.1016/j.ins.2019.08.006
0020-0255/

have several limits when applied in IoT. For example, bar code can only store a small amount of data; the cost of optical character recognition is too high; the biological recognition is imperfect; magnetic card identification and contact IC card identification need close contact, which is inflexible. Some of these identification technologies do not yet have the ability to protect privacy. However, RFID is a non-contact automatic identification technology, without a mechanical or optical contact between the system and a specific target, and can protect user privacy by security protocols. These advantages make RFID one of the most promising technologies and have been widely used in the IoT [16].

An RFID system consists of RFID tags, RFID readers and a database server. Tags attached to objects identify them uniquely and store the identification information of objects. They exchange data with reader via radio signals [8]. In the traditional RFID system, the database server is a local back-end server. The communication channel between the server and the reader can be considered safe in this traditional RFID architecture. The backend server is completely trusted. For the case of a small number of tags and a low authentication frequency, the computing and storage performance of the backend server is sufficient. However, in order to achieve the Internet of Everything, there are numerous objects that need to be identified by the tag. When RFID systems generate a large amount of data, cloud computing is able to address the limited performance of back-end servers in the IoT environment. Thus, it is necessary to bring cloud platform into RFID system. In our protocol, cloud platform is applied to replace physical servers. Cloud platform has many outstanding advantages, including data storage, high scalability, and low-cost services [11,22]. The introduction of cloud computing greatly enhances the robustness and data processing capabilities of RFID systems. Robustness ensures the stability and reliability of the system, which is why cloud computing systems are rapidly evolving and deployed. In RFID system, almost all the data collected by the sensors is processed in the cloud platform, which can solve some serious problems, such as data delay and data loss [18].

However, the RFID system's security risks cannot be ignored. Due to that tags and readers use wireless broadcast channels to transmit data, the data is completely exposed to the outside world. Communication content may be intercepted, replayed and tampered. The privacy of the tag may be revealed. The reader and cloud server can communicate over a wireless, wired or dedicated channel. This complex channel is less secure than the channel between the reader and the local backend server. Public cloud servers that are widely used in the IoT are semi-trustworthy. The above characteristics make the RFID system vulnerable to attack. Therefore, a secure and reliable RFID authentication protocol is necessary for IoT.

1.2. Related work

Authentication protocols are the key to solve many security issues. Due to the limited computing power and storage space of the tags, security solutions are different from the traditional network. The biggest challenge of the RFID certification scheme is to design a secure, efficient and low-cost authentication protocol [25]. For the security problems of RFID system, many protocols have been proposed [4,19,20].

Some protocols are based on back-end servers. However, there are some problems. In the hash-lock protocol [17], the *metaID* used by each tag is always invariant, which makes it vulnerable to track. When the number of tags in the system is large, the reader authentication will take a long time in the hash-lock protocol [17] and the hash-chain protocol [13], which makes it difficult to apply in practice. Timestamp-based protocol [24] implements mutual authentication and does not have the same problems with the two protocols, but it cannot resist the synchronization attack.

Server-less RFID architecture is designed to identify and verify tags by using offline mobile readers [2,9,10]. The simple idea of offline authentication is to download an AL (Access List) from a CA (Certification Agency) into a mobile reader, which affects the real-time of the authenticity. Upon the rise of cloud computing, some cloud-based RFID schemes have been proposed.

In 2013, Xie [23] proposed a cloud-based RFID authentication protocol. In that paper, VPN is used to ensure the secure communication between the reader and the cloud. However, there is too much data transfer between the reader and the cloud server. The reader needs a lot of symmetric decryption operations. Cloud servers cannot delete asynchronous tag data.

In 2015, Abughazalah, Markantonakis and Mayes [1] proposed a secure improved cloud-based RFID authentication protocol. In their proposal, $H(IDi)$ is transmitted in plain text, and the next authentication ID_i^j is equal to $H(IDi)$, so that the protocol cannot guarantee the backward security and tags may be tracked.

Xiao [8] proposed a cloud-based RFID authentication protocol to resist the attacks that may arise in the insecure communication channel between the reader and the cloud server. The scheme prevents the location tracking attack even if the tag fails to update its identifier. The computational overhead of the tag is not fully satisfactory.

From the above introduction, it can be found that the security of the RFID protocol is gradually increasing. However, the security threats facing RFID are also increasing. Many of the current protocols do not achieve the desired security. Some newly proposed protocols use hash functions frequently to ensure adequate security, which increases the cost of the tag and the computational overhead of the system.

1.3. Our contribution

Motivated by the above discussions, the paper is aimed to propose a cloud-based RFID authentication protocol. This protocol can resist the attacks that may arise, and ensure that the tag overhead is lightweight. Protecting the privacy of tags is the focus of our protocol.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

(1) We exploit permutation and rotation operation instead of hashing operation to encrypt data, reducing the computational cost of tags.
(2) This scheme replaces random number generator with timestamps to update information and ensure the freshness of the message. Timestamps can also help the cloud server index tags. The cloud stores one or more timestamps for each tag to resist desynchronization attacks.
(3) In the whole process of authentication, the cloud server cannot obtain the important plaintext information of the tag, which further protects the privacy of the tag.

1.4. Organization

The rest of the paper is organized as follows: in Section 2, we briefly introduce the attack model and security requirements. In Section 3, the detailed steps of the proposed protocol are given. In Section 4, we analyze the security of the proposed protocol and compare it with other three protocols. Performance analysis and comparison is also given. In Section 5, we prove the proposed protocol by BAN logic. In Section 6, our conclusions are given.

