A novel secure group RFID authentication protocol

[1]**SOURAV RANJAN SAHU,**
Gandhi Institute of Excellent Technocrats, Bhubaneswar, India
[2]**STUTEEREKHA PATTNAIK,**
Aryan Institute of Engineering & Technology, Bhubaneswar, Odisha, India

---

Abstract

The trend of researching group radio frequency identification devices (RFID) authentication protocol has become increasingly popular in recent years. One of the newest work in this area is from Batina and Lee, they presented a privacy-preserving multi-players grouping-proof protocol based on the elliptic curve cryptography (ECC), and claimed their protocol have the ability to resist five potential attacks, including compromised tag attack, man-in-the-middle attack, colluding tags attack, etc.

In this paper, we first take a counterexample to demonstrate their protocol is vulnerable to compromised tag attack. Then we propose a novel secure RFID authentication protocol, and analyze its security by merging formal analysis, provable security, and mathematical inductive method, so as to solve the weakness of Batina and Lee's work. Furthermore, compared with another two classic protocols (secure ownership transfer protocol (SOTP) and secure multiple group ownership transfer protocol (SMGOTP)), the performance analysis show that our protocol provides not only a lower tags' communication cost at about 50.0% and 14.3%, but also a lower reader's computation cost (approximate 14.5% and 55.1% respectively), when transferring a large number of tags.

Keywords group RFID authentication, compromised attack, elliptic curve, RFID, internet of things (IOT)

1 Introduction

Radio frequency identification devices (RFID) is the most common perception technique of IOT. It applies radio signals to automatically and uniquely identifies objects, has been widely applied in several important areas nowadays, such as Logistics Industry, Retailing, Apparel Industry, Asset Management, Identity Recognition, E-Business, etc. To be specific, RFID handheld devices are used to identify the information of identity (ID), and then the real-time information about the state of the corresponding objects or the surrounding environment can be transferred rapidly via 3G networks. It is generally used primarily for mobile enforcement of the policemen, traffic police and border armed police. There are practical scenarios where grouping-proofs could meaningful develop the capabilities of RFID-based systems [1]. For instance, (1) pharmaceutical sector could check a medicine which is sold joined with its prescription or with its information leaflet; (2) government paperwork could verify a single form which is enclosed with its corresponding stamp or label; (3) meetings or access control systems could generate a kind of evidence which a group of people are present at a specific location. (4) airport check-in desks could link your boarding card with your passport and baggage. Nevertheless, all sorts of security threats reported have been growing dramatically over the past several years. No doubt that authentication of RFID is the essence to tackle with the potential risks.

In fact, after the experiences with six decades from its invention, scientific research about the security of RFID has mainly concentrated on the function of authentication between one or two tags and a reader, such as Park and Hur [2] and Sadighian and Jalili [3]. However, group RFID authentication protocols remain have not been lucubrate until Batina and Lee [4]. More specially, Batina and Lee [4] extended the notion of RFID authentication protocol to the public-key cryptography, and proposed a privacy-preserving multi-players grouping-proof protocol which is exclusively dependent on the use of ECC.

After that, a large body of work concentrated on the topic of group RFID technology in recent years [5–12]. Sato and Mitsugi [13] proposed a group verification method, named ‘group coding’, which verified the integrity of a group of tags without a network connection. The ‘group coding’ could be aware of the number of RFID tags missing from the group when the group was attacked. But they cannot take any potential attacks into account. Fornaciari and Cucchiara [14] proposed a camera and RFID subtly mingle method which can hamper attacker to location in a real noisy and complex wide environment. Their method can address uncertain data and manage conflicts which combines from the two sources, thus they refrain from some potential intruders. Yang [15] proposed a secure multiple group ownership transfer protocol, which can perform ownership transfer across different authorities and achieve mutual authentication among tags, the reader and the verifier. They claimed that their protocol is able to prevent from replaying attack, eavesdropping and message modification, although not including compromised tag attack.

Our main contributions can be summarized as follow:

1) A novel secure group RFID authentication protocol is proposed.

2) The security of the proposed protocol is analyzed by merging formal analysis, provable security, and mathematical inductive method.

3) Compared with another two classic protocols (SOTP and SMGOTP), the performance analysis show that our protocol provides not only a lower tags’ communication cost at about 50% and 14.3%, but also a lower readers computation cost (approximate 14.5% and 55.1% respectively), when transferring a large number of tags.

This paper has been divided into the following sections: Sect. 2 presents the preliminaries; Sect. 3 provide a counterexample to state that the protocol in Ref. [4] cannot resist compromised tag attack; Sect. 4 presents the revised grouping proof protocol and proves its security; Sect. 5 carries out some experiments and analyzes the computation and communication loads with other two protocols. Finally, in Sect. 6 we show the conclusion.

2 Preliminaries

In this section, we first introduce ID-transfer scheme. Before introducing the definition, we will introduce the notation used in this work. We denote P as the base point on an elliptic curve, y and $Y = yP$ are the trusted verifier’s private key and public key pair, where yP denotes the point derived by the point multiplication operation on the elliptic curve group. We let the notation $x(T)$ to denote the x-coordinate of the point T on the elliptic curve. The values s_t and $S_t(= s_tP)$ are the private key and public key pair of tag t. One point should note, although the name suggests that it can be publicly known, that a tag should not reveal its public key during the execution of the protocol, as this would cause tracking attacks [16].

