J Ambient Intell Human Comput
DOI 10.1007/s12652-017-0485-5

ORIGINAL RESEARCH

A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications

Fan Wu[1] · Lili Xu[2] · Saru Kumari[3] · Xiong Li[4] · Ashok Kumar Das[5] · Jian Shen[6]

Received: 6 December 2016 / Accepted: 30 March 2017

Abstract As an important part of Internet of Things, Radio Frequency Identification (RFID) system employs low-cost RFID tag to communicate with everything containing animate and inanimate objects. This technology is widely used in the e-healthcare applications. However, the malicious communication environment makes people more and more worried. In order to overcome the hazards in the network, RFID authentication schemes for e-healthcare have been proposed by researchers. But since the computation ability of the tag is relatively weak, it is necessary to put forward a lightweight and secure scheme for medical systems. Moreover, cloud is widely accepted by people and used in many kinds of systems. So we propose a novel and lightweight RFID authentication scheme with cloud for e-healthcare applications. We use an enhanced formal security model to prove the security of our scheme. In this model the channel between the server and the reader is considered to be insecure and informal analysis is used to prove the security of the proposed scheme. Through the formal and informal analysis, our scheme not only resists the common attacks, but also keeps mutual authentication, information integrity, forward untraceability and backward untraceability. Moreover, both the tag and the reader can reach the anonymity. Our scheme is only hash-based and suitable to realize various security requirements. Compared to recent schemes of the same sort, it is more applicable in e-healthcare.

Keywords Forward untraceability · Backward untraceability · Radio frequency identification · Mutual authentication · Formal proof · e-healthcare application

1 Introduction

Internet of Things (IoT) is a popular notion, which means that existing objects in the network are embedded with tiny electronic devices such as sensors and tags. These devices are used for collecting data from objects and people can master the information of every object in realtime. Based on IoT, the Internet is expanded by wireless networks covering every corner of the world, like the wireless sensor networks (Zhang et al. 2016; Wu et al. 2016). Moreover, radio frequency identification (RFID) is an important technology of IoT since it is a way to deal with the unique identified objects.

Nowadays, RFID-integrated e-healthcare systems are developed very fast. According to Wamba et al. (2013), there are three kinds of RFID-based applications for e-healthcare: patient management, asset management, and staff management. One example is that patients or assets can be guarded in a smart space equipped with

✉ Fan Wu
conjurer1981@gmail.com

1 Department of Computer Science and Engineering, Xiamen Institute of Technology, Xiamen 361021, China

2 School of Information Science and Technology, Xiamen University, Xiamen 361005, China

3 Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250005, India

4 School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

5 Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India

6 School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Published online: 12 April 2017

many RFID readers (Amendola et al. 2014). Via near field communication or bluetooth, the readers interact with small passive tags which are distributed in the whole space. Another issue is that the correctness of billing and diminishing medication error level are important in healthcare service. Above all, three styles can be listed. The first is the ambient RFID tag, which is installed at a concrete position, e.g., on the wall, and indexes like humidity, temperature, or concentration of toxic agent can be detected. The realtime information such as variations of motions, blood pressure from a patient, etc., can be collected by wearable RFID tags on skin and the implanted RFID tags under skin, which are the second and third style, respectively. Via the Internet, data are transmitted to a database, from which the doctors and nurses can obtain patient related information, or relative staffs can get the data of medical materials in order to prevent the loss of assets. Here each doctor, nurse and other worker must have a legal role to access the corresponding information after passing the identification check with RFID-based card. Cloud storage is an urgent technology which is broadly employed by many systems. It can be seemed as storing online data in the cloud. In fact, the storage resources used and accessed are located in different places. From the view of transparent definition, all the resources can be seen as a cloud. There are many advantages for cloud server, such as low storage fee and protection of backup, good reliability and accessibility, and secure storage and access technologies are deeply studied (Fu et al. 2016a; Xia et al. 2016; Fu et al. 2016b; He et al. 2015; Fu et al. 2016c; Zhang et al. 2016; Fu et al. 2016d; Jiang et al. 2016; Fu et al. 2015). However, due to the vulnerable nature of communication environment, security and privacy problems attract researchers (Ma et al. 2015; Yuan et al. 2016; Xia et al. 2016; Fu et al. 2016c; Wu et al. 2016). A secure and reliable RFID authentication scheme assisted by the cloud is helpful for the e-healthcare system. Recently, researchers have put their focus on two main issues: anonymous and mutual authentication. For the first aspect, tag anonymity is noticed in nearly all schemes, but to keep the readers anonymous is a new method, which has not been popular until now. According to (Chen et al. 2013), the channel between the reader and the cloud server can be wired or wireless. Generally, it is not fit to consider such long channel as a secure one. This view was proposed in Niu et al. (2014). To protect the privacy of the reader and the tag, anonymity is necessary. For the second, since forgery attacks may happen in the authentication process, mutual authentication is very important to guarantee the correct message sources and destinations.

1.1 Network model

Generally speaking, the RFID-based system includes three sorts of elements: RFID tags, RFID readers and the cloud server which maintains a database storing information about the above two kinds of entities. There are three kinds of RFID tags: active, semi-active and passive. An active tag has a battery on it and sends its identification signal periodically. A semi-active tag also has a smaller battery and does nothing until it receives an activating signal. A passive tag has no battery and needs the energy sent by the reader. The first two are only for high-value goods and more costly than passive tags (rfidjournal.com 2013). In fact, only passive tags are commonly used in e-healthcare systems. So the general process of identification for the passive tags is executed as follows: the RFID reader sends a challenging message to the tag. The tag sends back a response after some computations with its information. Then the information is transmitted to the cloud server for authentication. We illustrate the probable structure of RFID-based e-healthcare systems in Fig. 1. As we have mentioned above, data of persons as well as of assets can all be gained from the RFID-based systems. Moreover, as we have mentioned above, the channels between the participants are insecure, no matter wired or wireless.

1.2 Related work

In recent years, RFID authentication is a hot topic on which many schemes have been proposed (Huang et al. 2014; Benssalah et al. 2014; Chen and Chou 2015; Ryu et al. 2015; Li et al. 2015b; Akgün et al. 2015; He et al. 2014; Kumar et al. 2015; Jin et al. 2016; Fernando and Abawajy 2011; Zhu et al. 2011; Cho et al. 2011; Niu et al. 2014; Wang et al. 2015; Srivastava et al. 2015; Li et al. 2015a; Gope and Hwang 2015; Safkhani et al. 2014).