2. Overview of attacks and security requirement

In this section, we briefly introduce the common attack methods and security requirements of RFID system.

2.1. Overview of attacks

2.1.1. Eavesdropping

An attacker may be near a tag or reader, listening for long periods of time for the communication between the tag and the reader. If there is significant plaintext information in the message, the attacker will get it. Since eavesdropping attack is a passive attack, the system is difficult to prevent.

2.1.2. Forging

An attacker may gain some significant identity information about the tag by eavesdropping attacks, thereby disguising himself as a legitimate tag and gaining reader authentication.

2.1.3. Replay

The attacker can obtain authentication messages in the communication between the tag and the reader by eavesdropping. Then the attacker replays the same messages again to pass the authentication.

2.1.4. Desynchronization

To counter replay attacks, tags and cloud servers may update information during each authentication process. If an attacker intercepts the important communication information during a session, updates between the tag and cloud may be inconsistent. In the next authentication, the legitimate tag cannot be recognized by the reader, resulting in the authentication failure. Some accidents or natural disasters can also cause the same problem.

2.1.5. Tracing

If an attacker can distinguish different tags by listening for certain information sent by tags, it can continuously track a particular tag, thereby obtaining the privacy of the tag, such as location information.

2.2. Security requirements

RFID systems face so many security threats that many requirements for authentication protocols are put forward. Next, the specific RFID system security requirements are introduced.

2.2.1. Privacy

Privacy is the most fundamental and important security requirement of RFID tags, including the privacy of the tag's location, identity, consumer activities and other information.

2.2.2. Confidentiality

It means that important information is not available to unauthorized entities. If critical authentication information is obtained by an attacker, it can impersonate a tag or a reader to carry out other attacks. In this way, the attacker breaks the system and obtains tags' privacy.

2.2.3. Availability

It means that in the RFID system, the legitimate tag can be authenticated by the system; legitimate readers can read legitimate tag and unimpeded access to data in the cloud service.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

Table 1
Notation and description.

Notation	Description
IDi	the identity of an RFID tag
Tt	the timestamp stored in the tag
Rt	the timestamp generated by an RFID reader
M( )	obscuring the timestamp such as turning 20181207172530 into 20181207170000
Per(*A, B*)	the operation of new permutation
Rot(*A, B*)	the operation of rotation
*E*1( )	the first symmetric encryption algorithm using a key shared between readers
*D*1( )	the first symmetric decryption algorithm using a key shared between readers
*E*2( )	the second symmetric encryption algorithm using a key shared between readers and cloud
*D*2( )	the second symmetric decryption algorithm using a key shared between readers and cloud
$(\,)_L$	the left half of the data
$(\,)_R$	the right half of the data
$\oplus$	the bitwise *XOR* operation

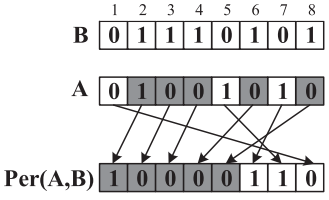

Fig. 1. The computation of the example.

2.2.4. Mutual authentication

In RFID systems, Mutual authentication between tags and readers are necessary. Reader's authentication of tags ensures the authenticity of collected information. The tag's authentication of the reader prevents the owner's private information from being obtained by the attacker.

2.2.5. Backward security

It means that the attacker obtains the current session and all previous session messages of the tag and the reader, and still cannot predict the content of the authentication information in the future session. Thus it is difficult to impersonate a legitimate tag, reader or, server.

3. Cloud-based lightweight secure RFID mutual authentication protocol in IoT

In this section, we first describe the notations and two operations, permutation and rotation, exploited in our scheme. The storage structure of the cloud sever is shown. Finally, we give a detailed introduction of our proposed efficient and reliable scheme named timestamp-permutation.

3.1. Notations

The definition of notations used in this paper is shown in Table 1. The protocol does not have back-end server. The database is stored in a public cloud. Since the public cloud is an untrusted third party, the storage of important data needs encrypting. The communication channel between the reader and the cloud is not secure. The timestamps in this protocol are based on reader's current time.

3.2. Permutation and rotation

Tian et al. [21] introduced a new operation—permutation. The permutation operation is defined as follows.
Suppose A and B are two n-bits strings, where

$$A = a1\ a2\ \ldots\ an,\quad ai \in \{0, 1\}, i = 1,\ 2, \ldots,\ n$$

$$B = b1\ b2\ \ldots bn,\quad bj \in \{0, 1\}, j = 1,\ 2, \ldots,\ n$$

Moreover, the Hamming weight of B is $wt(B)$ $(0 < < wt(B) < < n)$.
In Fig. 1. there are $b2 = b3 = b4 = b6 = b8 = 1$ and $b1 = b5 = b7 = 0$. Then, the permutation of A according to B, denoted as $Per(A, B)$, is $Per(A, B) = a2\ a3\ a4a6\ a8\ \ a7a5a1 = 10000\ 110$.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

Table 2
Storage structure.