Definition 1 ID-Transfer scheme [16]

The ID-transfer scheme of elliptic curve based randomized access control (EC-RAC) is shown in Fig. 1. In this scheme, a tag firstly chooses a random value $r_{t1} \in_R Z$, and then computes $T_1 = r_{t1}P$. After that, it sends T_1 to the reader. Then, the reader responds with a random challenge $r_{s1} \in_R Z$. Hence, the tag produces $T_2 = (r_{t1} + r_{s1}x_1)Y$ by using its own private key x_1 and sends the message T_2 to the reader. Upon receipt of message T_2, the reader sends it to the verifier in an secure network environment. The verifier calculates $x_1P(= X_1) = (y^{-1}T_2 - T_1)r_{s1}^{-1}$, which is used to check whether the corresponding tag is registered in the reader.

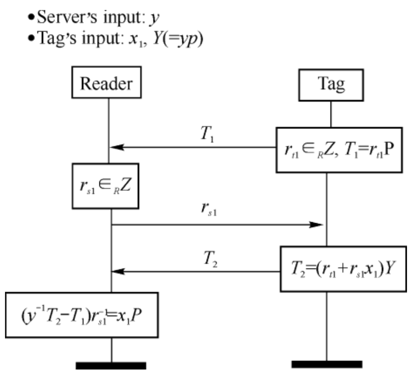

Fig. 1 ID-Transfer Scheme

Here, we recall the definition of the ECC-based grouping-proof protocol with colluding tag prevention. The main idea of ECC-based grouping-proof protocol is to intermingle runs of the ID-transfer protocol with multiple tags into a single grouping-proof protocol which denotes as the colluding tag prevention (CTP) protocol [4].

Definition 2 ECC-based grouping-proof protocol

The 2-party CTP protocol, which allows a couple of

RFID tags (denoted by tag *A* and *B*) to illustrate that they have been scanned simultaneously, is shown in Fig. 2. Note that we assume that the underlying communication protocol is able to detect and solve collisions. During the entire execution of the protocol, tags or the reader abort when a timeout occurs, when they receive the EC point at infinity, or when they receive an EC point with its x -coordinate equal to zero or the order of the point P on the curve. The protocol works as follows. The reader first sends the messages 'start left' and 'start right' to indicate the role of the tags in the protocol. Next, tag *A* chooses a random number r_a and computes the corresponding EC point $T_{a,1} = r_a P$. The reader generates $r_s \in_R Z$, and then forwards it to tag *B*. Upon receipt of $T_{a,1}$, tag *B* will first choose a random number r_b and compute the corresponding message $T_{b,1} = r_b P$. Next, it also computes the response $T_{b,2} = [r_b + x(r_s T_{a,1}) s_b]Y$ using its private key s_b, the random number r_b, the x-coordinate of the challenge $T_{a,1}$, and a random challenge r_s generated by the reader. Both $T_{b,1}$ and $T_{b,2}$ are handed down to the reader. In the next phase of the protocol, the reader forwards $T_{b,2}$ to tag *A*. This tag will compute the response $T_{a,2} = [r_a + x(T_{b,2}) s_a]Y$ using its private key s_a, the random number r_a and the x-coordinate of challenge $T_{b,2}$. The result is forwarded to the reader. The grouping proof, collected by the reader, consists of the following tuple: $(T_{a,1}, T_{a,2}, r_s, T_{b,1}, T_{b,2})$.

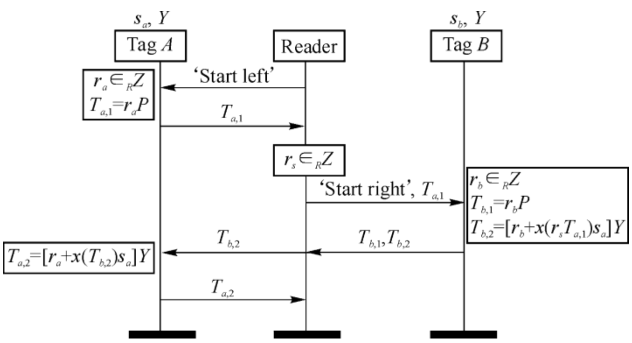

Fig. 2 Two-party grouping proof protocol with CTP

To verify the grouping-proof protocol constructed by tag *A* and *B*, the verifier will accomplish the following computations:

$$s_a P = (y^{-1} T_{a,2} - T_{a,1})[x(T_{b,2})]^{-1}$$

$$s_b P = (y^{-1} T_{b,2} - T_{b,1})[x(r_s T_{a,1})]^{-1}$$

Note that this does not require an exhaustive search through a database with all known public keys, but rather uses the verifier's secret key y to compute these public keys from the proof. If the public keys of tag *A* and *B* (S_a and S_b, respectively) are registered in the database of the verifier, the grouping proof is accepted and the timestamp is added. Otherwise, the proof is not accepted and the timestamp is not added [4]. The two-party CTP grouping-proof protocol can be easily extended to multipletags.

3 Compromised tag attack

A novel extended ECC-based RFID authentication protocol is shown, which can resist against compromised tag attack. However, we found that a compromised tag attack can be carried out on the adversarial model. We describe it as shown in Fig. 3.

The 2-party CTP protocol allows a pair of RFID tags (denoted by tag *A* and *B*) to prove that they have been scanned simultaneously. The protocol works as follows.

