Handling Authentication and Detection Probability in Multi-tag RFID Environment

Abstract: In Radio Frequency Identification (RFID) technology, an adversary may access classified information about an object tagged with RFID tag. Therefore, authentication is a necessary requirement. Use of multiple tags in an object increases the detection probability and simultaneously ensures availability of multiple resources in the form of memory and computability. Authentication process in multi-tag arrangement may increase the traffic between reader and object and/or decrease the detection probability. Therefore the challenge is to keep intact the detection probability without increasing the traffic. Existence of multiple number of tags helps to distribute the authentication responsibility for an object among multiple number of tags. In this paper, we assume that an object is attached with multiple number of active tags and in each session a randomly selected tag is responsible for authentication process. The detection probability is intact since an active tag within the range of reader can be an intermediator.

Keywords: authentication; multi-tag; RFID; active tag.

1 Introduction

RFID technology helps to identify an object efficiently. A small chip called RFID tag is attached with an object and the relevant information about it are kept in the memory of that chip. Four type of RFID tags are available and they are active, passive, semi-passive, and semi-active. A RFID reader scatters an electromagnetic signal to read some information about the objects within the communication range. On reply, the RFID tag backscatters a signal which contains the information requested from reader. The tags used in any kind of object should be affordable and hence need to be cheaper. However, a low-cost tag can be built with the use of limited resources such as computation ability, communication ability, and memory, etc. Since the memory of a tag is limited, the whole information about an object may not be kept. In those situations, only the identification information is kept in the tag and other information about the object are kept in the database in a workstation termed as backend server. The tag responds with the identification information termed as id and the reader use this id to retrieve information about the object from backend server.

Any information from a tag needs to be legitimate. This is because many non-legitimate entities may try to respond with fake information. For example, a customer in a shopping mall may put a fake tag removing the original tag from an item and may be able to buy that item in much less cost. She would be successful because the validity of the information from tag has not been checked. Therefore, any response from tag needs to be checked for its validity. A non-legitimate reader also can fool a legitimate tag by sending fake information and the tag may update that information in its memory without checking its validity. Thus the legitimate tag becomes non-legitimate. Therefore, the determination of validity of any information exchanged between reader and tag is an important requirement and hence needs authentication.

An adversary may misuse the information during authentication process and can perform many attacks. Therefore, the authentication process needs to be secure in such a way that the probable attacks can be avoided. The classical cryptography techniques although can provide maximum security benefits, however, cannot be applicable to this kind of communication due to various resource limitations. Hence there is a requirement to use lightweight cryptography schemes that are applicable to this kind of resource constraint devices.

The detection probability of an object is less (Bolotnyy and Robins, 2007) if we use single tag in it. This is because the position at which the tag has attached with may not be within the communication range of reader. However, some other positions of the same object may be within the communication range. Thus, being within the communication range of reader, the object is undetectable. The use of multiple tag in the same object can solve this problem (Bolotnyy and Robins, 2007). The idea is that the multiple number of tags are attached in such a way that at least one tag is visible to reader if any part of the same object is within the communication range of reader. It has been proved that the detection probability has increased enormously using this idea (Bolotnyy and Robins, 2007).

In earlier authentication schemes (Weis et al., 2004; Tan et al., 2007; Tsudik, 2007; Burmester et al., 2006; Conti et al., 2007; Zhang et al., 2008; Tzu-Chang et al., 2010; Jing Huey et al., 2010; Kim and Jun, 2010), researchers did not assume the existence of multiple number of tags in the same object and hence the detection probability in their schemes are less. Use of multiple number of tags in the same object increases the detection probability (Bolotnyy and Robins, 2007) and to keep the detection probability intact, we may replicate the authentication information into all the tags and hence any tag within the communication range may successfully do the authentication. However, any updated information may cause the other tags to be desynchronized. In another solution, a threshold number of tags can do the authentication which increases the traffic between reader and object. We propose a lightweight authentication scheme assuming that there is multiple number of active tags in the same object. This helps to keep intact the detection probability.

In our approach, a number of tags are attached with an object one among which is selected as master and this is responsible to carry out all kind of tasks relevant to authentication of that object. In each successful session, the backend server selects a new tag as master that will take the responsibility in the next session. The backend server sends this information to current master tag along with the updated information. The current master tag transfers the master responsibility to newly selected tag.

In accordance with the state of the art and our approach, we have tried to find out the answers to the following questions.

1. How an object and a reader can be authenticated with each other?
2. What are the security benefits we can obtain using extra resources?
3. What is the resiliency of our scheme in comparison to others?
4. What is the performance of our scheme?

The remainder of the paper is organized as follows. In section 2 we have briefly discussed the related schemes which have been proposed recently. In section 3, we have introduced the communication model we have assumed and the vulnerabilities in it. We have described our proposed authentication scheme in section 4. In section 5, we have analyzed the proposed scheme and compared it with the exiting schemes. We have concluded with the references in section 6.

2 Related works

In literature, we found a number of authentication schemes and those are based on the idea that an object is attached with single tag and hence the detection probability of object for those schemes are less.

Many authors (Weis et al., 2004; Tan et al., 2007; Tsudik, 2007; Burmester et al., 2006; Conti et al., 2007) have proposed hash based authentication scheme. The schemes proposed by Weis et al. (2004) are such kind of authentication scheme. According to their protocol, a tag will be given a meta id which is a hashed value of secret key. After getting this id the tag gets locked. Reader initiates a communication and the tag responds with the meta id it has assigned. Reader checks the validity of tag and sends the corresponding secret key on validation. This secret key is used to unlock the tag and the tag sends its id to reader. This is a very basic scheme which is prone to many attacks such as eavesdropping, location privacy, replay attack, etc. They have modified the scheme and proposed a **Randomized Hash-Lock (RHL)** scheme which resolves the location privacy problem (Weis et al., 2004). In this scheme, instead of sending the same meta id for every request, the tag generates a new random number and attaches it with its id and then get the hashed value from it. It then sends this value and hence the response is not traceable in other sessions. However, it is still prone to other kind of attacks.

In authentication scheme proposed by Zhang et al. (2008), the reader collects identifier and secrete key from backend server and uses those to prove the authentication to tag. The tag attached in an object responds with a session random number. The reader proves its authentication using this random number along with its own identifier and session key. The tag proves its authentication by providing its identifier and after the validation of identifier, the reader generates a new session random number and sends to tag. The tag receives the new session random number and replies with a OK message. This scheme may suffer form synchronization problem since an adversary may block the original updated session random number and send another arbitrary value. However, the tag is unable to detect the validity of new session random number and hence the tag and backend server get desynchronized. This also suffers from location privacy problem between two successful authentication sessions. If there is more readers then the tag needs to keep pairwise secrets for all reader and hence the tag may suffer from scalability problem.