Index		Content
Tt, 1	*Rt*, 0	*E*1(*IDi*)
Tt′, 1		*E*1(*IDj*)
…	…	…

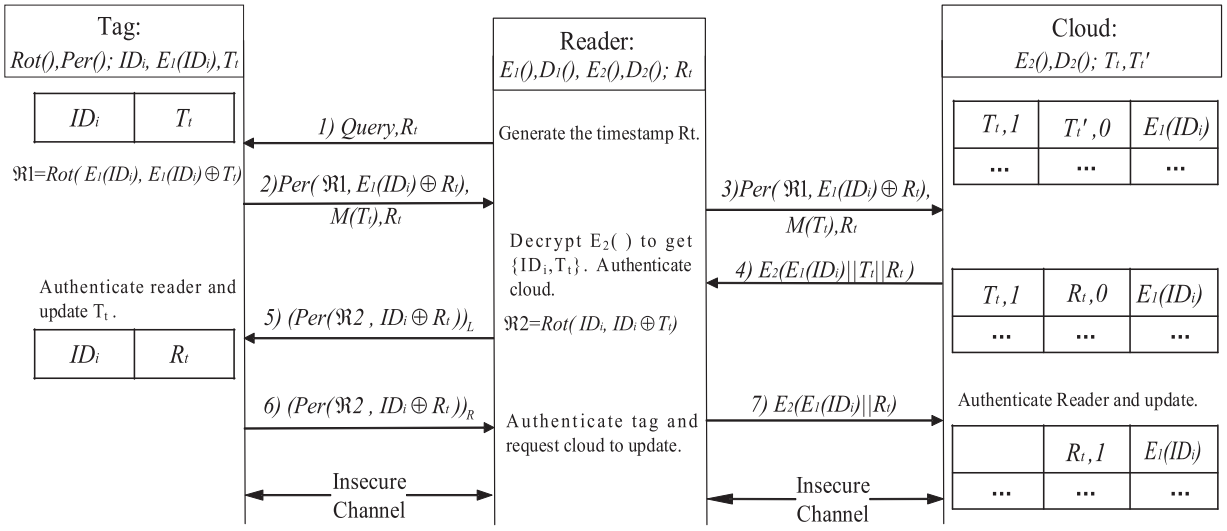

Fig. 2. Timestamp-permutation protocol.

However, there are two weaknesses in this permutation. One is that the rightmost bit of string *B* does not affect the result *Per*(*A, B*). That is to say, given $A = 01001010$, B=01110101, B'=01110100. Then we have $Per(A, B) = Per(A, B') = 10000110$.

The other weakness is the invariability of Hamming weight. It is obvious that,$wt(Per(A, B)) = wt(A)$. This property leaks some of *A*'s information and reduces the security. Therefore, we explore an improved permutation operation in our scheme. The new permutation operation can be computed using the old permutation operation: $Per_{\text{new}}(A, B) = Per_{old}(A \oplus B, B)$. We can find this new permutation without the above two weaknesses. The new permutation is a more secure operation, thus this operation is exploited in this paper. Obviously for a given result *Per*(*A, B*), when *B* is determined, there must be one and only one *A*. Since *B* has 2^n values, for a given result *Per*(*A, B*), there are 2^n combinations of *B* and *A*. If the adversary only gets *Per*(*A, B*), it is difficult to get the real *A* and *B*.

However, we still cannot directly exploit the new permutation operation encryption, because the third can be easily calculated when two of *Per*(*A, B*), *A* and *B* are known. In this paper, we also exploit the rotation operation, *Rot*(*A, B*).*Rot*(*A, B*) means that *A* is left rotated *wt*(*B*) bits, where *wt*(*B*) is the Hamming weight of *B*. We hope $0 < < wt(B) < < n$.

3.3. Storage structure

In this scheme, the storage structure of database is shown in Table 2. Each data may have multiple indexes which are followed by a mark bit '0′ or '1′. This mark bit is exploited to record which index is more likely to synchronize with the tag. If this authentication of a tag is successfully completed, the unsynchronized indexes whose mark bit is '0′ will be cleared. Hence, in this case, the data has only one index.

3.4. Timestamp-permutation

Our proposed lightweight RFID authentication protocol is showed in Fig. 2(referred to as timestamp-permutation in this paper). The scheme is divided into two stages as following:

Initialization stage:
It is assumed that the initialization phase is conducted in a secure environment.

(1) The system assigns a unique identity ID to each tag. Each tag stores the identity *IDi*, its encryption value *E*1(*IDi*), and timestamp *Tt*.
(2) Readers have the encryption and decryption keys of *E*1(), *D*1(), *E*2(), *D*2(). Therefore, readers can calculate these two symmetric encryption algorithms.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

(3) The database on the cloud stores the encrypted value of each tag's identity and the corresponding timestamps which are followed by a bit '0′ or '1′. This mark bit is exploited to record which timestamp is more likely to be synchronized with the tag. The cloud only has the encryption and decryption key for the second symmetric encryption algorithm. In other words, it cannot decrypt $E1(IDi)$.

Authentication phase:

(1) The reader generates a timestamp Rt and sends Rt to tag.
(2) After receiving Rt, the tag computes: $\Re 1 = Rot(E1(IDi), E1(IDi) \oplus Tt)$, $Per(\Re 1,\ E1(IDi) \oplus Rt)$ and sends the messages $Per(\Re 1,\ E1(IDi) \oplus Rt), M(Tt), Rt$ to the reader. Then, the reader forwards the messages to the cloud.
(3) The cloud looks for timestamps Tt in the database that matches $M(Tt)$. Then, the cloud looks for $E1(IDi)$ that matches $Per(\Re 1,\ E1(IDi) \oplus Rt)$ in the result of the first search. If $E1(IDi)$ exists, the cloud proceeds further as follows:

Step1: If the mark bit of Tt is '1′, the timestamp marked '0′ will be replaced by Rt.
If the mark bit of Tt is '0′, the last certification may not end normally. Rt will be stored and the previous timestamps will not be deleted.
Step2: The cloud calculates $E2(E1(IDi)||Tt||Rt)$ and sends it to the reader.