First of all, we describe a session (denote as S_1 in the Fig. 3): the reader sends the message 'start left' to the tag *A*. Upon receipt of the message 'start left' from the reader, tag *A* chooses a random value $r_a \in_R Z$, and then computes $T_{a,1} = r_a P$. After that it sends $T_{a,1}$ to the reader. Upon receipt of the message $T_{a,1}$, the reader chooses a random value $r_s \in_R Z$ and the corresponding EC point $T_{a,1}$. This message is then forwarded to Tag *B*. At this time, this session S_1 isn't completed. Then we start another session (denote as S_2 in the Fig. 3). An attacker can resend the set of messages $r_a \in_R Z$, $T_{a,1} = r_a P$, which has been received from S_1, to the reader. The reader chooses a random value $r_s' \in_R Z$ and the corresponding EC point $T_{a,1}$. At this time, the attacker corrupts the tag *B*, sets $r_b = x(r_s' T_{a,1}) s_b$, computes $T_{b,1} = r_b P$, and $T_{b,2} = [r_b + x(r_s T_{a,1}) s_b]Y$, then $T_{b,1}$, $T_{b,2}$ is forwarded to the reader. After that, the reader forwards the message $T_{b,2}$ to tag *A*. Upon receipt of the message $T_{b,2}$, tag *A* computes $T_{a,2} = [r_a + x(T_{b,2}) s_a]Y$ via the private key s_a. After that the message $T_{a,2}$ is sent to the reader. Later on, the verifier performs the following computations:

$$s_a P = (y^{-1} T_{a,2} - T_{a,1})[x(T_{b,2})]^{-1}$$

$$s_b P = (y^{-1} T_{b,2} - T_{b,1})[x(r_s T_{a,1})]^{-1}$$

[redacted]

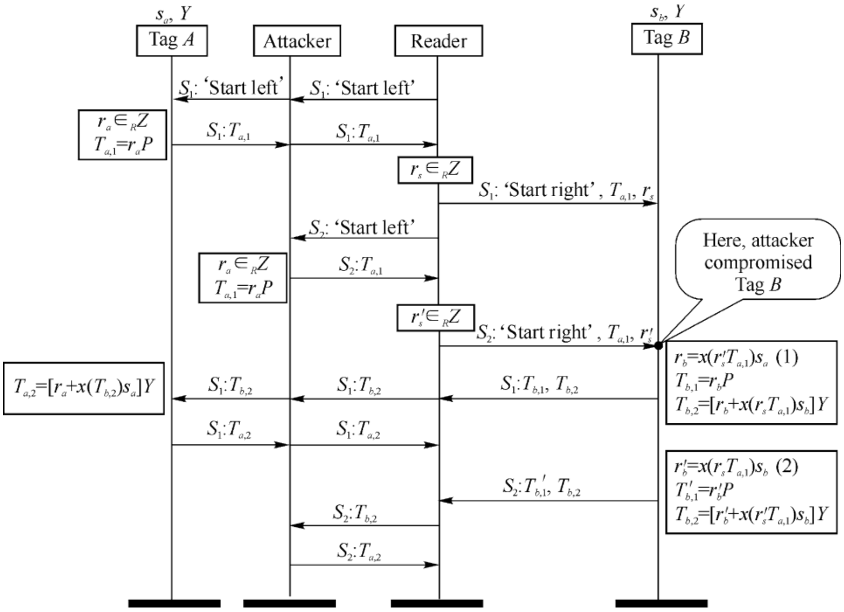

Fig. 3 Compromised Tag on the 2-party grouping proof protocol with CTP

Here we complete the S_1 by now.

Secondly, the attacker sets $r_b' = x(r_s T_{a,1})s_b$, $T_{b,1}' = r_b' P$ and $T_{b,2} = [r_b' + x(r_s' T_{a,1})s_b)Y]$, and then sends $T_{b,1}'$, $T_{b,2}$ to the reader. When the reader forwards the message $T_{b,2}$ to tag *A*, the attacker makes use of the message $T_{a,2}$, which sent in the S_1, to forward the message to the reader.

At last, the verifier performs the following computations:

$$s_a P = (y^{-1}T_{a,2} - T_{a,1})[x(T_{b,2})]^{-1}$$
$$s_b P = (y^{-1}T_{b,2} - T_{b,1})[x(r_s' T_{a,1})]^{-1}$$

By checking, the reader can accept the forged message which utilized the eavesdropped messages during the previous session in S_1. If the public keys of *A* and *B* are registered in the database of the reader, the grouping proof is accepted and a timestamp is added.

4 The novel secure grouping-proof protocol against compromised tag attack

Due to the compromised tag attack described in the Sect. 3, we construct a novel extended two-party grouping proof protocol with CTP which solve the weakness of the original ECC-based RFID authentication protocol. Firstly, we introduce the new scheme, which is shown in Fig. 4.

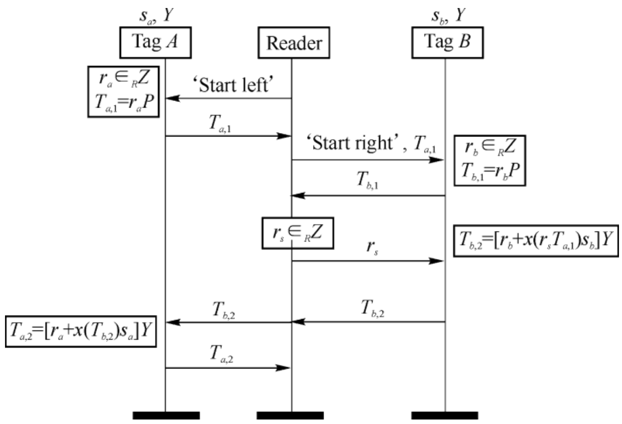

Fig. 4 The revised 2-party grouping-proof protocol against compromised tag attack

Scheme description

In the new scheme, the reader sends the message 'start left' to tag *A*. Upon receipt of the message 'start left' from the reader, tag *A* chooses a random value $r_a \in_R Z$, and computes $T_{a,1} = r_a P$, after that it sends $T_{a,1} = r_a P$ to the reader. Upon receipt of the message $T_{a,1}$, the reader forwards it and the message 'start right' to the tag *B*. When tag *B* receives the message from the reader, it chooses a random value $r_b \in_R Z$ and generates $T_{b,1} = r_b P$, then the message $T_{b,1}$ is sent back to the reader. Once receiving the message, the reader chooses a random value $r_s \in_R Z$ and sends the message r_s to the tag *B*. When receiving the message from the reader, tag *B* utilizes its private key s_b to compute $T_{b,2} = [r_b + x(r_s T_{a,1}) s_b]Y$. The message $T_{b,2}$ is sent to the reader, and the reader forwards the message to tag *A*. Once receipt of the message $T_{b,2}$, tag *A* will use its own private key s_a to compute $T_{a,2} = [r_a + x(T_{b,2}) s_b]Y$, and then send this $T_{a,2}$ to the reader. The verifier performs the following computations:

$$s_a P = (y^{-1}T_{a,2} - T_{a,1})[x(T_{b,2})]^{-1}$$

$$s_b P = (y^{-1}T_{b,2} - T_{b,1'})[x(r_{s'} T_{a,1})]^{-1}$$

and checks it whether correct or not.