Tzu-Chang et al. (2010) has improved the authentication scheme proposed by Chien and Chen (2007). They have removed the synchronization problem and introduced indexing in the database in backend server to speedup authentication process, and hence their improvement has made the Chien's scheme more stronger. According to their protocol, the reader initiates by sending a random number. The tag attached with an object replies with authentication information along with the index value C_i. The reader forwards these along with its signature and the random number it had sent to tag. The backend server checks the authentication of both reader and object and it generates updated information for the object on validity confirmation. The tag attached in the object updates its information after getting the updates from backend server. The backend server uses two tables, one keeps the most updated information and the other keeps the information checked last time the validity was checked successfully for the same object. Use of two records for same object helps to remove synchronization problem. This scheme has the following problems. If attacker changes the value of C_i to 0, then index becomes useless and in that case the backend server checks both new and old entry serially. However, if validity confirms using the information from new then there is unnecessary check for the information in old. Moreover, the tag sends same C_i on request from an adversary during the time between two successive and successful sessions and hence there is no location privacy during this period. There is no forward security. This is because; if attacker knows K_i then she will be able to generate K_{i+1} using Pseudo Random Number Generation (PRNG) function.

A fingerprint based mutual authentication protocol is proposed by Jing Huey et al. (2010). They use power response of a tag as the fingerprint. In their scheme, the reader initiates by broadcasting a request message. The tag generates authentication information using the fingerprint, the session key and the identifier. It then sends this information to backend server via reader. The backend server uses this as an index in the database and checks the validity. On successful validation, the backend server replies with updated information through reader. The tag then checks the validity of updated information and updates accordingly. Both backend server and tag update the session key using a PRNG function. The scheme does not guarantee any forward security since the disclose of K_i will help the adversary to generate K_{i+1}. This scheme also suffers from location privacy problem between consecutively two successful sessions.

Kim and Jun (2010) proposed a lightweight mutual authentication protocol for single tag, double tag and multiple tags. In the single tag authentication scheme, the reader initiates sending a hello message along with a random number. The tag generates another random number and two session keys and it uses these to generate an intermediate information $M_1||M_2$. It then sends this to reader along with the random number it had generated and a pointer which acts as index in backend server. The reader forwards these to backend server along with the random number it had generated earlier. The backend server generates a third session key and uses it to generate an intermediate information $M_3||N_3$ and sends it to tag via reader. Tag then checks the validity of M_3 and generates fourth session key and using this session key it generates final authentication information M_4. It sends this to server via reader. The backend server finally checks the validation of M_4 and thus authentication is completed. In another scheme, two tags are authenticated simultaneously. Actually they extend the authentication process for single tag which is applicable to authenticate two tags in such a way that the overall computation in the form of random number generation is less. They use the information generated by one tag as a random number for other tag. In third scheme, more than two tags are authenticated in the similar manner. However, the tag which was intervened first is authenticated after completion of authentication of all the other tags. The reuse technique decreases the computation overhead a lot. We found a few drawbacks in this scheme. For single tag authentication, any attack can be detected at the end of authentication process and hence there is some unnecessary computations in this situation. For double and and multiple authentication scheme, the tags are dependent on each other and the first tag will be authenticated at the end of authentication process.

The schemes (Weis et al., 2004; Zhang et al., 2008; Tzu-Chang et al., 2010; Jing Huey et al., 2010; Kim and Jun, 2010) we have described has one or more security flaws and all the schemes assume that an object is attached with single tag and hence has low detection rate. Therefore, there is a need to deduce an authentication scheme which will satisfy most of the security requirements with increased detection probability. We have proposed such a lightweight scheme which not only increases the detection probability but also satisfies the possible security requirements.

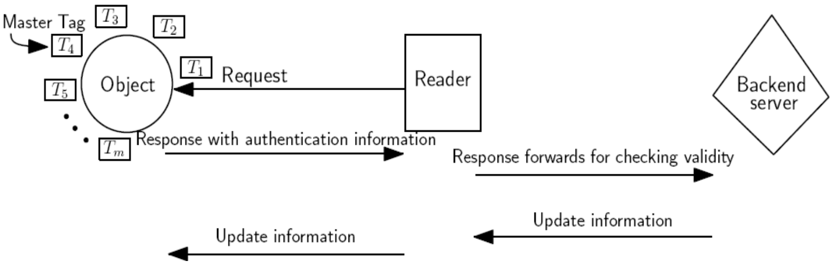
3 Communication model and possible threats in it

During authentication process, the RFID reader, backend server and RFID tag communicate with each other. In this section, we show the model we have assumed for communication during authentication process. The model has some vulnerabilities and any adversary may utilize these to implement a number of attacks. We have mentioned the possible attacks in the communication model in later part of this section.

3.1 Communication model

The components involves in RFID communication are RFID tag, RFID reader, and backend server. In a few works, the backend server has not been used and the RFID reader itself has necessary database. However, the reader may not have sufficient memory to keep information about all the objects in a set and hence may suffer from scalability problem. Due to this limitation, we keep backend server in our communication model. In figure 1, we have shown the communication model we have assumed.

Figure 1 Communication model



There are many objects. We have shown only one object in the figure for the sake of clarity. An object is attached with m number of tags in such a way that at least one tag is reachable to reader if any part of the object is within the communication range (Bolotnyy and Robins, 2007). In each session, a tag from these tags is selected as master. It is responsible to check the validity of any information it receives and also generates authentication information for the object. The master tag may not be reachable from reader and hence it accesses any information for the object through other tags. Each tag contains a static routing table and it is created just after the attachment of all tags in the object. A tag uses its routing table for sending any information if it knows the destination. Otherwise, it broadcasts.

RFID reader initiates the communication by sending a request message. It forwards the response from object to backend server and vice versa. We have assumed that the communication between reader and backend server is secure while the communication between reader and object is insecure.

Backend server is a workstation. We have assumed that it is scalable in terms of memory and computation. A database in it contains information about each object. The authentication information for each object are kept in a table within that database as a record.

3.2 Possible threats

Since the communication between RFID reader and object is insecure, any adversary may utilize it to implement a number of attacks. She may eavesdrop and silently listen to the sensitive information such as identifier, session key, etc. The adversary may try to get the location pattern of an object. Thus an object can be traceable by a non-legitimate entity. The adversary may modify the information communicated through insecure medium and hence implement the man-in-the-middle attack. The authentication information of a session may be saved and replayed for successful validation in later sessions. The adversary may somehow be able to compromise information in one session and try to obtain the information used in previous sessions and/or later sessions. The security requirement of this kind of attack is termed as backward and forward security respectively. In some situations, the information such as identifier, session key, etc. for an object are modified and the modified information is communicated from either reader to object or object to reader in each successful session. However, if an adversary blocks the updated information then there can be a synchronization problem between backend server and the object. The adversary may clone a tag and communicate fake information using that tag.