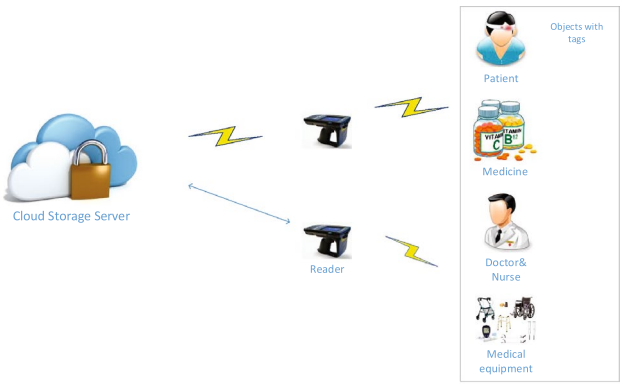

Fig. 1 Structure of RFID-based e-healthcare systems

Yang et al. (2005) presented an RFID authentication scheme based only on hash functions. But the paper Piramuthu (2011) pointed out that Yang et al.'s scheme had weaknesses like traceability and vulnerability to forgery attack. Yu et al. (2012) presented a binging proof protocol for RFID tags with low cost. But in Wu et al. (2012) found that the impersonation attack could be done on Yu et al. (2012) by attackers. Cho et al. (2011) proposed a hash-based RFID authentication scheme, which was claimed to overcome the common attacks. But Safkhani et al. (2014) showed that Cho et al.'s scheme was under the de-synchronization attack, the tag forgery attack and the reader forgery attack. Srivastava et al. (2015) proposed a hash-based RFID tag authentication scheme for telecare medicine information systems (TMISs), with shared secret keys between the tag and the server. Also Li et al. (2015a) pointed out that Srivastava et al.'s protocol was vulnerable to the reader lost attack and destitute of mutual authentication and scalability. Also, they presented an improved one for TMISs. But in the same year, Ghaemmaghami et al. (2015) showed that weaknesses like the de-synchronization attack and reader traceability affect in the scheme of Li et al. (2015a). However, since passive tags abiding by Electronic Product Code (EPC) Class-1 Generation-2 standard (we will use Gen-2 in the latter part) are generally employed due to their low price, the schemes (Li et al. 2015a; Ghaemmaghami et al. 2015; Srivastava et al. 2015) are not suitable. The reason is that timestamp is used in the three schemes, but in Gen-2, timestamp is not supported. Besides the schemes mentioned above, researchers also proposed some more schemes (Kumar et al. 2015; Jin et al. 2016) for e-healthcare circumstance based on elliptic curve cryptography. Due to the heavy computation burden of the scalar multiplication process, it is unsuitable to use public-key infrastructure on the weak-powered tags. So hash-based authentication schemes are popular due to their lightweight computation cost.

Vaudenay (2007) gave an RFID privacy model. Until now, this model is used the most widely. Many researchers have discussed it and presented their supplements or improvements, like mutual authentication (Paise and Vaudenay 2008), time-measuring enhancement (Avoine et al. 2010), traceability for schemes (Ng et al. 2009), etc. Nearly all of them concentrate on the model including two elements in the system: one is the tag, and the other is the reader/server with a database. In Li et al. (2014), all the three elements of the system are considered, but the channel between the reader and the cloud server is still supposed to be secure. Researchers consider the security of tag, which is the weakest in all of the three entities and the insecure channel which is directly linked to the tag. Since there may be a complicated environment between the reader and the cloud server, as referred in Niu et al. (2014), it is an urgent task to propose a model where an insecure channel connects the reader and the cloud server.

1.3 Our contributions

The main contributions of the paper are listed below:

1. A new hash-based RFID mutual authentication scheme which keeps anonymity for both the tag and the reader is proposed.
2. We employ an enhanced formal model to demonstrate the formal proof. And through the proof, our scheme reaches the Timeful-Weak privacy level, which is the best for symmetric secret key synchronization schemes (Ng et al. 2009). It also owns correctness and security. Finally, it keeps forward untraceability and backward untraceability.
3. Through informal analysis we show that our scheme also meets the common security requirements and is applicable in practice.

1.4 Organization

The rest of the paper is arranged as follows. The preliminary knowledge is provided in Sect. 2. Our scheme, the formal proof and informal analysis are provided in Sects. 3, 4 and 5, respectively. Finally, the conclusion is in Sect. 6.

2 Preliminaries

2.1 Notations

We list the notations used throughout the paper in Table 1.

2.2 Security requirements of RFID systems for informal analysis

Here we consider the communication between the tag, the reader and the server. The channels between them are assumed to be public for the process of authentication. According to Niu et al. (2014), Srivastava et al. (2015), Gope and Hwang (2015), Li et al. (2015a), the security requirements for informal analysis are illustrated below:

1. Mutual authentication: because three entities are used in the scheme, each of them should be authenticated by the others.
2. Anonymity: a tag's identity should not only be hidden in the transmission but also be not tracked. It is better that if a reader's identity cannot be tracked, either.

Table 1 Notations

Symbols	Meaning
$\mathscr{T}_i$	The i-th tag
$\mathscr{R}_j$	The j-th reader
ID_i, PID_i	The i-th identity and pseudo-identity of the tag $\mathscr{T}_i$
$RID_j, PRID_j$	The j-th identity and pseudo-identity of the reader $\mathscr{R}_j$
S	The cloud server
R_r, R_t	Random numbers generated by the reader and the tag, respectively
$Data_i$	Information of the tag, which is stored in S
$h(.)$	Collision resistant one-way hash function
$\|\|$	The concatenation operation
$\oplus$	The X-OR operation
$\mathscr{A}$	An adversary
l	Security parameter used for lengths of hash results and random numbers

3. Scalability: the server should search the tag's identity in its database to authenticate it. But when the number of tags increases, the computation for searching should vary insignificantly. Or we can say that the identity of the tag must be ascertained by directly searching without evolving any extra checking.
4. Resistance to common attacks: attacks such as the replay attack and the de-synchronization attack must be avoided.

2.3 Basic knowledge about hash function

$l \in N$ is a security parameter and a hash function can be defined as $H{:}\{0,1\}^* \rightarrow \{0,1\}^l$ where N is the set of natural numbers. Furthermore, three conditions are needed:

1. For a given y, it is hard to find a x where $H(x) = y$.
2. It is hard to find two different binary strings x_1 and x_2 such that $H(x_1) = H(x_2)$.
3. It is hard for an adversary who queries H in polynomial time to tell apart the outputs of H from the random numbers.

3 Outline of our scheme

We totally divide our scheme into two phases: registration and authentication.

3.1 Registration phase

– Step 1. $\mathscr{T}_i$ and S share ID_i. Then S selects a random string PID_i as the pseudo-identity and a secret key x_i, and stores them in $\mathscr{T}_i$. S defines $x_i^{new} = x_i^{old} = x_i$ and $P_i^{new} = P_i^{old} = PID_i$ and stores $(x_i^{new}, x_i^{old}, P_i^{new}, P_i^{old})$ in database.
– Step 2. $\mathscr{R}_j$ and S share RID_j. Then S selects a random string $PRID_j$ as the pseudo-identity and a secret key x_j, and stores them in $\mathscr{R}_j$. S sets $x_j^{new} = x_j^{old} = x_j$ and $P_j^{new} = P_j^{old} = PRID_j$ and stores $(x_j^{new}, x_j^{old}, P_j^{new}, P_j^{old})$ in its own database. So the records $(ID_i, (x_i^{new}, P_i^{new}), (x_i^{old}, P_i^{old}))$ and $(RID_j, (x_j^{new}, P_2^{new}), (x_j^{old}, P_j^{old}))$ are for $\mathscr{T}_i$ and $\mathscr{R}_j$, respectively.

3.2 Authentication phase

This phase includes six steps. The steps are shown in Fig. 2.