Security analysis

Intuitively, we consider that the 2-party grouping proof protocol is secure.

It's trivial to find that the protocol shown in Fig. 2 does not consider resistance to compromised tag attack. As an illustration, we construct the scheme shown above which can avoid the defection of two-party grouping-proof protocol. Owing to the message $r_s \in_R Z$ generated by the reader between the message $T_{b,1}$ and $T_{b,2}$, and therefore, in the revised scheme, we can solve the problem of the compromised tag attack. More specifically, when the attacker computes the message r_b , the message r_s' must be fixed in prior based on Eq. (1) in Fig. 3. This leads that r_b' must have been set based on the specification of the revised scheme as in Eq. (2) in Fig. 3. Since r_b' is computed by r_s , the reader must have chosen the value $r_s \in_R Z$, which means that r_b has been fixed before. This paradox illustrates that the attack we found in Fig. 3 did not exist, thus our scheme avert this kind of attack.

Three-party grouping-proof isn't exactly the same as two-party grouping proof, we will illustrate it (constructed by tags *A*, *B* and *C*). In this case, the message $T_{b,2}$ is sent to tag *C* rather than the reader. Upon receipt of the message, tag *C* will choose a random value $r_c \in_R Z$ and perform the following two computations: $T_{c,1} = r_c P$, $T_{c,2} = [r_c + x(T_{b,2}) s_c]Y$. The outcome $T_{c,1}$, $T_{c,2}$ are sent back to the reader, which forwards $T_{c,2}$ to the tag *A*. Once tag *A* receipt of the message $T_{c,2}$, it uses its own private key s_a to compute $T_{a,2} = [r_a + x(T_{c,2}) s_b]Y$, and then sends this $T_{a,2}$ to the reader. For the sake of validating correctness, the verifier needs to compute the following computations:

$$s_a P = (y^{-1}T_{a,2} - T_{a,1})[x(T_{c,2})]^{-1}$$

$$s_b P = (y^{-1}T_{b,2} - T_{b,1})[x(r_s' T_{a,1})]^{-1}$$

$$s_c P = (y^{-1}T_{c,2} - T_{c,1})[x(T_{b,2})]^{-1}$$

By computing, the verifier can verdict whether the identifications of all the tags are the parties which the reader intend to communicate with or not. (See the rigorously prove farther below)

First of all, we are about to introduce the formalized definition of the grouping-proof protocol with CTP.

Definition 3 The implementation of grouping-proof protocol with CTP

To begin with, we will list the notations used in the following definitions. Since reader and verifier communications are through a secured channel, we will only discuss the transmittal messages among the reader and all the tags. The notations used in our definition are listed in Table 1.

Table 1 Notation used in the definition

Notations	Explanation for Notations
$i \in_R Z$	The number of the tags
p_i	The *i*th tag among all the tags
P	The base point in an elliptic curve
y	The trusted verifier's private key
$Y(=yP)$	The trusted verifier's public key
T	The point on the elliptic curve
$x(T)$	The *x*-coordinate of the point *T*
s_i	The *i*th tag's private key
$S_i(=s_i P)$	The *i*th tag's public key, which equals to the point multiplication operation S_i and the base point *P*

It's assumed that all tags are denoted as p_i , ($i \in_R Z$), which are logically ordered like a cycle. One point should note, each p_i cannot reveal its own public key during the execution of the protocol. Steps of protocol to perform in the formalized way are as follows:

Step 1 In round *i*, (*i*=1), the reader sends the message 'start left' to the tag p_i . Tag p_i chooses a random value $r_i \in_R Z$ and computes $T_{i,1} = r_i P$, and then sends $T_{i,1}$ back to the reader.

Step 2 In round *i*, (*i*=2), upon receipt of $T_{i-1,1}$, the reader sends the messages 'start right' and $T_{i-1,1}$ to the tag p_i . Once receipt of this message, p_i chooses a random value r_i and forwards $T_{i,1} = r_i P$ back to the reader. When receiving this message, the reader will choose a random value r_s and send it to tag p_i . Tag p_i utilizes its own

[redacted]

private key s_i to compute $T_{i,2}=[r_i+x(r_sT_{i-1,1})s_i]Y$ and forwards it to p_{i+1}.

Step 3 In round i, (i=3,…, $n-1$), p_i chooses a random value r_i and computes $T_{i,1}=r_i P$, and then utilizes its own private key s_i to compute $T_{i,2}=[r_i+x(T_{i-1,2})s_i]Y$, after that forwards it to tag p_{i+1}.

Step 4 In round i, (i=n), p_i chooses a random value r_i and forwards the message $T_{i,1}=r_iP$ and $T_{i,2}=[r_i+x(T_{i-1,2})s_{i-1}]Y$ to the reader. The reader receives these two messages, and then forwards $T_{i,2}$ to tag p_{n-i+1}. Tag p_{n-i+1} utilizes its own private key s_{n-i+1} to compute the message $T_{n-i+1,2}=[r_{n-i+1}+x(T_{i,2})s_{n-i+1}]Y$.