(4) The reader decrypts $E2(E1(IDi)||Tt||Rt)$ to get $\{E1(IDi), Tt, Rt\}$.If $\{E1(IDi), Tt, Rt\}$ and $Per(\Re 1,\ E1(IDi) \oplus Rt)$ are matched, the reader authenticates the cloud. Next, the reader decrypts $E1(IDi)$ to get IDi and computes: $\Re 2 = Rot(IDi, IDi \oplus Tt)$,$Per(\Re 2,\ IDi \oplus Rt)$. The left half of the result $(Per(\Re 2,\ IDi \oplus Rt))_L$ is sent to the tag.
(5) After receiving $(Per(\Re 2,\ IDi \oplus Rt))_L$, the tag authenticates the reader if $(Per(\Re 2,\ IDi \oplus Rt))_L$ checks out. Then, the tag replaces timestamp Tt with Rt and sends $(Per(\Re 2,\ IDi \oplus Rt))_R$ to the reader.
(6) The reader authenticates the tag and sends $E2(E1(IDi)||Rt)$ to cloud if $(Per(\Re 2,\ IDi \oplus Rt))_R$ is correct.
(7) On receiving $E2(E1(IDi)||Rt)$, the cloud authenticates reader and updates database as follows:

Step1: Change the mark bit of timestamp Rt to 1.
Step2: Delete the timestamps except Rt.

3.5. The changes

We have already proposed an earlier version in [5] (referred to as timestamp-hash in this paper). There are two major changes to timestamp-permutation compared to the timestamp-hash.

(1) In the communication channel between tag and reader, we replaced the hash operation with a combination of permutation and rotation. The simulation of permutation and rotation will be given in Section 4. It shows that both permutation and rotation are lightweight operation. Compared to the middleweight hash operation, the computational overhead is much smaller.
(2) In the timestamp-hash protocol, the cloud sends all tags' information $E(ID||B)$ that match $M(Tt)$ to the reader, which consumes a lot of communication overhead. In the timestamp-permutation, the cloud server can accurately find the information $E1(IDi)$ of the tag, so only one piece of $E1(IDi)$ needs to be transmitted. Moreover, the reader only needs to decrypt the information of a tag.

4. Analysis and evaluation

In this section, we show the security analysis and performance simulation of the proposed protocol.

4.1. Security analysis specifications

In this section, we analyze the security of the proposed protocol timestamp-permutation in six aspects: data confidentiality, mutual authentication, anti-desynchronization, anti-replay, backward security and anti-tracking.

4.1.1. Data confidentiality

In the front-end channel between tag and reader, the tag's secret data are encrypted using permutation and rotation before being transmitted. The attacker is difficult to get IDi from $Per(Rot(E1(IDi),\ E1(IDi) \oplus Tt), E1(IDi) \oplus Rt)$ or $Per(Rot(IDi,\ IDi \oplus Tt), IDi \oplus Rt)$. In the back-end channel between reader and cloud, the tag's secret data are encrypted using symmetric encryption algorithm. Therefore, this attack is hard to obtain confidential information from the channels. The cloud database has only $E1(IDi)$, which avoids the risk of cloud leakage IDi.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

Table 3
Security performance comparison.

Properties	timestamp-hash [5]	Sarah Protocol [1]	Xie Protocol [23]	Xiao Protocol [8]	timestamp-permutation
Privacy-preserving	√	Δ	√	√	√
Mutual authentication	√	Δ	Δ	Δ	√
Anti-desynchronization	√	√	×	√	√
Anti-replay	√	√	√	√	√
Backward security	√	×	√	√	√
Anti-tracking	√	×	√	√	√

(×: not satisfied; Δ: partially satisfied; √: satisfied).

4.1.2. Mutual authentication

In Authentication Phase (5), the tag authenticates the reader by comparing the left half of $Per(\Re 2, IDi \oplus Rt)$ with $(Per(\Re 2, IDi \oplus Rt))_L$. Similarly, the reader authenticates the tag in Authentication Phase (6). In Authentication Phase (4), the reader decrypts $E2(E1(IDi)||Tt||Rt)$ to get $\{E1(IDi), Tt, Rt\}$. The reader does not authenticate the cloud unless $\{E1(IDi), Tt, Rt\}$ and $Per(\Re 1, E1(IDi) \oplus Rt)$ match. In Authentication Phase (7), cloud verifies $\{E1(IDi), Rt\}$ sent by the reader to authenticate the reader. In short, the protocol enables mutual authentication between tags and readers, and between the reader and the cloud.

4.1.3. Anti-desynchronization

In Step1 of Authentication Phase (3), the cloud stores the new timestamp *Rt* regardless of whether the subsequent authentication process goes on. At this point, the old timestamp *Tt* is also stored in the cloud database. This feature avoids the cloud is not synchronized with the tag. In Authentication Phase (7), the cloud is convinced that the tag has updated timestamp. Therefore, the cloud can delete timestamps that are not synchronized.