– Step 1. $\mathscr{R}_j$ generates a nonce R_r, and sends the message $M_1 = \{R_r\}$ to $\mathscr{T}_i$.
– Step 2. After $\mathscr{T}_i$ receives M_1, it generates R_t, computes $B_1 = h(x_i||R_r||R_t||ID_i)$ and sends the message $M_2 = \{B_1, R_t, PID_i\}$ to $\mathscr{R}_j$.
– Step 3. $\mathscr{R}_j$ computes $B_2 = h(x_j||R_r||R_t||RID_j)$, and sends the message $M_3 = \{B_1, B_2, R_r, R_t, PID_i, PRID_j\}$ to S.
– Step 4. After receiving M_3 from $\mathscr{R}_j$, S searches PID_i and $PRID_j$ in the database. Either unsuccessful search will lead to the rejection. We divide the following operations to three cases:
  – **Case 1**: If $PID_i = P_i^{new}$ and $PRID_j = P_j^{new}$, S retrieves the corresponding ID_i, x_i^{new}, RID_j and x_j^{new}, and checks if $B_1 = h(x_i^{new}||R_r||R_t||ID_i)$ and $B_2 = h(x_j^{new}||R_r||R_t||RID_j)$. If either of them fails, the session will be rejected. Otherwise, S updates $(x_i^{old}, P_i^{old}, x_j^{old}, P_j^{old})$ with $(x_i^{new}, P_i^{new}, x_j^{new}, P_j^{new})$ in database, and computes the following new data:

$$x_i^{new} = h(x_i^{old}||R_t||R_r||ID_i) \quad (1)$$

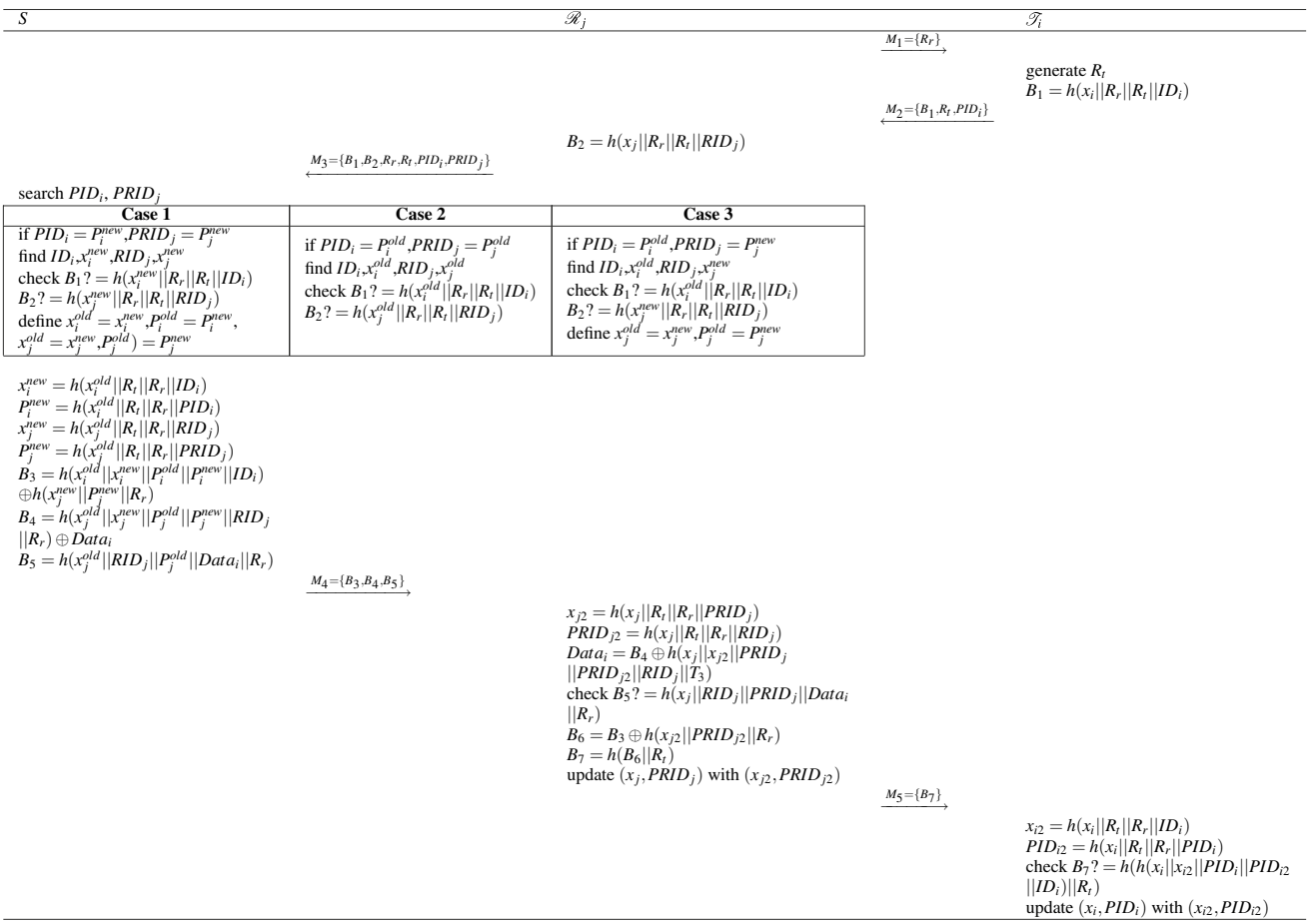

Fig. 2 Authentication phase

$$P_i^{new} = h(x_i^{old}||R_t||R_r||PID_i) \tag{2}$$

$$x_j^{new} = h(x_j^{old}||R_t||R_r||RID_j) \tag{3}$$

$$P_j^{new} = h(x_j^{old}||R_t||R_r||PRID_j) \tag{4}$$

$$B_3 = h(x_i^{old}||x_i^{new}||P_i^{old}||P_i^{new}||ID_i) \oplus h(x_j^{new}||P_j^{new}||R_r) \tag{5}$$

$$B_4 = h(x_j^{old}||x_j^{new}||P_j^{old}||P_j^{new}||RID_j||R_r) \oplus Data_i \tag{6}$$

$$B_5 = h(x_j^{old}||RID_j||P_j^{old}||Data_i||R_r) \tag{7}$$

Finally, S sends the message $M_4 = \{B_3, B_4, B_5\}$ to $\mathscr{R}_j$. The fact that S uses the operations of this case indicates that the last session normally ran.

– **Case 2**: Otherwise, if $PID_i = P_i^{old}$ and $PRID_j = P_j^{old}$, S retrieves the corresponding ID_i, x_i^{old}, RID_j and x_j^{old}, and checks if $B_1 = h(x_i^{old}||R_r||R_t||ID_i)$ and $B_2 = h(x_j^{old}||R_r||R_t||RID_j)$. If either of them fails, the session will be rejected. Otherwise, the elements in Eqs. (1)–(7) are calculated, and S sends the message M_4 to $\mathscr{R}_j$. The fact that S uses the operations of this case indicates that M_4 in the last session is blocked.

– **Case 3**: Otherwise, if $PID_i = P_i^{old}$ and $PRID_j = P_j^{new}$, S retrieves the corresponding ID_i, x_i^{old}, RID_j and x_j^{new}, and checks if $B_1 = h(x_i^{old}||R_r||R_t||ID_i)$ and $B_2 = h(x_j^{new}||R_r||R_t||RID_j)$. If either of them fails, the session will be rejected. Otherwise, S updates (x_j^{old}, P_j^{old}) with (x_j^{new}, P_j^{new}), computes Eqs. (1)–(7), and sends the message M_4 to $\mathscr{R}_j$. The fact that S uses the operations of this case denotes that M_5 in the last session is blocked.