4 Proposed authentication scheme

The problems that we have identified are two fold, among which one highlights the need of a lightweight authentication scheme and the other is to keep the detection probability intact in multi-tag arrangement. Therefore, we have defined the problems as follows.

4.1 Problem definition

The owner of a set of objects wants to identify the objects she owned. Her identification process needs to be efficient and secure in such a way that any adversary cannot misuse any information relevant to objects she owns in any form.

An object is attached with multiple number of tags and a tag among those is selected as master. This is authorized to prove the legitimacy of the object and responsible to check the validity of any message from reader or any other entity. The challenge is that the master tag must do the task efficiently keeping the detection probability intact in multi-tag arrangement.

4.2 Our approach

In our approach, an object is attached with m number of active tags. In each session, a tag is selected randomly to do the authentication task. This tag is designated as master for the same session. The master tag responds with authentication information and checks

Table 1 Table of symbols

Symbols	Meaning	Size(bits)
n	Number of objects	$-$
m	Number of tags attached in G	$-$
T_i	i^{th} Tag in object G	$-$
TID_i	Tag id of T_i	b
TID_{old}	Tag id of previously selected master tag in backend server	b
TID_{new}	Tag id of newly selected master tag in backend server	b
MTID	Master tag id in master tag	b
$MTID_{new}$	Master tag id of newly selected master tag in backend server	b
$MTID_{old}$	Master tag id of previously selected master tag in backend server	b
N	Session key in master tag	$2b$
N_{new}	Session key of newly selected master tag in backend server	$2b$
N_{old}	Session key of previously selected master tag in backend server	$2b$
IN_i	Index of tag T_i	$\lceil \log_2 m \rceil$
L_i	Pairwise secret key of tag T_i	$2b$
L'_{nm}	Lower part of pairwise secret key of new master tag	b
g	Random number	b
v	Random number	$2b$
g_1	Random number	$b - \lceil \log_2 m \rceil$
$\oplus$	Exclusive-OR operator	$-$
$\|\|$	Attachment operator	$-$

the validity of any updated information. After successfully completing the authentication process, it transfers the master responsibility to newly selected master tag for the next session. In our approach there is a need of inter-tag communication without reader's intermediation for master responsibility transfer operation and also it helps in the situation when the master tag is not within the communication range of reader and any other tag attached to the same object is within the communication range then the later tag can receive and forward the information to master tag. Therefore, active tag as the tag entity is an appropriate choice. Table 1 shows the symbols we have used in our scheme.

4.3 Routing table

A tag attached in an object contains a routing table in its memory. It uses this table to send any information in shortest path to another tag attached in the same object. This is a static table and it is created immediately after the attachment of all tags in the object. Table 2 shows the routing table of tag T_2 in an object G as an example.

Table 2 Routing table of tag T_2

T_1	T_2	T_3	...	T_m
IN_1	IN_2	IN_4	....	IN_1

In table 2, first row shows the tags attached in the object and second row shows the index of nearest neighbor through which there is a shortest path from tag T_2 to the tag mentioned in the corresponding column. For example, if tag T_2 wants to send any information to tag T_3, it consults the routing table and obtains IN_4 as the index of nearest neighbor. Hence, it sends the information to tag T_4. The tag T_4 then sends it to its nearest neighbor and thus the information can reach to its correct destination.

4.4 Database:

We illustrates the data structure we have used in our proposed scheme. Figure 2 shows various fields in the memory of tag T_i in object G.

Figure 2 Information in a tag memmory

TID_i	L_i	Routing table	MTID	N

The *TID*$_i$ is the tag id of the tag. L_i is the pairwise secret assigned to it. Routing table contains the location information of other tags in G with respect to tag T_i. The other fields are applicable only for master tag. Among these, the *MTID* is master tag id and N is session key. The database in backend server has a table contains the authentication information about each object. Table 3 shows the various fields for an object G in backend server.

Table 3 Information for an object in backend server

$TID_1, TID_2, ..., TID_m$	$L_1, L_2, ..., L_m$	N_{new}	$MTID_{new}$	TID_{new}	N_{old}	$MTID_{old}$	TID_{old}

The first field and second field contains the tag ids and secret keys respectively assigned to G. The next three fields contains the latest values of master tag id, session key and the tag id and the last three values are old values of master tag id, session key, and the tag id.

4.5 The protocol

We have proposed an authentication scheme based on multi-tag arrangement with active tag as its tag entity. A few parameters are initialized and loaded into the memory of each tag as well as in the database in backend server.

Initialization: There are n objects. We do the same initialization process for all objects. However, for the sake of clarity, we describe the initialization process for an object G.

i) Both $MTID_{new}$ and $MTID_{old}$ fields in database kept in backend server are assigned with a unique master tag id *MTID*. Similarly both N_{new} and N_{old} fields are assigned with a unique session key N.

ii) Each tag T_i in G are assigned and loaded with separate tag ids *TID*$_i$ and secret key L_i.

iii) The tag ids and secret keys for G are kept in the database in backend server and a tag id among the stored tag ids is selected randomly for making the corresponding tag as master. The selected tag id is loaded in the fields TID_{new} and TID_{old} in the database.

iv) Both *MTID* and N are loaded in the memory of selected master tag.

v) Attach the tags in G in a proper alignment. Create the routing tables for each tags in object G and load the tables in respective tag memory.

Authentication: We have identified the entities involves during authentication of an object and they are reader, interface tag, current master tag, backend server, and new master tag. The *reader* in RFID communication scatters electromagnetic signal to identify any tag within its communication range. The tag lies within the communication range of reader and helpful in the situation when the current master tag is not within the communication range called *interface tag(λ)*. It usually receives information from reader and forwards them to current master tag and vice versa. The tag in object G which is responsible to prove authentication in current session is *current master tag*. This may or may not be within the communication range of reader. The workstation termed as *backend server* which contains the entire information about all the objects in its database and checks the validity of information from an object. This also generates new information about an object. In each session a new tag is selected randomly to do the authentication task. The tag which is selected in current session after successful authentication of G is the *new master tag*. The tasks assigned to reader, interface tag, current master tag, backend server and new master tag are specified in following algorithms.