4.1.4. Replay attack resistance

All messages sent in the protocol are affected by the timestamp *Rt*. *Rt* is generated by the reader and is different for each authentication. Thus, each time the information is different and unpredictable. The attacker replaying the last round messages of tag or cloud is unable to be authenticated by system. If the previous round of authentication is successful, the timestamps for both the tag and the cloud will be updated. Therefore, that the attacker replays the last reader's message is unable to be authenticated.

4.1.5. Backward security

In this proposed protocol, tag's *ID* has never been transmitted in plaintext.*ID* is encrypted by permutation or symmetric encryption algorithm. Therefore, it is difficult for an attacker to get *ID* from the messages sent before. The timestamp *Rt* sent by the reader is hard to predict. Thus, the attacker cannot predict the content of the authentication information in the future session.

4.1.6. Anti-tracking

The information that the tag sends except the *Tt* will change with the change of the *Rt*, where the sent *Tt* is blurred by $M(\,)$. Notice that different timestamps *T* may correspond to the same $M(T)$. An attacker cannot distinguish tags by $M(Tt)$. However, if an attacker sends the same *Rt* to a tag each time, $Per(\Re 1, E1(IDi) \oplus Rt)$ can be used to track the tag. To solve this problem, the tag records *Rt* received recently. If the received *Rt* is repeated, the tag does not respond.

4.2. Security comparison

In this section, we compare our protocol with three protocols, Sarah protocol [1], Xie protocol [23] and Xiao protocol [8] in six aspects. It is clear from Table 3 that the security of the timestamp-permutation is the highest. In Sarah protocol [1], $H(IDi)$ can be obtained from the last certification, which makes the protocol without backward security. With this feature, attackers can track the tag. Thus, Sarah protocol [1] cannot completely protect the privacy of the tag. The Xie protocol [23] cannot resist the special synchronization attacks. None of the three protocols implements mutual authentication between the reader and the cloud.

4.3. Performance simulation

This paper uses the hardware description language Verilog HDL to design the tag circuit. The simulation software used is the Vivado released by FPGA vendor Xilinx in 2012. The cryptographic operations used in the tags are like $Per(Rot(A, A \oplus B), A \oplus C)$, whose circuit implementation is shown in Fig. 3. In the simulation, we assume that the input length is 80 bits. From Fig. 4, we can see that permutation and rotation need 952 and 197 leaf cells, respectively. The combination operation requires 1151 leaf cells, which is less than that of the hash function. It takes only three clocks for the tag to run $Per(Rot(A, A \oplus B), A \oplus C)$.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

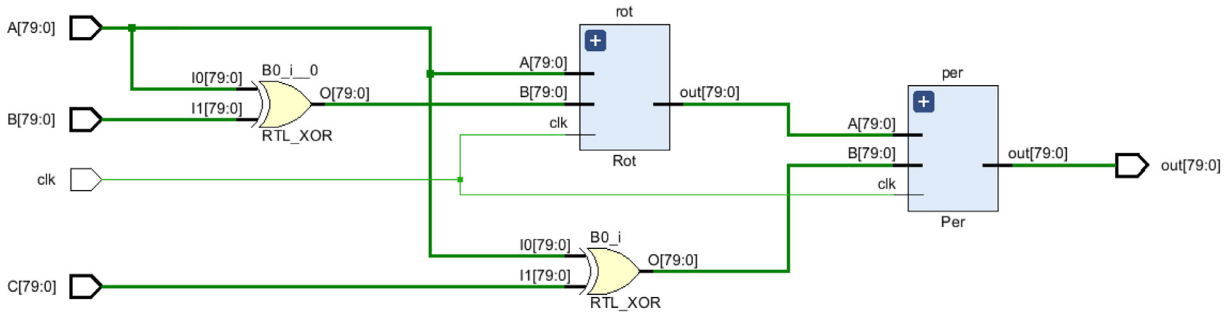

Fig. 3. Simulation circuit.

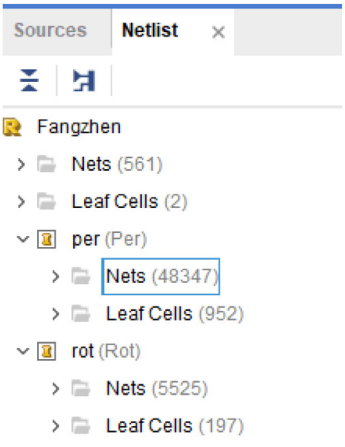

Fig. 4. Resource consumption.

Table 4
Computation and storage overhead of tag.

Protocol	Hash	PRNG	Per+ Rot	Storage
timestamp-hash [5]	2	0	0	*IDt*, *Tt*
Sarah Protocol [1]	6	1	0	*IDi*, *Ki*
Xie Protocol [23]	4	1	0	*T*, *R*, *S*
Xiao Protocol [8]	5	1	0	*Tidi*, *Rid*, *Ki*
timestamp-permutation	0	0	2	*IDt*, *E*(*IDt*), *Tt*

Table 5
Computation overhead of reader.

Protocol	Hash	PRNG	Encryption and Decryption	Per+Rot
timestamp-hash [5]	2	0	m+1	0
Sarah Protocol [1]	6	1	3	0
Xie Protocol [23]	5	1	2	0
Xiao Protocol [8]	4	1	2	0
timestamp-permutation	0	0	3	2

m: the number of tags corresponding to the same *M*(*Tt*).