The verifier will check the correctness of the grouping-proof protocol by the following computations:

$$s_1P=(y^{-1}T_{1,2}-T_{1,1})[x(T_{i,2})]^{-1}$$

$$s_2P=(y^{-1}T_{2,2}-T_{2,1})[x(r_sT_{1,1})]^{-1}$$

$$s_iP=(y^{-1}T_{i,2}-T_{i,1})px(T_{i-1,2})]^{-1}$$

where $i=3,\dots,n$.

By computing, the verifier can verdict whether the identifications of all the tags are the participants that the reader intends to communicate with or not.

Secondly, we are about to prove that the 3-party grouping-proof protocol is secure.

Lemma 1 3-party grouping-proof protocol is secure.

Proof We analyze this 3-party grouping-proof protocol via the tool of ProVerif which utilizes the symbolic analysis approach. The specific description of algorithm named 3-party grouping-proof security (3-GPS) algorithm is proposed as follows:

Algorithm 3-GPS algorithm

```
(* Secure channels *)
free car.
free cbr.
free ccr.
free cbc.
(* Free variables *)
private free s_a.
private free s_b.
private free s_c.
private free y.
free S_a, S_b, S_c, Y, P, T_b2.
(* Active adversary *)
param attacker = active.
(* Inverse function *)
fun inver/1.
equation inver(inver(x)) = x.
(* Add function *)
fun add/2.
(* Sub function *)
fun sub/2.
(* Multiply function *)
fun multi/2.
(* Get x-value function *)
fun getx/1.
(* Reduction equation *)
reduc T1(x, P) = multi(x, P).
reduc T2(x, y, z, w) = multi(add(z, multi(getx(x), y)), w).
(* The process *)
let p_a = (new r_a; in(car, m0);
let T_a1 = T1(r_a, P) in out(car, T_a1); in(car, m1);
let T_a2 = T2(m1, S_a, r_a, Y) in out(car, T_a2)).
let p_b = (in(cbr, m2); new r_b;
let T_b1 = T1(r_b, P) in out(cbr, T_b1); in(cbr, m3);
let T_b2 = T2(multi(m3, m2), S_b, r_b, Y) in out(cbc, T_b2)).
let p_c = (new r_c; in(cbc, m4);
let T_c1 = T1(r_c, P) in
let T_c2 = T2(m4, S_c, r_c, Y) in out(ccr, (T_c1, T_c2))).
(new left; new right;out(car, left); in(car, m5);
let T_a1 = m5 in out(cbr, T_a1);in(cbr,m6);let T_b1 = m6 in;
new r_s; out(cbr, r_s);
let (T_c1, T_c2) = m7 in
in(ccr, m7); out(car, T_c2); in(car, m8);
let T_a2 = m8 in
out(car, choice[sP(T_c2, y, T_a2, T_a1),
multi(sub(multi(inver(y), T_a2),T_a1),
inver(getx(T_c2)))]);
out(cbr, choice[sP(multi(r_s, T_a1),y, T_b2, T_b1),
multi(sub(multi(inver(y), T_b2),T_b1),
inver(getx(multi(r_s, T_a1))))]);
out(ccr, choice[sP(T_b2, y, T_c2, T_c1),
multi(sub(multi(inver(y), T_c2),T_c1),
inver(getx(x)))])).
```

In this algorithm, we denote processes p_a, p_b, p_c and p_r as participants a, b, c and the reader r respectively. Each process can compute data, as well as sends and receives messages with others. We utilize operation choice (x, y) to distinguish term x and y. If the result of choice is that observational equivalence is true, i.e. x and y is observational indistinguishable, then we can

[redacted]

illustrate that the protocol is secure. The result of the specific implementation is shown as follows.

-- Observational equivalence

Termination warning: v_282 <> v_283 & attacker2:v_281, v_282 & attacker2:v_281,v_283 -> bad:

Selecting 0

Termination warning: v_285 <> v_286 & attacker2:v_285, v_284 & attacker2:v_286,v_284 -> bad:

Selecting 0

Completing...

Termination warning: v_282 <> v_283 & attacker2:v_281, v_282 & attacker2:v_281,v_283 -> bad:

Selecting 0

Termination warning: v_285 <> v_286 & attacker2:v_285,v_284 & attacker2:v_286,v_284 -> bad:

Selecting 0

200 rules inserted. The rule base contains 200 rules. 32 rules in the queue.

RESULT Observational equivalence is true (bad not derivable).

It implies that the 3-party grouping-proof protocol is secure. In other words, the output of this protocol success indicates that the result observational equivalence is true.

Theorem 1 *n*-party (*n*>2) grouping-proof protocol is a secure protocol.

Proof By mathematical inductive method, we firstly prove that 3-party grouping proof protocol is secure. Secondly, assume that *k*-party grouping proof protocol is secure, then ($k+1$)-party grouping-proof protocol is also secure.

3-party grouping-proof protocol has been proved in Lemma 1, thus we just need to prove the security of *k*-party grouping proof protocol. First of all, we utilize k tags' protocol π_k to simulate $k+1$ tags' protocol π_{k+1}. Assume to the contrary, if there is a $k+1$ tags' attacker Œ_{k+1} against π_{k+1}, then there must exist k tags' attacker Œ_k which against the protocol $\bar{\pi}_k$.

Out of this attacker Œ_k, we will run of Œ_{k+1} by Œ_k. The simple sketch will be given as follows.

1) Attacker Œ_k performs protocol π_k, and it will obtain all the messages before tag A receives message which tag K sends. At this point, Œ_k make the protocol temporarily halt.

2) Attacker Œ_k simulates the ($k+1$)th tag which participate in the protocol π_{k+1} and generates the private key s_{k+1}.