Algorithm 1 executed by reader

```
Generates a random number $v$
Broadcasts a request message with $v$.
Receives $K$ and forwards it along with $v$ to backend server
Receives and forwards $P_1, P_2, P_3, P_4$ to object
```

Algorithm 2 executed by interface tag(λ)

```
Receives $v$ from reader
Adds $IN_\lambda$ with $v$ and broadcasts these to send to current master tag through 0 or more intermediate tags
Ignore any further $v, IN_\lambda$
Receives $K$ from current master tag through 0 or more intermediate tags
Forwards $K$ to reader
Receives $P_1, P_2, P_3, P_4$ from reader
Forwards $P_1, P_2, P_3, P_4$ to current master tag through shortest path
```

Algorithm 3 executed by a current master tag (cm)

```
Receives either $v$ directly from reader or $v, IN_\lambda$ from interface tag through 0 or more intermediate tags
Generates a random number $g$
Calculates $K \leftarrow [(MTID - g) \| g \oplus (N - v)]$
Sends $K$ to reader directly or through $\lambda$ and 0 or more intermediate tags
Receives $P_1, P_2, P_3, P_4$ directly from reader or through $\lambda$ and 0 or more intermediate tags
Calculates $(TID_{cm} \| g_1 \| IN_{nm}) \leftarrow P_1 \oplus (L_{cm} - N)$
Extracts $TID_{cm}$ and $IN_{nm}$
if the extracted $TID_{cm} \neq$ its own tag id then
    Stop
else
    Send $P_2, P_3, P_4$ to new master tag through shortest path using $IN_{nm}$
end if
```

Algorithm 4 executed by backend server

```
Receives $K, v$ from reader
valid $\leftarrow 0$
$j \leftarrow 1$
**repeat**
    Selects $N_{\text{new}}, MTID_{new}$ from $j^{th}$ record
    Calculates $(MTID^{'} - g)\|g \leftarrow K \oplus (N_{new} - v)$,
    Detaches $(MTID^{'} - g)$ and $g$
    Calculates $MTID^{'} \leftarrow (MTID^{'} - g) + g, j \leftarrow j + 1$
    **if** $MTID^{'} = MTID_{new}$ **then**
        $valid \leftarrow 1$
        $N_{cm} \leftarrow N_{new}, MTID_{cm} \leftarrow MTID_{new}, TID_{cm} \leftarrow TID_{new}$
    **end if**
**until** $valid = 1$ or $j > n$
**if** $valid = 0$ **then**
    $j \leftarrow 1$
    **repeat**
        Selects $N_{old}, MTID_{old}$ from $j^{th}$ record
        Calculates $(MTID^{'} - g)\|g \leftarrow K \oplus (N_{old} - v)$,
        Detaches $(MTID^{'} - g)$ and $g$
        Calculates $MTID^{'} \leftarrow (MTID^{'} - g) + g, j \leftarrow j + 1$
        **if** $MTID^{'} = MTID_{old}$ **then**
            $valid \leftarrow 2$
            $N_{cm} \leftarrow N_{old}, MTID_{cm} \leftarrow MTID_{old}, TID_{cm} \leftarrow TID_{old}$
        **end if**
    **until** $valid = 2$ or $j > n$
**end if**
**if** $valid = 0$ **then**
    Stop
**end if**
**if** $valid = 1$ **then**
    $j^{th}$ record has been satisfied for authentication of $G$
    Replaces $N_{old}, MTID_{old}, TID_{old}$ by $N_{new}, MTID_{new}, TID_{new}$ in $j^{th}$ record
**end if**
Randomly generates $MTID_{nm}, N_{nm}$
Selects a tag id $TID_{nm}$ randomly from the valid record for new master tag in $G$
Calculates $P_1 \leftarrow (TID_{cm}\|g_1\|IN_{nm}) \oplus (L_{cm} - N_{cm})$,
$P_2 \leftarrow (MTID_{nm} - TID_{nm}) \oplus (TID_{nm} - L^{'}_{nm})$,
$P_3 \leftarrow (L^{'}_{nm} - MTID_{nm}) \oplus TID_{nm}$,
$P_4 \leftarrow (N_{nm} + L_{nm}) \oplus (TID_{nm}\|MTID_{nm})$
Sends $P_1, P_2, P_3, P_4$ to reader
Replaces $N_{new}, MTID_{new}, TID_{new}$ by $N_{nm}, MTID_{nm}, TID_{nm}$ in $j^{th}$ record
```

Algorithm 5 executed by a new master tag (nm)

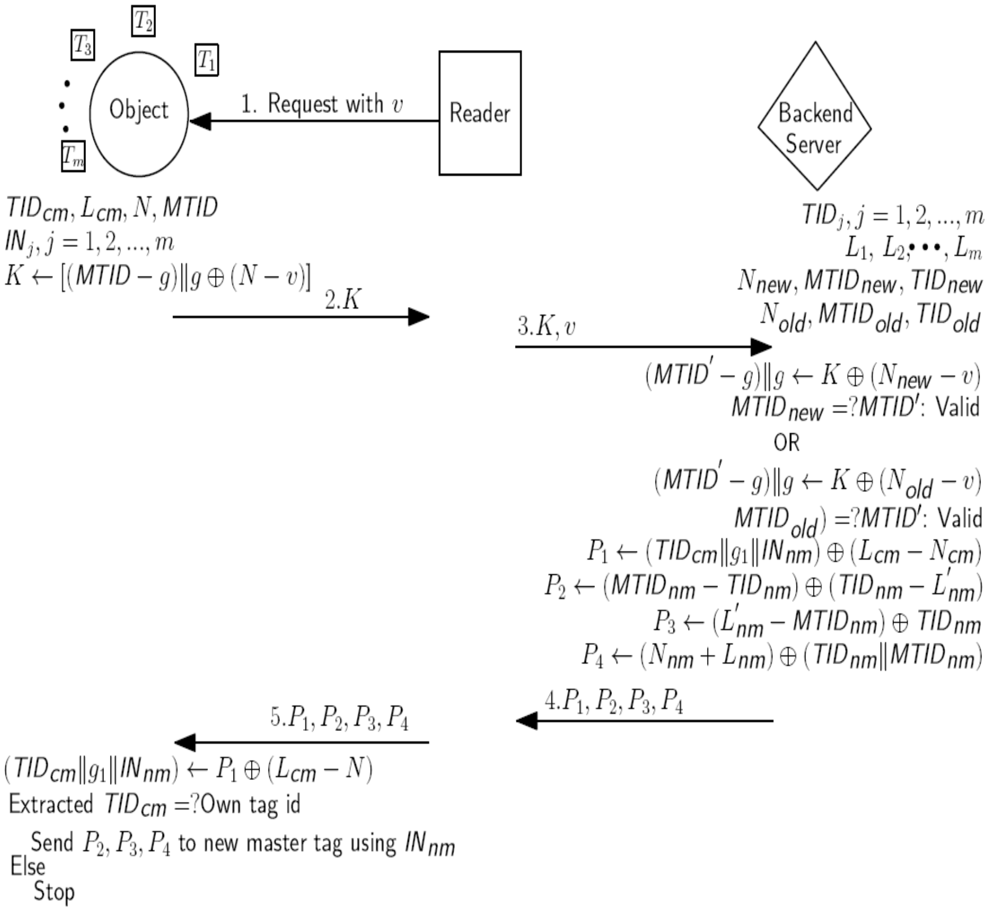
```
Receives $P_2, P_3, P_4$
Calculates $(MTID_1 - TID_{nm}) \leftarrow P_2 \oplus (TID_{nm} - L'_{nm})$
Calculates $MTID_1 \leftarrow (MTID_1 - TID_{nm}) + TID_{nm}$
Calculates $TID_{nm} \leftarrow P_3 \oplus (L'_{nm} - MTID_1)$.
if $TID_{nm} =?$ it's own id then
    Becomes new master tag and calculates
    $(N_1 + L_{nm}) \leftarrow P_4 \oplus (TID_{nm} \| MTID_1)$
    $N_1 \leftarrow (N_1 + L_{nm}) - L_{nm}$
    Stores $MTID_1, N_1$ in its memory
else
    Rejects the message
end if
```