4.4. Performance analysis and comparison

In the timestamp-permutation, the tag discard hash and PRNG, but uses permutation and rotation. It can be seen from Table 4 that the timestamp-permutation uses only two times permutation and rotation operation. Since permutation and rotation are less computationally expensive than hash and PRNG, the tag computational overhead in this protocol is minimal. The reader also does not use hash. Regarding storage overhead, the tag in our scheme needs to store three data, which is close to other schemes. In Table 5, *m* represents the number of tags corresponding to the same *M*(*Tt*) in the timestamp-hash. When the number of tags in the system is large, *m* may be relatively large. However, this problem does not exist in the timestamp-permutation. As we can see from Table 5, the reader computational overhead of this protocol is satisfactory.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

Because the computing power of the cloud server is strong enough, we do not consider the impact of cloud computing overhead on the system.

5. Ban logical proof

At the same time of protocol design, it needs a set of protocol analysis methods to analyze the protocol and guide the design of the protocol. Formal analysis of security protocols is a formal, standard way to analyze the protocol to check if the protocol meets its security goals [7,12,15]. Currently, there are four main types of formal analysis protocols for security protocols [3,14]:

(1) Logical method: The formal analysis method based on belief and knowledge logic is the most widely used one. The best known is BAN logic.
(2) Common formal analysis: This method uses some common formal analysis methods to analyze security protocols, such as the application of Petri net.
(3) Model detection method: This is a finite state system model based on algebraic method to construct a running protocol, and then use the state detection tool to analyze the security protocol.
(4) Theorem proof method: The security of cryptographic protocol is proved as a theorem, which is a new research hotspot. As the corroboration of the proof procedure, the agreement proofs are reduced to proving some cyclic invariants.

In this section, we use BAN logical protocol proof methods to verify the improved protocol. Before the proof, we introduce the BAN logic symbols and rules used in this paper. P and Q are entities. X is a message. K is an encryption key.

Symbols:

(1) $P|\equiv X$: P believes that X is true.
(2) $P\triangleleft X$: P received a message containing X, that is, a network agent Q sent a message containing X to P.
(3) $P|\sim X$: P has sent a message containing X.
(4) $\#(X)$: X is fresh, that is, X was not sent before the current round.
(5) $P \overset{K}{\longleftrightarrow} Q$: K is the shared encryption key between P and Q. Other network agents except P and Q do not know K.
(6) $P \overset{X}{\leftrightarrows} Q$: X is the shared secret of P and Q. Other network agents except P and Q do not know X.
(7) $\{X\}_K$: X is encrypted with key K.
(8) $A \vdash B$: Formula A can derive the Formula B.

Rules:

(1) Message-meaning rules $\frac{P|\equiv P \overset{Y}{\leftrightarrows} Q,\ P\triangleleft\{X\}_Y}{P|\equiv Q|\sim X}$: If P believes that Y is the shared secret between P and Q, and P receives the message $\{X\}_K$, then P believes that Q has sent message X.
(2) Freshness rule $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$: If a part of the message X is fresh, the message (X, Y) is fresh.

The BAN logical proof process is divided into five parts:

A. Protocol description

The message transmitted between entities in the protocol is described. In the following formula, T represents the tag; R represents the reader; and C represents the cloud server.

① R → T: $\{Query,\ Rt\}$
② T → R: $\{Per(\ Rot(E1(IDi),\ E1(IDi)\oplus Tt), E1(IDi)\oplus Rt), M(Tt), Rt\}$
③ R → C: $\{Per(\ Rot(E1(IDi),\ E1(IDi)\oplus Tt), E1(IDi)\oplus Rt), M(Tt), Rt\}$
④ C → R: $\{E2(E1(IDi)||Tt||Rt)\}$
⑤ R → T: $\{Per(\ Rot(IDi,\ IDi\oplus Tt), IDi\oplus Rt)_L\}$
⑥ T → R: $\{Per(\ Rot(IDi,\ IDi\oplus Tt), IDi\oplus Rt)_R\}$
⑦ R → C: $\{E2(E1(IDi)||Rt)\}$

B. Protocol idealization

Rewrite the message in the protocol to a formula that conforms to the BAN logic syntax. Eliminate messages or entities that have no impact on security goals.

① R → T: $T\triangleleft\{Rt\}$
② T → R: $R\triangleleft\{Per(\ Rot(E1(IDi),\ E1(IDi)\oplus Tt), E1(IDi)\oplus Rt)\}$
③ R → C: $C\triangleleft\{Per(\ Rot(E1(IDi),\ E1(IDi)\oplus Tt), E1(IDi)\oplus Rt)\}$
④ C → R: $R\triangleleft\{\{\{IDi\}_{E1}, Tt,\ Rt\}_{E2}\}$
⑤ R → T: $T\triangleleft\{Per(\ Rot(IDi,\ IDi\oplus Tt), IDi\oplus Rt)_L\}$

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

⑥ $T \rightarrow R: R \triangleleft \{Per(\ Rot(IDi,\ IDi \oplus Tt), IDi \oplus Rt)_R\}$
⑦ $R \rightarrow C: C \triangleleft \{\ \{\ \{IDi\}_{E1}, Tt\ \}_{E2}\}$

C. Initial assumptions

Based on the scenario in which the protocol is located and the capabilities of each entity, the following rationalization assumptions are derived:

I. $T| \equiv T \overset{Tt}{\leftrightarrows} R$
II. $R| \equiv R \overset{E2}{\longleftrightarrow} C$
III. $R| \equiv \#(Rt)$
IV. $R| \equiv T \overset{\mathrm{Tt}}{\rightleftarrows} R$
V. $C| \equiv C \overset{E2}{\longleftrightarrow} R$