3) Based on the private key, random values and the transfer message which tag *A*, *B*,…,*K* generate in the actual protocol, attacker Œ_k creates a random value $r_{k+1} = r_k + x(T_{k-1,2})s_k - x(T_{k,2})s_{k+1}$.

4) Based on the messages r_{k+1} r_{k+1} and $T_{k,2}$, Œ_k utilizes its own private key s_{k+1} to compute the message $T_{k+1,2} = [r_{k+1} + x(T_{k,2})s_d]Y$. In other words, Œ_k runs of the attacker Œ_{k+1} to simulate the protocol π_{k+1}. In line with the messages from step 2 and 3, Œ_k responds the messages which Œ_{k+1} should receive in the process during the execution of protocol π_{k+1}. After that, attacker Œ_k sends the message $T_{k+1,2}$ which equals to the message $T_{k,2}$ to tag A_1. Upon receipt of the message $T_{k+1,2}$, tag A_1 takes the message $T_{k+1,2}$ as the message $T_{k,2}$ to compute the value $T_{1,2}$, and then it sends this message to the reader. The verifier could pass the validation.

Next, we detail the specific process of the simulation.

Assume that the order of the tags is exactly like p_1, p_2 ,…, p_i ($\{p_i \mid i \in Z\}$) during the process of grouping-protocol performance. Values s_i and r_i are the tag p_i's private key and random number pair. Let l be an upper bound on the number of these kinds of protocol which might be executed.

(a) Choose $r \xleftarrow{R} \{1 \cdots l\}$.

(b) Invoke adversary Œ_i, running the protocol interaction with parties $p_i, (i \in Z_n)$, just to obtain all the messages before tag p_1 receives message which tag p_i sends. At this point, Œ_i let the protocol temporarily halt.

(c) Adversary Œ_{i+1} runs of the protocol with parties $p_1, p_2, \ldots, p_{i+1}$ normally. Invoke Œ_i to simulate the ($i+1$)th tag p_{k+1}, set $r_{i+1} = r_i + x(T_{i-1,2})s_i - x(T_{i,2})s_{i+1}$, for $i \geq 3$, $i \in Z$. s_{i+1} is the ($i+1$)th tag's private key which generate by its own. Upon receipt of the message $T_{i+1,2} = [r_{i+1} + x(T_{i,2})s_{i+1}]Y$, Œ_{i+1} forwards it to Œ_i, at this point, let Œ_i send this message to tag p_1, which was halted in the step 2. Then, we make step 2 continue to run.

(d) Once receipt of the message $T_{i+1,2}$, tag p_1 takes this message as the message which tag p_1 sends. Out of $T_{i+1,2}$ equals to $T_{i,2}$, tag p_1 utilizes its own private key s_1 to compute $T_{1,2} = [r_1 + x(T_{i+1,2})s_1]Y$, and then sends it to the reader. The verifier could pass the validation.

[redacted]

In this case, let q be a security parameter. We assume that the probability which attacker $Œ_{i+1}$ win the game with non-negligible probability. We have that:

$$\Pr[Œ_i = 1, q] = \frac{1}{P_n^{i+1}(q)} \cdot \frac{1}{l(q)} \cdot \varepsilon(q)$$

P_n^{i+1} denotes the probability of polynomial time for selecting $i+1$ tags from n tags, among which the order of the $i+1$ tags is fixed. And P_n^{i+1} is the polynomial of security parameter q. Let l be an upper bound on the number of these kinds of protocols which might be executed. And l is the polynomial concerning to security parameter q. $\varepsilon(q)$ denotes non-negligible probability with security parameter q. And therefore, the probability which $Œ_i$ win this game is still a non-negligible $\varepsilon(q)$ with respect to the security parameter q.

In conclusion, i-party grouping-proof protocol is secure, then (i+1)-party grouping proof protocol is also secure.

5 Performance analysis

First of all, we compare our proposed scheme with other three schemes in terms of the following aspects: resistance to replay attack (RA), resistance to denial of service (DoS) attack, resistance to man-in-middle (MiM) attack and resistance to compromised tag attack (CT). In Table 2, we use the notation '✓' to denote functions achieved; and use the notation '×' to denote function not achieved. We obtain other protocols' assumptions and their weaknesses from their researchers' evaluation in their papers. Table 3 depicts that our protocol has the highest security than others, and it illustrates that our protocol can secure against all the RFID attacks which we mentioned above.

Table 2 Comparison with other protocols

	RA	DoS	MiM	CT
BATINA [4]	✓	✓	✓	×
SOTP [17]	✓	✓	×	×
SMGOTP [15]	✓	✓	×	×
OUR WORK	✓	✓	✓	✓

Next, we analyze the computation and communication loads of our protocol, and demonstrate that our scheme is feasible for the lightweight passive RFID tags. Hence, we compare our protocol with other three protocols and show their performances when transferring n tags in a group. The depicted and analyzed data of the three protocols are shown in the Table 3. T_{ran} denotes the time of choosing a random value; T_{ecc} denotes the time for elliptic curve operation; T_{lw} denotes the time for one lightweight en/decryption; T_{en} denotes the time for one en/decryption; T_{rev} denotes the time for reversible operations. Since addition and subtraction operations require only a little computation, compared with multiplication and reversible operations, we leave them out in our performance analysis.