Brief illustration: We briefly describe how authentication process takes place in our scheme. In initialization, each object is assigned with different set having m number of tags where each tag T_i is loaded with a tag id TID_i and a pairwise secret key L_i. A tag from each object is selected as master for the respective object and it is loaded with an id *MTID* and a session key N. These information are also kept in backend server. The tags are then attached with respective objects in proper alignment. In an object, the tags attached to it has an index *IN* and it is used during inter-tag communication. After attachment, a routing algorithm is executed for an object to generate routing tables for each tag in that object which will help a tag within the object to send any information in shortest path to another tag attached in the same object. The routing table for each tag is kept in the respective tag memory and it is static and hence does not need any modification afterwards.

During authentication, the RFID reader will generate a random number v and broadcast it as a request message. For an object G, if any of the tag attached to it is within the communication range then the object can be detectable. If the reachable tag is current master tag then it will generate an authentication information K and send to reader. Otherwise, the interfacing tag will receive v and broadcast this along with the index value (IN_λ) assigned to it. Any intermediate tag will simply receive and broadcast. Thus, v and IN_λ will reach to the current master tag. The current master tag will then generate authentication information K and send it to interface tag through shortest path along with index value IN_{cm} of its own. The index value IN_λ and routing table helps the current master tag to obtain the shortest path. The interface tag will receive K and IN_{cm} and send to reader. The reader will receive K and forward it to backend server along with v. The backend server, after receiving K and v, will check the authentication using the information in its database. If authentication fails then the backend server will stop and the session will be terminated. Otherwise, it will be treated as response from a legitimate object.

If the validity confirms using the latest information i.e, $N_{new}, MTID_{new}$ then the backend server will replace the old information i.e, $N_{old}, MTID_{old}$, and TID_{old} by $N_{new}, MTID_{new}$, and TID_{new} respectively. Otherwise it will keep intact the old information i.e, $N_{old}, MTID_{old}$, and TID_{old}. However in both situation, it will randomly generate new master tag id $MTID_{nm}$ and session key N_{nm}. It will then randomly select another tag id from the tag ids in corresponding record and then generate update information P_1, P_2, P_3, and P_4 and send these to reader. The backend server will then update the information $N_{new}, MTID_{new}$, and TID_{new} using the newly generated

Figure 3 Authentication protocol for multi tag RFID


T_2
T_3
T_1
T_m
Object
1. Request with v
Reader
Backend Server
$TID_{cm}, L_{cm}, N, MTID$
$IN_j, j = 1, 2, ..., m$
$K \leftarrow [(MTID - g)\|g \oplus (N - v)]$
2.K
$TID_j, j = 1, 2, ..., m$
$L_1, L_2, \cdots, L_m$
$N_{new}, MTID_{new}, TID_{new}$
$N_{old}, MTID_{old}, TID_{old}$
3.K, v
$(MTID' - g)\|g \leftarrow K \oplus (N_{new} - v)$
$MTID_{new} =?MTID'$: Valid
OR
$(MTID' - g)\|g \leftarrow K \oplus (N_{old} - v)$
$MTID_{old}) =?MTID'$: Valid
$P_1 \leftarrow (TID_{cm}\|g_1\|IN_{nm}) \oplus (L_{cm} - N_{cm})$
$P_2 \leftarrow (MTID_{nm} - TID_{nm}) \oplus (TID_{nm} - L'_{nm})$
$P_3 \leftarrow (L'_{nm} - MTID_{nm}) \oplus TID_{nm}$
$P_4 \leftarrow (N_{nm} + L_{nm}) \oplus (TID_{nm}\|MTID_{nm})$
4.P_1, P_2, P_3, P_4
5.P_1, P_2, P_3, P_4
$(TID_{cm}\|g_1\|IN_{nm}) \leftarrow P_1 \oplus (L_{cm} - N)$
Extracted TID_{cm} =?Own tag id
Send P_2, P_3, P_4 to new master tag using IN_{nm}
Else
Stop

information. The backend server will raise an alarm if the validation of authentication has carried using old information. The reader after getting P_1, P_2, P_3, and P_4, will forward these to object. The current master tag will obtain these information directly from reader or through interface tag. It will then check the validity using the pairwise secret key, session key, and its own tag id. If validity confirms, it will send P_2, P_3, P_4 to new master tag through shortest path using the extracted index value of new master tag. The new master tag, on getting these, will check validity and store the new master tag id and session key in its memory. Thus it will become the master tag for object G. Figure 3 provides a pictorial representation of our proposed authentication scheme.

5 Analysis of the scheme

It is desirable for any authentication scheme to be efficient and secure against possible attacks and hence we make a thorough analysis of our scheme in various aspects such as security, computations etc.

5.1 Authentication analysis

Our proposed scheme needs to ensure the validity of every messages transfers through insecure medium since authentication is major requirement in RFID communication as a problem we have encountered. We briefly describe how our scheme ensures the authentication of various components involved in this communication.

a) **Object**: The backend server checks the validity of K in algorithm 4. The source of this information is master tag of an object. If it matches any of records in backend server then the object is considered as legitimate. A similar information from a non-legitimate object will not be validated since there is no entry related to the same object in backend server. Thus our scheme distinguish between a legitimate and non-legitimate object.

b) **Reader**: Since the communication between reader and backend server is secure there is no requirement to check the validity of reader in backend server. However, the master tag checks the validity of information from reader in algorithm 3 and 5. Thus any information transfers through insecure medium from reader are checked and hence the reader is authenticated.

c) **Backend server**: In our scheme, although a reader send updated information to object through insecure medium, however, they have originally generated in backend server and the master tag of an object checks the validity of that information in algorithm 3 and 5. Thus our scheme also ensures the authentication of backend server.