– Step 5. $\mathscr{R}_j$ computes $x_{j2} = h(x_j||R_t||R_r||PRID_j)$ and $PRID_{j2} = h(x_j||R_t||R_r||RID_j)$, gets $\mathscr{T}_i$'s information $Data_i = B_4 \oplus h(x_j||x_{j2}||PRID_j||PRID_{j2}||RID_j||T_3)$, and checks if $B_5 = h(x_j||RID_j||PRID_j||Data_i||R_r)$. If it is true, $\mathscr{R}_j$ computes $B_6 = B_3 \oplus h(x_{j2}||PRID_{j2}||R_r)$ and

$B_7 = h(B_6||R_t)$, updates $(x_j, PRID_j)$ with $(x_{j2}, PRID_{j2})$ and sends the message $M_5 = \{B_7\}$ to $\mathscr{T}_i$.
- Step 6. $\mathscr{T}_i$ calculates $x_{i2} = h(x_i||R_t||R_r||ID_i)$ and $PID_{i2} = h(x_i||R_t||R_r||PID_i)$, and checks if $B_7 = h(h(x_i||x_{i2}||PID_i||PID_{i2}||ID_i)||R_t)$. If it is true, $\mathscr{T}_i$ replaces (x_i, PID_i) with (x_{i2}, PID_{i2}).

4 Formal security analysis

Based on the historical models in Vaudenay (2007), Paise and Vaudenay (2008), Avoine et al. (2010), Ng et al. (2009), Li et al. (2014), we extend the model and consider that the channel between them is also insecure. Then we discuss the security of the proposed scheme in a formal style.

4.1 Security model

4.1.1 System model

An RFID scheme contains procedures as follows:

- $SetupServer(1^l) \rightarrow (K_s, K_p)$: it produces a public parameter K_p and a private parameter K_s, with a security parameter l. Also a database storing information of readers and tags is built.
- $SetupTag^{K_p}(ID) \rightarrow (K_T, S_T)$: it produces a tag owning an identity ID, a key K_T and an updateable state S_T.
- $SetupRead^{K_p}(RID) \rightarrow (K_R, S_R)$: it produces a reader owning an identity RID, a key K_R and an updateable state S_R.
- $Auth \rightarrow out$: the entire protocol is executed. If the tag is authenticated, the identifier of the tag will be output. Otherwise, $\perp$ is the result. Moreover, if the protocol is correctly executed, OK is output as the tag side. Otherwise, $\perp$ is the result.

4.1.2 Adversary model

At first a challenger C executes $SetupServer(1^l)$. The pair (K_s, K_p) is produced and 1^l and K_p are sent to the attacker $\mathscr{A}$. To be convenient, only one *SetupReader* procedure is also used. Since the reader's privacy cannot be divided into several classes. We do not discuss much about the reader but only the possibility of forging its message. Only a server and a reader exist in the system. No tag appears at the beginning. A tag can be distributed in two sets: *free* or *drawn*. A *drawn* tag can be accessed by $\mathscr{A}$ while a *free* tag cannot be visited by the adversary. Other queries such as Result, Corrupt and Timer remain same as those defined in the schemes. So $\mathscr{A}$ can perform the following queries on a simulator according to his class, which is introduced in Sect. 4.1.3:

- $CreateTag^b(ID)$ creates a tag in *free* status with a unique identity ID. The oracle employs *SetupTag* to build a tag. If $b = 1$, the tag is legitimate and stored in the database. We consider $b = 1$ by default.
- $DrawTag(distr)$ randomly moves the tag from the *free* set to the *drawn* set based on the distribution *distr*. If the tag ID is legitimate, a status bit $b = 1$. Otherwise, $b = 0$. Also, a record $(vtag, ID)$ is added into a table $\mathbb{T}$. Finally, the status bit is returned.
- $Free(vtag)$ moves the $vtag$ to the *free* set and the temporary identity $vtag$ is invalid. And the record $(vtag, ID)$ is deleted from $\mathbb{T}$.
- $Launch()$ is to start a new session from the reader and returns the session identifier π.
- $SendReader(m, \pi)$, $SendTag(m, vtag)$ and $SendServer(m, \pi)$ are queries to send the message m to the reader, the tag and the server, respectively. Also, they can be recognized by π or $vtag$ to tell apart from other sessions.
- $Execute(vtag)$ finishes the whole session π between $vtag$, the reader and the server. It starts with $Launch()$ and uses *Send* queries based on the scheme. At last all transcript and π are as the return values.
- $Result(\pi)$ outputs 1 if π is finished, i.e., the legitimate tag is identified. Otherwise it outputs 0. In other words, if all messages in the session π are successively generated by *Send* queries according to the correct order, 1 is output. Or 0 is output.
- $Corrupt(vtag)$ returns the internal secret information of $vtag$.
- $Timer(\pi)$ returns the time cost in the server for session π . It is for $\mathscr{A}$ to tell apart tags related to special protocol instances.

4.1.3 Privacy classes

Privacy classes are divided for discussing tag authentication and security. There are eight classes of attackers mentioned in Vaudenay (2007): (NARROW)-STRONG, (NARROW)-DESTRUCTIVE, (NARROW)-FORWARD and (NARROW)-WEAK. As a WEAK adversary, he cannot query $Corrupt(vtag)$ oracle. As a FORWARD adversary, he can only query $Corrupt(vtag)$ at last. For example, do *Corrupt* on different tags. As a DESTRUCTIVE adversary, he can do other queries after *Corrupt*, but the corrupted $vtag$ is simulated as the destroyed tag, which cannot be used again. As a STRONG adversary, he can query any oracle without limitation. A NARROW adversary cannot use $Result(\pi)$ query. But according to Vaudenay (2007), Ng et al. (2008) and Ng et al. (2009), STRONG level only

appears under the condition of public-key cryptography with asymmetric key setting, and DESTRUCTIVE level is not meaningful, only for analyzing schemes employing correlative secret keys among tags while the schemes do not reach the STRONG level. Also, even if the example in Vaudenay (2007) is classified as Narrow-Destructive privacy, it is considered to be insecure due to under the desynchronization attack and destitution of WEAK privacy. So our aim is to construct a scheme which achieves some privacy level in fact. Our proposed scheme belongs to Type 2a (Ng et al. 2009), where the tag key is updated after server/reader authentication message arrives, and the secret key in server is synchronized by storing both the updated key and the previous key. Such schemes can reach the weak level.

We also employ the time tracking attack in Avoine et al. (2010), which is called TIMEFUL, as another privacy level. The aim of such attackers is to put the emphasis on if the storage position in database can be tracked. Due to Burmester et al. (2008), if the server can find the identity of tag by directly checking the received elements, the time cost is constant. Otherwise, if some computation, generally the hash function, is needed before checking every record in the database, $\mathscr{A}$ may track the tag since the record is stored in a special position of data table and fixed times of mentioned computation are needed for every access. The *Time* oracle is used to tell apart each tag by the time cost on the server side.

4.1.4 Security properties

1. *Correctness*: if the output of Auth process is right, or we can say that the tag identity can be output correctly on the server side and OK can be output on the tag side finally, the RFID authentication scheme is said to be correct.
2. *Security*: tag authentication means there is no polynomial time adversary for the scheme where if the server recognizes a legitimate tag which is not corrupted, but there is no matching conversation among the tag, the reader and the server. Server authentication means that there is not any polynomial time adversary for the scheme where if a legitimate tag outputs OK but there is no matching conversation among the tag, the reader and the server. Here we concentrate on such mutual authentication between the tag and server. If the RFID scheme provides mutual authentication, it is secure.