D. Proving goals

According to the functions of the protocol, the security goals that the protocol should meet are as follows:

a. $T| \equiv R| \sim IDi$
b. $R| \equiv C| \sim E1(IDi)$
c. $R| \equiv T| \sim IDi$
d. $C| \equiv R| \sim E1(IDi)$

E. Proof process

From BAN logic message-meaning rules $\frac{P| \equiv P \overset{Y}{\rightleftarrows} Q,\ P \triangleleft \{X\}_Y}{P| \equiv Q| \sim X}$, Protocol Idealization ⑤ and Initial Assumptions I. We can get:

$T \triangleleft \{Per(\ Rot(IDi,\ IDi \oplus Tt), IDi \oplus Rt)_L\}, T| \equiv T \overset{\mathrm{Tt}}{\rightleftarrows} R \vdash\ T| \equiv R| \sim IDi$. Proving Goals a. is proved.

From freshness rule $\frac{P| \equiv \#(X)}{P| \equiv \#(X,Y)}$, Initial Assumptions III, we can get:

$$R| \equiv \#(Rt) \vdash\ R| \equiv \#(\{\ \{IDi\}_{E1},\ Tt, Rt\ \}_{E2})$$

From BAN logic message-meaning rules $\frac{P| \equiv P \overset{K}{\longleftrightarrow} Q,\ P \triangleleft \{X\}_K}{P| \equiv Q| \sim X}$, Protocol Idealization ④ and Initial Assumptions II., we can get:

$R| \equiv R \overset{E2}{\longleftrightarrow} C, R \triangleleft \{\ \{\ \{IDi\}_{E1},\ Tt, Rt\ \}_{E2}\} \vdash\ R| \equiv C| \sim E1(IDi)$. Proving Goals b. is proved.

From freshness rule $\frac{P| \equiv \#(X)}{P| \equiv \#(X,Y)}$, Initial Assumptions III, we can get:

$$R| \equiv \#(Rt) \vdash R| \equiv \#(Per(\ Rot(IDi, IDi \oplus Tt),\ IDi \oplus Rt)_R)$$

From BAN logic message-meaning rules $\frac{P| \equiv P \overset{Y}{\rightleftarrows} Q,\ P \triangleleft \{X\}_Y}{P| \equiv Q| \sim X}$, Protocol Idealization ⑥ and Initial Assumptions IV., we can get:

$R| \equiv T \overset{\mathrm{Tt}}{\rightleftarrows} R, R \triangleleft \{Per(\ Rot(IDi, IDi \oplus Tt),\ IDi \oplus Rt)_R\} \vdash\ R| \equiv T| \sim IDi$. Proving Goals c. is proved.

From BAN logic message-meaning rules $\frac{P| \equiv P \overset{K}{\longleftrightarrow} Q,\ P \triangleleft \{X\}_K}{P| \equiv Q| \sim X}$, Protocol Idealization ⑦ and Initial Assumptions V., we can get:

$C| \equiv C \overset{E2}{\longleftrightarrow} R, C \triangleleft \{\ \{\ \{IDi\}_{E1},\ Tt\ \}_{E2}\} \vdash C| \equiv R| \sim E1(IDi)$. Proving Goals d. is proved.

The results show that our protocol ensures mutual authentication between tags and clouds, as well as mutual authentication between readers and clouds.

6. Conclusion

There are two architectures in the traditional RFID system: a back-end server-based architecture and a server-less offline authentication architecture. The expansibility of these two architectures is poor, and the capability of data storage and query is not satisfactory. This paper proposes a cloud-based authentication scheme, which takes advantages of the strengths of cloud computing. Aiming at the limitations of tag storage capacity and computing power in RFID systems, this scheme encrypts authentication data by permutation and update authentication data by timestamp. Security analysis is given to show the security properties of the timestamp-permutation protocol. The protocol can protect the privacy of the tags in a semi-trusted cloud environment. Performance evaluation validates that the timestamp-permutation protocol is efficient with a reduced storage and communication overhead, and controls the growth of the computing overhead of the tag. In summary, the timestamp-permutation protocol provides a higher level of security, and is suitable for low cost RFID systems.

Declaration of interest statement

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006

Acknowledgment

This work was funded by the National Key R&D Program of China (no. 2017YFB0802300), the National Natural Science Foundation of China (no. 61772403 and no. U1401251), the Fundamental Research Funds for the Central Universities, and National 111 Program of China B16037 and B08038.

An earlier version of this paper was presented at the DSC 2017 Conference and was published in its Proceedings, DOI: 10.1109/DSC.2017.41.