5.2 Security analysis

During authentication process, an adversary may try to misuse the information during authentication. Therefore any authentication scheme needs to assure the necessary security requirements. Since the communication between reader and object is insecure the adversary may implement various attacks in it. We make an analysis of how our scheme offers the necessary security benefits against the possible threats we have mentioned in section 3.2.

a) **Eavesdropping:** The adversary may try to listen the information communicated during authentication. In our scheme, the original information transfers through insecure medium i.e. between reader and tag or from one tag to another tag are encrypted using the pairwise secret key, session keys etc. and hence the adversaries are unable to obtain any knowledge. Also the communication between reader and backend server is assumed to be secure and hence eavesdropping is not possible.

b) **Man in the middle attack:** The attacker may send fake information blocking the original information. Every entity checks the validity of information transfers through insecure medium and the information are encrypted using the secret keys in our scheme. Since the adversaries do not have those secrets they are unable to generate any fake information which can be validated. Therefore, man in the middle attack is not possible in our scheme.

c) **Replay attack:** Keeping the information used in a session, the adversary may try to reuse the same information to prove the validity. In our scheme, the reader sends a fresh random number i.e. nonce v in each session and the master tag use that nonce in the encrypted information. An adversary may store the valid information during any valid communication and try to use it in future to prove the validity. She would request the master tag to send authentication information by declaring herself as a legitimate reader. Since she does not know the session key N, she will be unable to get the *MTID*. She may now keep this information and when any legitimate reader will ask for the id, she would send the stored information by declaring herself as a legitimate tag. However, this information will not be verified since the old nonce v is involved with the message instead of new nonce sent by reader. Moreover, it is unable to inject the new nonce since it does not know the secrets *MTID* and N. Thus, our proposed scheme satisfies security requirement against replay attack.

d) **Location tracing:** Use of same information to generate encrypted information may help the adversaries to trace an object. Thus they can obtain the behavioral knowledge about an object. If we randomly change the information used for encryption they can not obtain any pattern from the acquired encrypted information. We have used this idea in our scheme. In each session, the fresh master tag id and session key are used to generate authentication information and hence there is no relational pattern between the authentication information in one session and in the next or previous session.

e) **Location tracing between two successive and successful sessions:** The use of fresh master tag id and session key forces us to update in both backend server and the master tag. Therefore we update the information in the master tag and the backend server when the validity is confirmed. Thus, the update is done only when there is a successful session. However, if an adversary request the object a number of time within the period between two consecutive successful sessions then the object will send the same responses in each request. Thus the adversary may get a location pattern and hence can trace. In our scheme, the master tag uses a fresh random number for generating the encrypted response in each request. Hence the adversary can not obtain any relation from these responses and thus our scheme prevents this kind of attack.

f) **Forward security:** The adversary if somehow obtain the variable secrets *MTID* and N in one session she may be able to compromise that session. Moreover, she would try to generate the session key and master tag id for next sessions from the gathered information in compromised session to compromise the next sessions. Our scheme prevents this kind of attack since in each session the generated master tag id and session key are not related to the previous information. Moreover, compromising any of these variable secrets will not help the adversary to get the newly generated variable secrets from the update information.

g) **Backward security:** In our scheme the adversary is unable to generate *MTID* and N used in previous sessions by compromising the authentication information in current session. This is achieved due to the use of fresh random numbers in successful sessions.

h) **Synchronization attack:** We update the information in both backend server and master tag to prevent location tracing and to provide forward and backward security. In our scheme the update information are generated in backend server and those information are sent in encrypted form through insecure medium to new master tag. However, if any adversary blocks the updates then the information in backend server and the master tag in object are not same. Thus the response from master tag can not be verified. To prevent this kind of attack, we keep the old information in the same record. If the adversary blocks the updates the master tag will response with old authentication information in next session and that response will be verified in backend server due to the fact that there is the old information for the same master tag. Moreover the backend server raise an alarm and if the number of such alarms is more than a threshold value then the necessary steps will be taken accordingly.

i) **Physical attack:** The adversary may try to clone the master tag and can act as a legitimate entity. In our scheme, the master responsibility is not fixed to a particular tag and in each session a new tag in the same object is selected as the master tag for the next session. Since there is more number of tags it is difficult for the adversary to guess the correct tag as master for the next session and hence she has to compromise all the tags which is not an easy task. Again cloning the current master tag, the adversary may be successful to prove validity using old information. However this kind of validity can be done for a certain number of time since the backend server will take necessary action afterwards. We are working on to decide the threshold value on which the backend server can decide that there is a physical attack. Therefore we do not say that our scheme can prevent physical attack fully. However, the use of multiple number tags increases the difficulty for the adversary to mount physical attack.

5.2.1 Comparison of security assurance:

We compare our scheme with the other existing schemes in accordance with the necessary security requirements during authentication. Table 4 illustrates the comparative study.

Table 4 Assurance of security

	a	b	c	d	e	f	g	h	i
Weis et al.	N	Y	N	N	N	NA	NA	Y	N
RHL (Weis et al.)	N	Y	N	Y	Y	NA	NA	Y	N
Zhang et al.	Y	N	P	Y	N	Y	N	N	N
Tzu-Chang et al.	Y	Y	Y	Y	N	N	Y	Y	N
Jing Huey et al.	Y	N	Y	Y	N	N	Y	N	Y
Kim and Jun	Y	Y	N	Y	Y	NA	NA	Y	N
Our scheme	Y	Y	Y	Y	Y	Y	Y	Y	P