Definition 1 About *Matching*: $\mathscr{A}$ can call all of the send queries in all the sessions except a target one with the special session number π. $\mathscr{A}$ calls $SendReader(\pi)$ to get a fixed $\widehat{M_1} = \widehat{R_r}$ at time t_0 and ends the session by simulating the latter input data $\widehat{M_2}$, $\widehat{M_3}$, $\widehat{M_4}$ and $\widehat{M_5}$ for *Send* queries. Moreover, $\mathscr{A}$ can call different sessions starting at time t_i, with obtaining the transcript $\{M_1, M_2, M_3, M_4, M_5\}$. For each i, $t_0 < t_i$ and $(\widehat{M_1}, \widehat{M_2}, \widehat{M_3}, \widehat{M_4}, \widehat{M_5}) \neq (M_1, M_2, M_3, M_4, M_5)$, we call the status as *non-matching*.

3. *Privacy*: this notion is defined as a series of games to check the value that the attacker $\mathscr{A}$ outputs. At first the attacker queries corresponding oracles according to the privacy class. The table $\mathbb{T}$ is given to $\mathscr{A}$ when the queries are over. Through $\mathscr{A}$'s analysis about $(vtag_i, ID_i)$ via the information from the queries, $\mathscr{A}$ outputs either *true* or *false*. If the probability of *true* is not negligible, $\mathscr{A}$ wins. The RFID scheme is $P-private$ if all such attackers belonging to class P are trivial accompanied with a blind and trivial attacker. It only checks the tags, but not for the readers or the server.

Definition 2 A blinder $\mathscr{B}$ is a polynomial-time simulator which simulates *Launch*, *SendReader*, *SendTag*, *SendServer*, *Timer* and *Result* oracles. $\mathscr{A}^{\mathscr{B}}$ is a blinded attacker who does not use the above five real queries but only uses simulated oracles as the results, e.g., employing random numbers as results. An attacker $\mathscr{A}$ who can use the above random oracles is trivial if $\mathscr{B}$ exists where $|Pr[\mathscr{A}\ wins] - Pr[\mathscr{A}^{\mathscr{B}}\ wins]|$ is negligible.

4.2 Security proof

Theorem 1 *The proposed scheme is correct, secure and TIMEFUL-WEAK private.*

Proof 1 According to Sect. 4.1.4, we divide the proof into three parts and prove them according to Vaudenay (2007) and Paise and Vaudenay (2008). Here we define that the random numbers and hash results owns a security length l, q_s is the most time of querying *Send* and q_h is the most time of querying the hash oracles.

1. **Correctness**: we consider the ability of correct identification for this property. Suppose there are $n+1$ tags in total, including the target tag. Note that in our scheme, the critical key for searching the identity is the $l-bit$ binary string PID_i. Also B_1 is the verification string to authenticate $\mathscr{T}_i$. For a target tag, $\mathscr{A}$ can use the specially given input data R_r, R_t and any of other one tag's information including PID_k, ID_k and x_k to make a $SendReader(M_2, \pi)$ to get M_3 which needs to be authenticated by S, where $k \neq i$. In fact, we can see if

we only require the output of tag identity, S only needs to authenticate the elements of M_2 included in M_3. In other words, we can say that the aim of this property for $\mathscr{A}$ to forge a correct M_2. If the target tag is recognized correctly, we say that $\mathscr{A}$ wins. The probability of the wrong recognition is bounded by $\frac{O(n^2)}{2^{2l}}$, including checking PID_i and B_1 simultaneously, which are both the negligible probability $\frac{O(n)}{2^l}$. Since there is only one reader and one server, the reader and the server cannot be identified incorrectly.

2. **Security**: here we consider if all the messages can be produced correctly by $\mathscr{A}$. First we suppose M_1 in each session is different from other sessions so that it has its own relativity to the session number π. Like we have demonstrated in the last property, guessing PID_i and passing the verification of B_1 are both needed to make the server outputs the correct tag identity. Since the two elements are hash results calculated with the secret key of that session, $\mathscr{A}$ cannot get useful information from the former or latter transcripts. So the condition of non-matching is reached. Like the above analysis, we consider the case of producing a legal M_2. The probabilities of reaching the two goals are both bounded by $\frac{O(q_s)}{2^l}$. The data generated by the reader, including $PRID_j$ and B_2. The probabilities are also $\frac{O(q_s)}{2^l}$. On the other direction, B_3 and B_4 can be forged by $\frac{O(q_h)}{2^l}$.B_5 can be faked with the probability $\frac{O(q_s)}{2^l}$. Finally, to make the tag output OK, B_7 should be checked. R_t is an easily generated string, so we only consider the case of forging the hash result B_6 included in B_7 and the probability is $\frac{O(q_s)}{2^l}$. Finally, we can see the above values are negligible.
3. **TIMEFUL-WEAK privacy**: to prove the TIMEFUL-WEAK level, we set a serial of blinder $\mathscr{B}_i$ to prove $\mathscr{A}$ has no obvious advantage over $\mathscr{A}^{\mathscr{B}}$ where $\mathscr{B}_i$ is fit for Definition 2. Here *Launch* is a trivial query and $\mathscr{B}_i$ needs not to make a special simulation but uses the original one. At the beginning of every session, *Launch* is queried. And four games are analyzed below:

   - Game G_0: In this game, a blinder $\mathscr{B}_0$ who can use all random oracles is set. We can see that $\mathscr{A}$ is equal to $\mathscr{B}_0$ and $Pr[\mathscr{A}wins] = Pr[\mathscr{A}^{\mathscr{B}_0}wins]$.
   - Game G_1: We consider the *Send* oracles simulated by a blinder $\mathscr{B}_1$ stemming from $\mathscr{B}_0$. The difference between $\mathscr{B}_0$ and $\mathscr{B}_1$ is that $\mathscr{B}_1$ uses random numbers as the hash results. A can use oracles like *Execute* to analyze the messages in the public channel. According to the scheme, we divide the game into the following cases:

     - To forge $SendTag(M_1, vtag)$, $\mathscr{A}^{\mathscr{B}_1}$ uses a random string $\{0,1\}^l$ as the return value of B_1, not as $\mathscr{B}_0$ does, who uses the hash oracle. But since the query *Corrupt* is lacked, $\mathscr{A}$ cannot get any valuable information from the messages in the public channel. And he could not tell apart the difference between real B_1 and the random string. The probability is $\frac{O(q_s)}{2^l}$ at most. Similarly, it is the same probability for guessing PID_i.
     - To forge $SendReader(M_2, \pi)$, $\mathscr{A}^{\mathscr{B}_1}$ uses a random string $\{0,1\}^l$ as the return value of B_2. Also $PRID_j$ needs to be guessed. Both of the probabilities are $\frac{O(q_s)}{2^l}$ at most.
     - To forge $SendServer(M_3, \pi)$, both the last two cases should be combined to consider. The probability is $\frac{O(q_s^4)}{2^{4l}}$. It can even be ignored compared to $\frac{O(q_s)}{2^l}$.
     - To forge $SendReader(M_4, \pi)$, $\mathscr{A}^{\mathscr{B}_1}$ uses four random strings belonging to $\{0,1\}^l$ as the results of $h(x_i^{old}||x_i^{new}||P_i^{old}||P_i^{new}||ID_i)$, $h(x_j^{new}||P_j^{new}||R_r)$, $h(x_j^{old}||x_j^{new}||P_j^{old}||P_j^{new}||RID_j||R_r)$ and B_5. The probability of telling apart the real B_5 and a random string is $\frac{O(q_s)}{2^l}$. Similarly, for the remaining three strings, the probabilities are all $\frac{O(q_h)}{2^l}$.
     - To forge $SendTag(M_5, vtag)$, $\mathscr{A}^{\mathscr{B}_1}$ uses two random strings belonging to $\{0,1\}^l$ as the hash results of $h(x_{j2}||PRID_{j2}||R_r)$ and B_7. The probabilities of telling apart the above two hash results from random string are $\frac{O(q_h)}{2^l}$ and $\frac{O(q_s)}{2^l}$, respectively.