Reference

[1] S. Abughazalah, K. Markantonakis, K. Mayes, Secure improved cloud-based RFID authentication protocol, in: Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, 2015, pp. 47–164.
[2] Y. Chun, Y. Hwang, H. Lee, RFID tag search protocol preserving privacy of mobile reader holders, IEICE Electron. Express 8 (2) (2011) 50–56.
[3] H. Fan, D. Feng, Security Protocol Theory and Methods, Science Press, 2003.
[4] 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.
[5] K. Fan, Q. Lu, H. Li, et al., Cloud-Based lightweight RFID mutual authentication protocol, in: IEEE Second International Conference on Data Science in Cyberspace, IEEE, 2017, pp. 333–338.
[6] K. Fan, W. Wang, W. Jiang, Secure ultra-lightweight RFID mutual authentication protocol based on transparent computing for IoV,, Peer-to-Peer Netw. Appl. (2017) 1–12.
[7] D. Feng, Research on theory and approach of provable security, J. Softw. (2005) 1743–1756.
[8] X. Hannan, A. Awatif, C. Bruce, A Cloud-based RFID Authentication Protocol with Insecure Communication Channels, in: IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 332–339.
[9] E. Hoque, F. Rahman, I. Ahamed, Enhancing privacy and security of RFID system with serverless authentication and search protocols in pervasive environments, Wirel. Pers. Commun. 55 (1) (2010) 65–79.
[10] F. Lee, Y. Chien, S. Laih, Server-less RFID authentication and searching protocol with enhanced security, Int. J. Commun. Syst. 25 (3) (2012) 376–385.
[11] W. Liu, Research on cloud computing security problem and strategy, in: CECNet, 2012, pp. 1216–1219.
[12] D. Moinar, D. Wagner, “Privacy and security in library RFID: issues, practices, and architectures, in: 11th ACM Conference on Computer and Communication Security, 2004, pp. 210–219.
[13] M. Ohkubo, K. Suzuki, S. Kinoshita, Hash-chain based forward-secure privacy protection scheme for low-cost RFID, Symp. Cryptogr. Inf. Secur. (2004) 719–724 (SCIS2004).
[14] K. Qin, X. Zhang, Y. Hao, Network Security Protocol, University of Electronic Science and Technology Press, 2008.
[15] S. Qing, Design and logical analysis of security protocols, J. Softw. (2003) 1300–1309.
[16] W. Roy, Enabling ubiquitous sensing with RFID, Computer 37 (4) (2004) 84–86.
[17] E. Sarma, A. Weis, W. Engels, Radio-frequency identification: secure risks and challenges, RSA Lab. Cryptobytes 6 (1) (2003) 2–9.
[18] H. Shi, X. Bai, C. Ren, C. Zhao, Development of internet of vehicle’s information system based on cloud, J. Softw. 9 (7) (2014) 15–21.
[19] B. Surekha, L. Narayana, A realistic lightweight authentication protocol for securing cloud based RFID system, in: IEEE International Conference on Cloud Computing in Emerging Markets IEEE, 2017, pp. 54–60.
[20] A. Thange, RFID authentication protocol for security and privacy maintenance in cloud based employee management system, Int. J. Eng. Res. Gen. Sci. (2014) 446–453.
[21] Y. Tian, G. Chen, J. Li, A new ultralightweight RFID authentication protocol with permutation, IEEE Commun. Lett. (2012) 702–705.
[22] Z. Wei, Y. Zhang, J. Yang, Private assets protection system based on RFID and cloud computing, in: CECNet, 2013, pp. 196–198.
[23] W. Xie, L. Xie, C. Zhang, Q. Zhang, C. Tang, Cloud-based RFID authentication, in: IEEE International Conference on RFID (RFID ’13), 2013, pp. 168–175.
[24] B. Zhang, X. Ma, G. Qin, Design and analysis of a lightweight mutual authentication protocol for RFID, J. Univ. Electron. Sci. Technol. China (01) (2012) 425–430.
[25] Z. Zhang, M. Wang, RFID security privacy problem and strategy analysis, Microcomput. Inf. 24 (6–3) (2008) 47–49.

Kai Fan received his B.S., M.S. and Ph.D. degrees from Xidian University, P. R. China, in 2002, 2005 and 2007, respectively, in Telecommunication Engineering, Cryptography and Telecommunication and Information System. He is working as a professor in State Key Laboratory of Integrated Service Networks at Xidian University. He has published over 70 papers in journals and conferences. His research interest includes information security. The mailing address is 2 South Taibai Road, Xidian University, Xian 710071, China.

Qi Luo was born in 1993 in Hubei province of China. He received his B. S. degree in information security from Xidian University in 2016. He is studying as a master in State Key Laboratory of Integrated Service Networks at Xidian University. His research interest is IoT security.

Kuan Zhang received his B.S. and M.S. degrees from Northeastern University, P. R. China, in 2009 and 2011, respectively, in Communication Engineering and Computer Applied Technology. He received his Ph.D. degree from University of Waterloo, Canada, in 2016, in Electrical and Computer Engineering. He was a Postdoctoral Fellow from 2016 to 2017 at the University of Waterloo, Canada. He is working as an assistant professor in Department of Electrical and Computer Engineering at University of Nebraska–Lincoln, USA. He has published over 50 papers in journals and conferences. He was the recipient of Best Paper Award in IEEE WCNC 2013 and Securecomm 2016. His research interests include cyber security, big data, cloud/edge computing. The mailing address is PKI 206D, Scott Campus (Omaha), University of Nebraska–Lincoln, 1400 R Street Lincoln, NE 68588, USA.

Yintang Yang was born in 1962 in Hebei Province of China. He received his Ph.D. degree in semiconductor from Xidian University. He is now a professor at Key Lab. of Minist. of Educ. for Wide Band-Gap Semicon. Materials and Devices of Xidian University, Xi'an China. His research interests include semiconductor materials and devices, network and information security.

Please cite this article as: K. Fan, Q. Luo and K. Zhang et al., Cloud-based lightweight secure RFID mutual authentication protocol in IoT, Information Sciences, https://doi.org/10.1016/j.ins.2019.08.006