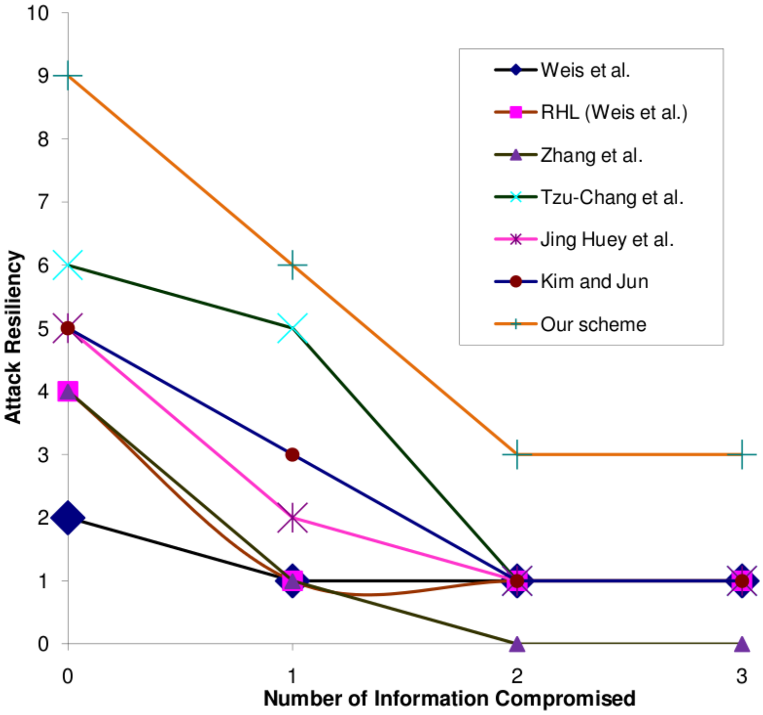
a: Eavesdropping, b: Man in the middle attack, c: Replay attack, d: Traceability, e: Traceability between two successive and successful sessions, f: Forward security, g: Backward security, h: Synchronization attack, i: Physical attack, Y: Satisfy, N: Not satisfy, P: Partially satisfy, *NA*: Not Applicable

In Table 4, we have specified the security requirements with the symbols a, b, c, etc. The meaning of symbols are written under the table. Each row indicates the scheme under consideration. Each column indicates a type of security requirement. Each entry in the table indicates the prevention status of security requirement in corresponding column for the scheme in corresponding row. There are four type of status and they are *satisfy, not satisfy,* *partially satisfy,* and *not applicable*. The symbols are Y, N, P, *NA* respectively. the last type of status is for the schemes which do not use any variable secrets. Therefore analysis of forward and backward security are not applicable to those schemes. From Table 4, we see that our scheme has satisfied all the security requirements we have mentioned except the physical security. However, use of variable tags for proving and checking authentication creates the physical attack harder and hence it is partially satisfied in our scheme. However, every schemes other than our scheme have one or more security flaws. Therefore, our scheme is more secure in compared to existing schemes.

5.2.2 Resiliency:

An adversary may somehow be able to compromise some sensitive information. We define a parameter, namely, resiliency which indicates the prevention capability of an authentication scheme against compromise of zero or more sensitive information. Compromise of some information may not necessarily make failure for an authentication scheme to prevent all the attacks we have mentioned in section 3.2. However, prevention capability of the scheme may decrease due to this compromise. Therefore, we have incrementally chosen the best combination of information and finds how many attacks are still preventable. The best combination implies that the compromise of a certain number of information which helps the adversary to mount maximum number of attacks. The other combination of same number of information does not help the adversary to mount this many attacks. We have done a thorough analysis of this kind for our scheme and the schemes we have visited in section 2 and plot a graph to compare the resiliency of our scheme and the other schemes.

Figure 4 Resiliency graph



In figure 4, we plot a graph where the X-axis represents the number of compromised information (best combination) and Y-axis indicates the resiliency of an authentication scheme. The graph for our scheme shows the maximum resiliency and keeps maximum on compromise of most number of information. The schemes (Tzu-Chang et al., 2010; Jing Huey et al., 2010; Kim and Jun, 2010) has good resiliency initially, however, falls quickly on compromise of few information and there is poor resiliency for the schemes (Weis et al., 2004; Zhang et al., 2008). Therefore, the resiliency of our scheme is best in comparison to the schemes we visited in section 2.

5.3 Operation

A few basic operations such as bitwise XOR, addition, subtraction, attachment, detachment, and random number generation has been used in our scheme. In section 2, we have described the existing authentication schemes and we have seen that they also have used these operations along with one or more hash functions. We have calculated the number of operation of each type to get an idea about efficiency of our scheme in compared to the existing schemes. To do this, we separately calculated the number of operations used in various components. The schemes we have visited in section 2 assume that an object is attached with single tag. However in our scheme, there is multiple number of tags in an object and the tags are classified as current master, new master, interface, and intermediate. In comparative study, we do not consider the different type of tags separately for our scheme. The type of tag which use maximum number of operation of a particular type will be considered as the tag entity for that particular operation.

Table 5 Number of operations performed in various scheme

	Tag					Reader					Backend Server				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
Weis et al.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RHL (Weis et al.)	0	0	1	1	1	0	0	n	n	0	0	0	0	0	0
Zhang et al.	1	0	0	$q+2$	0	2	0	0	3	1	0	0	0	0	0
Tzu-Chang et al.	8	0	0	6	1	1	0	0	1	1	$q+2n+7$	0	0	$q+n+5$	0
Jing Huey et al.	2	0	1	3	0	0	0	0	0	0	2	0	1	3	0
Kim and Jun	7	1	4	3	1	0	0	2	0	1	$4n+4$	n	$n+3$	$2n+2$	1
Our scheme	3	4	3	0	1	0	0	0	0	1	$2n+4$	$4n+5$	$2n+3$	0	4

a: XOR, b: Addition/Subtraction, c: Attachment/Detachment, d: Hash, e: Random number generation, q: Number of reader, n: Number of object

Table 5 shows the quantitative analysis of various operations performed in (Weis et al., 2004; Zhang et al., 2008; Tzu-Chang et al., 2010; Jing Huey et al., 2010; Kim and Jun, 2010) and in our proposed scheme. The operations we have considered in our analysis are *XOR, addition/subtraction, attachment/detachment, hash, random number generation* indicated by *a, b, c, d,* and *e* respectively. The components involved in RFID communication are RFID tags, RFID reader, and backend server. The Table 5 has three columns, namely, Tag, Reader, and Backend server. Each column is further divided into five sub columns. A sub column represents how many time the component specified in corresponding column performs a particular operation specified in it. Each row in Table 5 represents the scheme under consideration in our analysis. Therefore, an entry in the table indicates how many time a component specified in the corresponding column in the scheme specified in corresponding row performs an operation specified in corresponding sub column. We explore a comparative study by analyzing the entries in Table 5.

From table 5, we see that all the entities in our scheme does not use any hash function which is assumed to be a time consuming operation. However, in our scheme, the tag have use most number of addition and subtraction operation which are very basic operations. The other operations such as random number generation, attachment, detachment, etc. have used almost equally by tag entity in our scheme compared to other schemes. The reader in our scheme has only one computation, whereas in other schemes, the reader has many computations. The backend server on the other hand have used more number of operation such as XOR, addition, subtraction, attachment, detachment etc in our scheme. However, in some schemes (Tzu-Chang et al., 2010; Kim and Jun, 2010) the backend server also have used almost equal number of such operations. Therefore, since our scheme have used no hash operation and some more basic operations, we can say that our scheme is applicable and efficient in respect to computation.