   - Thus, the probability totally in G_1 is $\frac{O(q_s+q_h)}{2^l}$. So $|Pr[\mathscr{A}^{\mathscr{B}_0}wins] - Pr[\mathscr{A}^{\mathscr{B}_1}wins]|$ is negligible.
   - Game G_2: We use a blinder $\mathscr{B}_2$ stemming from $\mathscr{B}_1$ to simulate the *Result* query. The difference between the two blinders is that the simulated $Result(\pi)$ needs the transcript generated from real *SendTag*, *SendReader* and *SendServer* queries. Or $\mathscr{B}_2$ can make cheats by outputting 1 at will, with its own simulated *Send* results. So in this game, the point is that to fake a whole session which is acceptable for the simulator. In other words, all fakes in Game G_1 should be completed during one session. The probability is still $\frac{O(q_s+q_h)}{2^l}$. So $|Pr[\mathscr{A}^{\mathscr{B}_1}wins] - Pr[\mathscr{A}^{\mathscr{B}_2}wins]|$ is negligible.
   - Game G_3: We use a blinder $\mathscr{B}_3$ stemming from $\mathscr{B}_2$ to simulate *Timer* query. To simulate this query, $\mathscr{B}_3$ only need output the measured time cost in server. If there are significant differences between them, the tag can be tracked by a TIMEFUL adversary. According to Sect. 4.1.3, there is no exhaustive linear search which needs some extra computations such as hash function. So the result of the simulated *Timer* from $\mathscr{B}_3$ is same as the real *Timer*, and

there is no advantage for this simulation. We see that $Pr[\mathscr{A}^{\mathscr{B}_2} wins] = Pr[\mathscr{A}^{\mathscr{B}_3} wins]$.

– Up to now, five sorts of queries are all simulated by $\mathscr{B}_3$. We can consider $\mathscr{B}_3$ is $\mathscr{B}$, and $|Pr[\mathscr{A} wins] - Pr[\mathscr{A}^{\mathscr{B}} wins]$ is negligible. Or this theory is proved.

4.3 Forward and backward untraceability

According to Lim and Kwon (2006) and Alagheband and Aref (2014), if the tag is corrupted, $\mathscr{A}$ could track the message, in order to know state of the tag for a specific moment. Based on the tag corruption time, the attacker $\mathscr{A}$ can collect information from two directions: after the time point and before the time point. In other words, we may say forward and backward. Thus, two types of untracebilities can be deduced: forward untracebility and backward untracebility. They are illustrated as follows.

– Forward untraceability: even if a tag is breached at the r-th conversation, it is hard for a STRONG adversary to track the tag at the r'-th conversation such that $r' \geq r+2$. Here $(r+1)$-th conversation cannot be seen by $\mathscr{A}$.
– Backward untraceability: even if a tag is breached at the r-th conversation, it is hard for a STRONG adversary to track the tag at the r''-th conversation such that $r'' < r$.

Moreover, the tag's identity ID cannot be leaked while other data can be obtained by $\mathscr{A}$.

Theorem 2 *The proposed scheme keeps forward untraceability and backward untraceability.*

Proof 2 We prove the following two cases:

– Forward untraceability:
$CreateTag(ID_0), CreateTag(ID_1)$
$vtag \leftarrow DrawTag(ID_c)$ where $c \in \{0,1\}$
$\{PID_c^r, x_c^r\} \leftarrow Corrupt()$ at time interval $[r-1, r+1]$ before M_5^r is received by tag ID_c
$Free(vtag)$
$vtag^a \leftarrow DrawTag(ID_a)$ randomly from the two tags
$\mathscr{A}$ selects another time interval after $(r+1)-th$ session, usually $r+2$
$(r+1)$-th conversation is a synchronizing process. We can see $PID_a^{r+2} = h(x_a^r || R_t^{r+1} || R_r^{r+1} || PID_a^r)$ and $x_a^{r+2} = h(x_a^r || R_t^{r+1} || R_r^{r+1} || ID_a)$
Then $\{M_1^{r+2}, M_2^{r+2}, M_3^{r+2}, M_4^{r+2}, M_5^{r+2}\}$ are obtained by $\mathscr{A}$, and all of the messages are only relative to (PID_a^{r+2}, x_a^{r+2}) but not (PID_a^r, x_a^r).
Hence $\mathscr{A}$ cannot know which tag is drawn in the second time.
– Backward untraceability:
$CreateTag(ID_0), CreateTag(ID_1)$
$vtag \leftarrow DrawTag(ID_c)$ where $c \in \{0,1\}$
Here $\mathscr{A}$ obtains messages from a conversation, usually the $(r-1)$-th
$\mathscr{A} \leftarrow \{M_1^{r-1}, M_2^{r-1}, M_3^{r-1}, M_4^{r-1}, M_5^{r-1}\}$
$Free(vtag)$
$vtag^a \leftarrow DrawTag(ID_a)$ randomly from the two tags
$\{PID_a^r, x_a^r\} \leftarrow Corrupt()$ at time interval $[r-1, r+1]$ before M_5^r is received by tag ID_a
The messages are relative to (PID_c^{r-1}, x_c^{r-1}) and it is impossible to judge which tag is drawn in the second time only by obtain $\{PID_a^r, x_a^r\}$.

It is clear that the theorem can be deduced.

5 Informal analysis

We illustrate the security characters mentioned in Sect. 2.2. Also, some recently lightweight schemes (Srivastava et al. 2015; Li et al. 2015a; Gope and Hwang 2015; Safkhani et al. 2014) are compared in this section. The comparison results are demonstrated in Table 2.

Table 2 Security characters comparison

	Ours	Srivastava et al. (2015)	Li et al. (2015a)	Gope and Hwang (2015)	Safkhani et al. (2014)
Mutual authentication	✓	×	×	×	×
Tag anonymity	✓	✓	×	✓	✓
Reader anonymity	✓	*N / A*	×	×	*N / A*
Scalability	✓	×	✓	✓	✓
Resistant to replay attack	✓	✓	✓	✓	✓
Resistant to de-synchronization attack	✓	×	×	✓	✓
Resistant to data forgery attack	✓	✓	×	✓	×
Scheme for all tags	✓	×	×	×	✓

5.1 Mutual authentication

To authenticate $\mathscr{T}_i$ and $\mathscr{R}_j$, S verifies both B_1 and B_2. Then $\mathscr{R}_j$ checks B_5 to authenticate S. Here we concentrate on B_3 and B_7. S sends B_3 to $\mathscr{R}_j$ and $\mathscr{R}_j$ uses x_j^{new} and P_j^{new} which can only be computed by S and $\mathscr{R}_j$ itself to calculate B_6. Obviously B_6 is a string which can only be calculated by S and $\mathscr{T}_i$. $\mathscr{T}_i$ checks B_7 with B_6 and R_t. So via B_7, $\mathscr{T}_i$ authenticates S and $\mathscr{R}_j$ and the three entities achieve mutual authentication between either of the two.

However, we should point out that the schemes in [Srivastava et al. (2015), Gope and Hwang (2015), Safkhani et al. (2014), Li et al. (2015a)] are destitute of mutual authentication between the reader and the tag.