Table 3 Computation loads and communication loads when transferring n tags

Schemes	Devices	Computation loads	Communication loads
BATINA [4]	Tag	$nT_{\text{ran}} + (2n+1)T_{\text{ecc}}$	$n+4$
	Reader	T_{ran}	
	Verifer	$(2n+1)T_{\text{ecc}} + T_{\text{ran}} + 2nT_{\text{rev}}$	
SOTP [17]	Tag	$3nT_{\text{lw}} + 2nT_{\text{ran}}$	$11n$
	Reader	$3nT_{\text{en}}$	
	Verifer	$9nT_{\text{en}} + 3nT_{\text{lw}} + 2nT_{\text{ran}}$	
SMGOTP [15]	Tag	$5nT_{\text{lw}}$	$2n+7$
	Reader	$(n+1)T_{\text{en}} + T_{\text{ran}}$	
	Verifer	$(n+9)T_{\text{en}} + (3n+2)T_{\text{lw}} + T_{\text{ran}}$	
OUR WORK	Tag	$nT_{\text{ran}} + (2n+1)T_{\text{ecc}}$	$n+6$
	Reader	T_{ran}	
	Verifer	$(2n+1)T_{\text{ecc}} + T_{\text{ran}} + 2nT_{\text{rev}}$	

We implement a simulation to display our performance comparison with other relevant schemes SOTP [17], SMGOTP [15] and Batina and Lee [4]. We mainly discuss three parts: the computation loads on tag's part, the computation loads on the reader's part and the communication loads on the reader's part, are shown visually in Figs. 5–7.

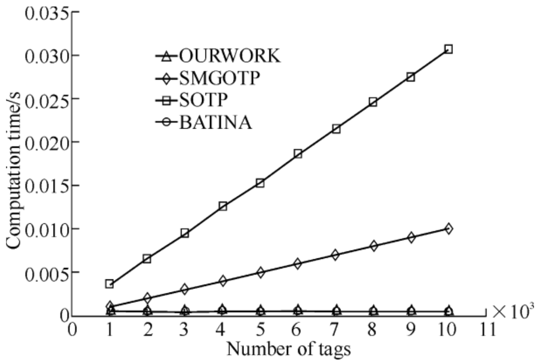

Fig. 5 Comparison of computation on reader's part

In Fig. 5, we compare the computation loads with four protocols on reader's part. Our protocol has the lowest computation time than the scheme SOTP [17] and SMGOTP [15], because the reader just needs to choose one random value during the whole protocol with all tags. We have almost the same computation loads as the scheme BATINA [4] in this part, and our scheme requires 50.0% computation loads of SOTP's scheme and 14.3%

[redacted]

computation loads of SMGOTP's scheme.

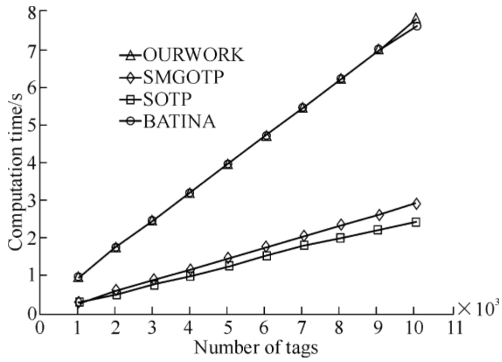

Fig. 6 Comparison of computation on Tags' part

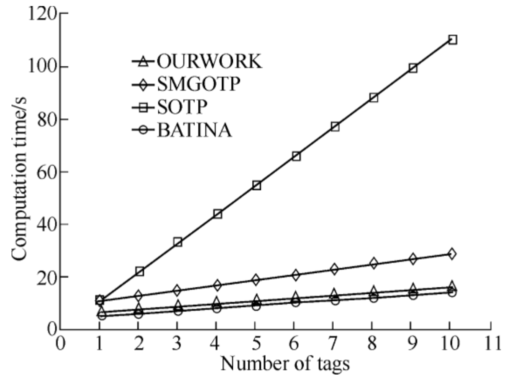

Fig. 7 Comparison of communication on Tags' part

We compare the computations with four protocols on tags' part. SMGOTP scheme requires three lightweight encryption algorithms, nevertheless our work and BATINA's need elliptic curve algorithms, see Fig. 6. Although our work has much higher computation time than their works, it's still acceptable. The implementation of our work is done with the thousands of tags. In addition, we have much higher security than other three schemes, is shown in Table 2.

We compare the communication loads with other three protocols on the tags' part, see shown in Fig. 7. The figure provide an illustration of communication loads, our scheme is clearly lower than the scheme SOTP and SMGOTP and have a slightly higher value than BATINA's. In this part, our scheme requires 14.5% communication loads of SOTP's, 55.1% communication loads of SMGOTP's and requires 14.3% higher communication loads than BATINA's.

6 Conclusions

A large number of RFID protocols which tags could be scanned simultaneously have been proposed in recent years. Most of the protocols take man-in-the-middle attack into consideration, but they have not considered for the compromised tag attack. In this article, we find a kind of compromised tag attack in the two-party grouping proof protocols which compromise security. In view of this weakness, we propose a novel secure RFID authentication protocol which can avert this drawback. By formal analysis, provable security, and mathematical inductive method, we prove that our grouping-proof protocol with n-party ($n \geq 3$) is secure and can resist against compromised tag attack. Finally, we provide some experimental analysis of the communication and computation loads. By analyzing, we found that our protocol provide a lower communication loads on the tags' part and has a lower computation loads on the reader's part than the protocols (SOTP and SMGOTP) when transferring a large number of tags. Although, our work has a much higher computation time than their works, it's still acceptable, because our protocol provide even more security than theirs.