5.4 Communication overhead

RFID tags(both active and passive) have less communication ability. This is because the passive tags usually communicates by using the power from reader and the active tags use its own battery. However, the active tags also have a limited communication ability. We have compared our scheme with the existing schemes in respect to the number of information which has been communicated by various entities during authentication process. Table 6 shows the comparative study. We have considered the communication overhead of the kind of tag among intermediate, current master, new master etc which sends and receives maximum number of information for our scheme.

Table 6 Communication overhead of various scheme

	Tag	Reader	Backend Server	*BRO*
Weis et al.	4	6	2	4
RHL (Weis et al.)	4	$n + 5$	$n + 1$	4
Zhang et al.	6	14	8	6
Tzu-Chang et al.	6	14	8	6
Jing Huey et al.	3	5	2	3
Kim and Jun	6	10	4	5
Our scheme	11	12	6	6

n: Number of objects, *BRO*: Between Reader and object

In Table 6, first three columns indicates the components involved in communication and the last column indicates the traffic between reader and object. Each row explores the schemes involved in our comparison. Therefore, each entry in the table corresponding to first three columns indicates the number of information communicated by the components specified in the corresponding column in the scheme specified in corresponding row. The entries in the last column indicates the traffic between reader and object for the scheme specified in corresponding row. From the table, we see that the communication overhead is maximum in our scheme in comparison to other schemes. This is due to the use of multiple number of tags and inter tag communication. However, the traffic between reader and object is almost same in comparison to other schemes. Therefore, although our scheme suffers from overall communication overhead due to the use of multiple number of tags in same object the scheme is capable to intact the communication overhead between reader and object.

5.5 Memory requirement

RFID tag also suffers from memory constraints and hence there is a need to do an analysis over memory requirements in various schemes. Table 7 shows the memory requirement in various entities for the existing schemes and our scheme.

Table 7 Memory requirement

	Tag	Reader	Backend Server
Weis et al.	2	0	$3n$
RHL (Weis et al.)	1	0	n
Zhang et al.	$2q + 2$	4	$4q + n$
Tzu-Chang et al.	4	1	$q + 8n$
Jing Huey et al.	2	0	$4n$
Kim and Jun	2	0	$2n$
Our scheme	$m + 4$	0	$n(2m + 6)$

q: Number of reader n: Number of objects m: Number of tags attached to an object

Besides listing the memory requirement in RFID tag, we have also listed the memory requirements in RFID reader and backend server which do not have such limitation. Each row in table 7 represents a particular scheme and each column represents a particular components. Therefore, an entry in the table indicates the number of information need to be kept in a component mentioned in corresponding column for the scheme mentioned in corresponding row. From the table we see that our scheme requires maximum memory. This is again due to the use of multiple number of tags in same object and inter tag communication. We have decreased the communication overhead using a static routing table in each tag. The static routing table in a tag contains the location information of other tags in respect to that tag. Hence for m number of tags in an object we requires at least m number of information in the routing table. Again, we have used some basic operations during authentication and hence requires us to keep more security related information. Thus the overall memory requirement for tag in our scheme is more in comparison to other schemes. However, there is no memory requirement in reader in our scheme whereas some other schemes (Zhang et al., 2008; Tzu-Chang et al., 2010) have this kind of requirement. In backend server, the memory requirement is almost same in comparison to other schemes.

6 Conclusion

RFID is most pervasive computing technology and hence easily susceptible to various kind of attacks. Therefore a secure authentication is necessary requirement in this technology. Attachment of multiple number of tags in an object increases the detection probability. However, replication of authentication information in each tag in the same object is not a good solution since there is a probability that some tags will get updated whereas some other tags will have old information. This will desynchronize the tags which have old information. Any threshold scheme may mitigate this problem, however, will cause an excessive traffic in between reader and object. We have proposed a lightweight authentication scheme which keeps intact the detection probability and also prevents most of the possible attacks. The resiliency of our scheme is best in compared to other schemes. Our scheme can also be able to detect the physical attack on current master tag depending on the threshold number of time an object is verified using old information. We are working on to decide the threshold value which can indicate the existence of such kind of attack. However, we have used active tag in our scheme which is costlier than passive tags. We assume that the active tags will be affordable in near future and the battery in it can be recharged using the electromagnetic signal scattered by RFID reader.

References

Bolotnyy, L. and Robins, G. (2007) 'Multi-Tag RFID Systems', *Journal of Internet Protocol Technology, Special issue on RFID: Technologies, Applications, and Trends*, Vol. 2, No. 3/4, pp.218–231

Weis, S.A., Sanjay, S., Ronald L.R. and Daniel W.E. (2004) 'Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems', *Proceedings of the First Security in Pervasive Computing, LNCS-2802 Springer*, pp.201–212

Tan, C., Sheng, B. and Li, Q. (2007) 'Serverless Search and Authentication Protocols for RFID', *Proceedings of the Fifth Anual IEEE International Conferance on Pervasive Computing and Communication (PerCom)*,

Tsudik, G. (2007) 'A family of dunces: Trivial RFID identification and authentication protocols', *Proceedings of PET*,

Burmester, M., Le, T.V. and Medeiros, B.D. (2006) 'Provably secure ubiquitous systems: Universally composable RFID authentication protocols', *Proceedings of SECURECOMM*, pp.1–9

Conti, M., Pietro, R.D. and Mancini, L.V. (2007) 'RIPP-FS: an RFID Identification, Privacy Preserving Protocol with Forward Secrecy', *Proceedings of the Fifth Anual IEEE International Conferance on Pervasive Computing and Communication (PerCom)*, pp.229–234

Zhang, Y., Li, D. and Zhu, Z. (2008) 'An Efficient RFID Tag-Reader Mutual Authentication Scheme', *Proceedings of WiCOM'08*, pp.1–4

Tzu-Chang, Y. Wanga, Y.J., Tsai-Chi, K. and Sheng-Shih W. (2010) 'Securing RFID systems conforming to EPC Class 1 Generation 2 standard', *Journal of Expert Systems with Applications*, Vol. 37, No. 12, pp.7678–7683

Jing, H.K., Widad I., Rahman, M.G., Younis, I. and Sulaiman, M.K. (2010) 'Security Problems in an RFID System', *Journal of Wireless Personal Communications, Springer, Special Issue on RFID*, pp.1–10

Kim, H.J. and Jun, M.S. (2010) 'Light-Weight Mutual Authentication RFD Protocol for Multi-Tags conforming to ESC Class-1 Generation-2 Standards', *Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pp.34–39

Chien, H.Y. and Chen, C.H. (2007) 'Mutual Authentication Protocol for RFID Conforming to EPC class 1 generation 2 standards', *Journal of Computer Standards and Interfaces*, Vol. 29, No. 2, pp. 254–259