5.2 Anonymity

In our scheme, both the tag and the reader are anonymous in the transmission. The adversary $\mathscr{A}$ cannot get the real identities of the tag and the reader from the messages since the identities are input data of the hash results. Also, the pseudo-identities of the tag and the reader vary in different sessions. Since each pseudo-identity is a hash result calculated by the last session's temporary secret key of the entity itself, both the reader and the tag cannot be traced by $\mathscr{A}$. Here we should point out that $\mathscr{A}$ can get the reader's identity from the messages in the scheme (Gope and Hwang 2015). And $\mathscr{A}$ can trace the reader easily. Besides, in the schemes of Srivastava et al. (2015) and Safkhani et al. (2014), the identity of the reader is not used and therefore we use N / A in Table 2.

5.3 Scalability

In our scheme, the server does not require exhaustive search operation for checking PID_i and $PRID_j$. No extra computation is needed before the search, e.g., even a lightweight hash function is not required. This is the privacy property *Timeful* discussed in Sect. 4. So our scheme is scalable. However, in the scheme (Srivastava et al. 2015), every record in database should be retrieved for checking. We use × in the table.

5.4 Resistant to the replay attack

Each new session $\mathscr{T}_i$ and $\mathscr{R}_j$ generate new random numbers R_t and R_r and this way prevents the replay attack. For example, if a new session starts with a fresh R_r sent from $\mathscr{R}_j$, $\mathscr{T}_i$ will generate a new R_t and the latter messages employ the two fresh nonces. They are obviously different from the historical ones. As a result, the replay attack can be avoid in our scheme.

5.5 Resistant to the de-synchronization attack

If $\mathscr{A}$ blocks M_4 or M_5, S will finish corresponding operations according to Case 2 or 3 in step 4 of our authentication phase in the next session. The steps in Authentication phase can synchronize the secret values between the three parties again. So this attack can be avoided in our scheme. But the schemes (Srivastava et al. 2015; Li et al. 2015a) are vulnerable for this attack due to lack of remediation if either of the last two messages is lost.

5.6 Resistant to the data forgery attack

In our scheme, $Data_i$ transmitted in M_4 is checked by $\mathscr{R}_j$ with calculating B_5. Both B_4 and B_5 are hard to forge since the secret key x_j is unknown to $\mathscr{A}$. So our scheme can resist this attack. But in Li et al. (2015a) and Safkhani et al. (2014), the transmitted data are not protected and any attacker can use a random string to replace the element which conveys the data. Then the reader will get wrong information about the tag.

5.7 Performance comparison

We compare the performance among the proposed scheme and recent schemes (Srivastava et al. 2015; Li et al. 2015a; Gope and Hwang 2015; Safkhani et al. 2014) which are also based on hash functions. We suppose the lengths of hash result, the timestamp and the random numbers are 160 bits long. And the related explanations are as follows:

– Any RFID-based e-health care system can employ our scheme, since the reader detects the tags first. According to our demonstration in Sect. 1, most tags are of this kind. But the scheme (Gope and Hwang 2015) employs the way that the tag sends messages first. This is only for active tags, which are more expensive and less used (rfidjournal.com 2013). This point is mentioned in Sect. 1.1. Furthermore, timestamps are used in Li

Table 3 Storage in tags

	Ours	Srivastava et al. (2015)	Li et al. (2015a)	Gope and Hwang (2015)	Safkhani et al. (2014)
Storage in tag (bits)	480	320	320	$352 + 320d$	480

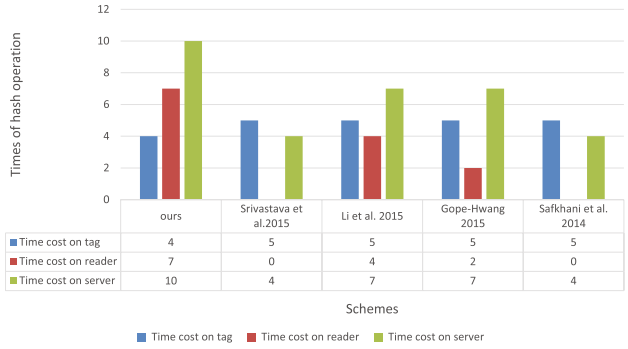

Fig. 3 Time cost of tag, reader and server

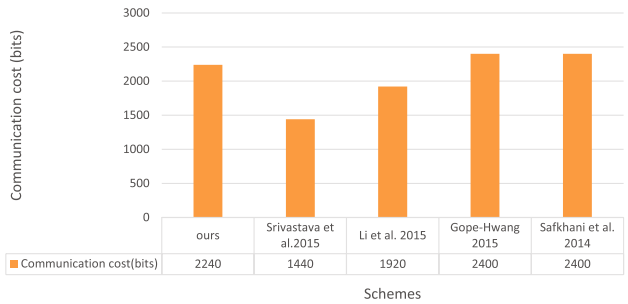

Fig. 4 Communication cost

et al. (2015a) and Srivastava et al. (2015) and the way is beyond Gen-2, which we have mentioned in Sect. 1.2.

- For the storage cost in the tag in Table 3, our scheme costs higher than schemes in Srivastava et al. (2015) and Li et al. (2015a), and same as Safkhani et al. (2014), but better than Gope and Hwang (2015). Here d means the pair of the pseudo-identity and emergency key stored in the registration phase.
- In Fig. 3, we consider how many times of hash functions each entity uses. Our scheme costs the least time on the tag side. It is fit for the tag which owns weak computation ability. The reason that our scheme costs the highest on the reader side and the server side is that our scheme avoids the hazard of tracking reader and the tag. Moreover, we make the secret keys and pseudo-identities of the tag and the reader change in every session.
- From Fig. 4, the communication cost of our scheme is in the middle.
- Above all, the most important issue is security. From Table 2 we can see our scheme satisfies all security properties while each of others has its own weaknesses.

So our scheme is the best one among the four listed schemes.

6 Conclusion

E-healthcare needs a secure communication circumstance, which satisfies the common security requirements, such as mutual authentication and anonymity. In this paper, we present a new RFID authentication scheme based on lightweight hash functions and bitwise-XOR functions. Then, we use the slightly extended formal model to prove that our scheme has properties such as correctness, security, TIMEFUL-WEAK privacy, forward untraceability and backward untraceability. Also through the informal analysis, our scheme overcomes the general attacks including the replay attack, the de-synchronization attack, etc. Moreover, tag and reader anonymity are kept in the scheme and all three entities in the session can achieve mutual authentication. By comparing with the recent hash-based schemes, ours is well-performed and eligible for e-healthcare applications.

Acknowledgements This research is supported by Fujian Education and Scientific Research Program for Young and Middle-aged Teachers under Grant No. JA14369, University Distinguished Young Research Talent Training Program of Fujian Province (Year 2016), and the this work was supported by the National Natural Science Foundation of China under Grant No. 61300220, and the Scientific Research Fund of Hunan Provincial Education Department under Grant No. 16B089. Also, it is supported by PAPD and CICAEET. Moreover, Dr. Saru Kumari is sponsored by the University Grants Commission, India through UGC-BSR Start-up grant under Grant No. 3(A)(60)31.

References

Akgün M, Bayrak AO, Çağlayan MU (2015) Attacks and improvements to chaotic map-based rfid authentication protocol. Secur Commun Netw. doi:10.1002/sec.1319

Alagheband MR, Aref MR (2014) Simulation-based traceability analysis of RFID authentication protocols. Wireless Person Commun 77(2):1019–1038

Amendola S, Lodato R, Manzari S, Occhiuzzi C, Marrocco G (2014) Rfid technology for iot-based personal healthcare in smart spaces. IEEE Internet Things J 1(2):144–152

Avoine G, Coisel I, Martin T (2010) Time measurement threatens privacy-friendly RFID authentication protocols. In: Radio frequency identification: security and privacy issues. Springer, Berlin, pp 138–157

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

Burmester M, De Medeiros B, Motta R (2008) Anonymous rfid authentication supporting constant-cost key-lookup against active adversaries. Int J Appl Cryptogr 1(2):79–90

Chen L, Ji J, Zhang Z (2013) Wireless network security: theories and applications. Springer Science and Business Media, New York

Chen Y, Chou JS (2015) Ecc-based untraceable authentication for large-scale active-tag rfid systems. Electron Commerce Res 15(1):97–120

Cho JS, Yeo SS, Kim SK (2011) Securing against brute-force attack: a hash-based rfid mutual authentication protocol using a secret value. Comput Commun 34(3):391–397

Fernando H, Abawajy J (2011) Mutual authentication protocol for networked rfid systems. In: 2011 IEEE 10th international conference on trust, security and privacy in computing and communications (TrustCom), IEEE, pp 417–424

Fu Z, Sun X, Liu Q, Zhou L, Shu J (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans Commun 98(1):190–200

Fu Z, Huang F, Sun X, Vasilakos A, Yang CN (2016) Enabling semantic search based on conceptual graphs over encrypted outsourced data. IEEE Trans Serv Comput. doi:10.1109/TSC.2016.2622697

Fu Z, Ren K, Shu J, Sun X, Huang F (2016b) Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst 27(9):2546–2559

Fu Z, Sun X, Ji S, Xie G (2016c) Towards efficient content-aware search over encrypted outsourced data in cloud. In: IEEE INFOCOM 2016—the 35th annual IEEE international conference on computer communications, IEEE. doi:10.1109/INFOCOM.2016.7524606

Fu Z, Wu X, Guan C, Sun X, Ren K (2016d) Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans Inf Forensics Secur 11(12):2706–2716

Ghaemmaghami SSS, Mirmohseni M, Haghbin A (2015) A privacy preserving improvement for SRTA in telecare systems. arXiv:151004197

Gope P, Hwang T (2015) A realistic lightweight authentication protocol preserving strong anonymity for securing rfid system. Comput Secur 55:271–280

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

He D, Zeadally S, Wu L (2015) Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst J. doi:10.1109/JSYST.2015.2428620

Huang HF, Yu PK, Liu KC (2014) A privacy and authentication protocol for mobile RFID system. In: 2014 IEEE international symposium on independent computing (ISIC), IEEE, pp 1–6

Jiang Q, Khan MK, Lu X, Ma J, He D (2016) A privacy preserving three-factor authentication protocol for e-health clouds. J Supercomput 72(10):3826–3849

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

Kumar N, Kaur K, Misra SC, Iqbal R (2015) An intelligent rfid-enabled authentication scheme for healthcare applications in vehicular mobile cloud. Peer-to-Peer Netw Appl. doi:10.1007/s12083-015-0332-4

Li CT, Weng CY, Lee CC (2015a) A secure rfid tag authentication protocol with privacy preserving in telecare medicine information system. J Med Syst 39(8):1–8

Li N, Mu Y, Susilo W, Guo F, Varadharajan V (2014) Privacy-preserving authorized RFID authentication protocols. In: Radio frequency identification: security and privacy issues. Springer, Berlin, pp 108–122

Li N, Mu Y, Susilo W, Guo F, Varadharajan V (2015b) Vulnerabilities of an ECC-based RFID authentication scheme. Secur Commun Netw. doi:10.1002/sec.1250

Lim CH, Kwon T (2006) Strong and robust RFID authentication enabling perfect ownership transfer. In: Information and communications security. Springer, Berlin, pp 1–20

Ma T, Zhou J, Tang M, Tian Y, Al-dhelaan A, Al-rodhaan M, Lee S (2015) Social network and tag sources based augmenting collaborative recommender system. IEICE Trans Inf Syst 98(4):902–910

Ng CY, Susilo W, Mu Y, Safavi-Naini R (2008) Rfid privacy models revisited. In: Computer security-ESORICS 2008. Springer, Berlin, pp 251–266

Ng CY, Susilo W, Mu Y, Safavi-Naini R (2009) New privacy results on synchronized rfid authentication protocols against tag tracing. In: Computer security—ESORICS 2009. Springer, Berlin, pp 321–336

Niu B, Zhu X, Chi H, Li H (2014) Privacy and authentication protocol for mobile RFID systems. Wireless Person Commun 77(3):1713–1731

Paise RI, Vaudenay S (2008) Mutual authentication in RFID: security and privacy. In: Proceedings of the 2008 ACM symposium on information, computer and communications security, ACM, pp 292–299

Piramuthu S (2011) RFID mutual authentication protocols. Decis Support Syst 50(2):387–393

rfidjournalcom (2013) Differences between active and passive tags. https://www.rfidjournal.com/faq/show?68 (online; accessed 23 Sept 2015)

Ryu EK, Kim DS, Yoo KY (2015) On elliptic curve based untraceable RFID authentication protocols. In: Proceedings of the 3rd ACM workshop on information hiding and multimedia security, ACM, pp 147–153

Safkhani M, Peris-Lopez P, Hernandez-Castro JC, Bagheri N (2014) Cryptanalysis of the Cho et al. protocol: a hash-based RFID tag mutual authentication protocol. J Comput Appl Math 259:571–577

Srivastava K, Awasthi AK, Kaul SD, Mittal R (2015) A hash based mutual RFID tag authentication protocol in telecare medicine information system. J Med Syst 39(1). doi:10.1007/s10916-014-0153-7

Vaudenay S (2007) On privacy models for RFID. In: Advances in cryptology—ASIACRYPT 2007. Springer, Berlin, pp 68–87

Wamba SF, Anand A, Carter L (2013) A literature review of RFID-enabled healthcare applications and issues. Int J Inf Manag 33(5):875–891

Wang S, Liu S, Chen D (2015) Security analysis and improvement on two RFID authentication protocols. Wireless Person Commun 82(1):21–33

Wu F, Xu L, Kumari S, Li X (2016) A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security. J Ambient Intell Hum Comput. doi:10.1007/s12652-016-0345-8

Wu S, Chen K, Zhu Y (2012) A secure lightweight RFID binding proof protocol for medication errors and patient safety. J Med Syst 36(5):2743–2749

Xia Z, Wang X, Zhang L, Qin Z, Sun X, Ren K (2016) A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans Inf Forensics Secur 11(11):2594–2608

Yang J, Park J, Lee H, Ren K, Kim K (2005) Mutual authentication protocol. In: Workshop on RFID and lightweight crypto, pp 17–24

Yu YC, Hou TW, Chiang TC (2012) Low cost RFID real lightweight binding proof protocol for medication errors and patient safety. J Med Syst 36(2):823–828

Yuan C, Sun X, Lv R (2016) Fingerprint liveness detection based on multi-scale LPQ and PCA. China Commun 13(7):60–65

Zhang Y, Sun X, Wang B (2016) Efficient algorithm for k-barrier coverage based on integer linear programming. China Commun 13(7):16–23

Zhu H, Zhao Y, Ding S, Jin B (2011) An improved forward-secure anonymous rfid authentication protocol. In: 2011 7th international conference on wireless communications, networking and mobile computing (WiCOM), IEEE, pp 1–5