References

1. Peris-Lopez P, Orfila A, Hernandez-Castro J, et al. Flaws on RFID grouping-proofs. Journal of Network and Computer Applications, 2011, 34(3): 833–845
2. Park C, Hur J, Hwang S, et al. Authenticated public key broadcast encryption scheme secure against insiders. Journal of Mathematical and Computer Modeling, 2012, 55(1/2): 113–122
3. Sadighian A, Jalili R. AFMAP: anonymous forward-secure mutual authentication protocols for RFID systems. Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'09), Jun 18–23, 2009, Athens, Greece. Piscataway, NJ, USA: IEEE, 2009: 31–36
4. Batina L, Lee Y, Seys S, et al. Extending ECC-based RFID authentication protocols to privacy-preserving multi-party grouping proofs. Personal and Ubiquitous Computing, 2012, 16(3): 323–335
5. Wong C, Gouda M, Lam, S. Secure group communications using key graphs. Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'98), Sep 2–4, 1998, Vancouver, Canada. New York, NY, USA: ACM, 1998: 68–79
6. Ma C, Lin J, Wang Y, et al. Offline RFID grouping proofs with trusted timestamps. Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12), Jun 25–27, 2012, Liverpool, UK. Los Alamitos, CA, USA: IEEE Computer Society, 2012: 674–681

7. Lin Q, Zhang F. ECC-based grouping-proof RFID for inpatient medication safety. Journal of Medical Systems, 2012, 36(6): 3527–3531
8. Dang D N, Kim K. On the security of RFID group scanning protocols. IEICE Transactions on Information and Communication Systems, 2010, 93D(3): 528–530
9. Leng X, Lien Y, Mayes K, et al. Select-response grouping proof for RFID tags. Proceedings of the 1st Asian Conference on Intelligent Information and Database Systems (ACIIDS'09), Apr 1–3, Dong Hoi, Vietnam. Piscataway, NJ, USA: IEEE, 2009: 73–77
10. Burmester M, Medeiros B, Motta R. Provably secure grouping-proofs for RFID tags. Proceeding of the 8th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Applications (CARDIS'08), Sep 8–11, 2008, London, UK. Berlin, Germany: Springer-Verlag, 2008: 176–190
11. Nair R, Perret E, Tedjini S. Chipless RFID based on group delay encoding. Proceedings of the 2011 IEEE International Conference on RFID-Technologies and Applications (RFID-TA'11), Sep 15–16 2011, Sitges, Spain. Piscataway, NJ, USA: IEEE, 2011: 214–218
12. Mandel C, Kubina B, Schussler M, et al. Group-delay modulation with metamaterial-inspired coding particles for passive chipless RFID. Proceedings of the 2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA), Nov 5–7, Nice, France. Piscataway, NJ, USA: IEEE, 2012: 5p
13. Sato Y, Mitsugi J, Nakamura O, et al. Theory and performance evaluation of group coding of RFID tags. IEEE Transactions on Automation Science and Engineering, 2012, 9(3): 458–466
14. Cucchiara R, Haider R, Mandreoli R, et al. Identification of intruders in groups of people using cameras and RFIDs. Proceedings of the 5th ACM/IEEE International Conference on Distributed SmartCameras (ICDSC'11), Aug 22–25, 2011, Ghent, Belgium. Piscataway, NJ, USA: IEEE, 2011: 6p
15. Yang M H. Secure multiple group ownership transfer protocol for mobile RFID. Electronic Commerce Research and Applications, 2012, 11(4): 361–373
16. Lee Y, Batina L, Singelee D, et al. Low-cost untraceable authentication protocols for RFID. Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec'10), Mar 22–24, Hoboken, NJ, USA. New York, NY, USA: ACM, 2010: 55–64
17. Yang M H, Hu H Y. Protocol for ownership transfer across authorities: with the ability to assign transfer target. Journal of Security and Communication Networks, 2012, 5(2): 164–177

14. Newman M, Leskovec J, Lang K J, et al. Finding community structure in networks using the eigenvectors of matrices. Physical Review E, 2006, 74(3): 15–31
15. Freeman L C. A set of measures of centrality based on betweenness. Sociometry, 1977, 40(1): 41–50
16. Zhang Z S. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. IEEE Communications Surveys Tutorials, 2006, 8(1): 24–37
17. Spyropoulos T, Psounis K, Raghavendra C S, et al. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. Proceedings of the ACM SIGCOMM Workshop on Delay Tolerant Networking (WDTN'05), Aug 22–26, 2005, Philadelphia, PA, USA. New York, NY, USA: ACM, 2005: 252–259
18. Yoneki E. Visualizing communities and centralities from encounter traces. Proceedings of the ACM MobiCom Workshop on Challenged Networks (CHANTS'08), Sep 14–19, 2008, San Francisco, CA, USA. New York, NY, USA: ACM, 2008: 25–27
19. Hui P, Yoneki E, Chan S, et al. Distributed community detection in delay tolerant networks. Proceedings of the 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture (MobiArch'07),
20. Conan V, Leguay J, Friedman T, et al. Characterizing pairwise inter-contact patterns in delay tolerant networks. Proceedings of the 1st Conference on Autonomic Computing and Communication Systems (AUTONOMICS'07), Oct 28–30, 2007, Rome, Italy. 2007: 9p
21. Banerjee N, Corner M D, Towsley D, et al. Relays, base stations, and meshes: Enhancing mobile networks with infrastructure. Proceedings of the 14th Annual International Conference on Mobile Computing and Communications (MOBICOM'08), Sep 14–19, 2008, San Francisco, CA, USA. New York, NY, USA: ACM, 2008, 81–91
22. Palla G, Derenyi I., Farkas I., et al. Uncovering the overlapping community structure of complex networks in nature and society. Nature, 2005, 435: 814–818
23. Li Y, Gao Y, Li S, et al. Integrating forwarding and replication in DTN routing: a social network perspective. Proceedings of the 73rd Vehicular Technology Conference (VTC-Spring'11), May 15–18, 2011, Budapest, Hungary. Piscataway, NJ, USA: IEEE, 2011: 5p
24. Hui P, Yoneki P, Chan S, et al. Distributed community detection in delay tolerant networks. Proceedings of the 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture (MobiArch'07), Aug 27–31, 2007, Kyoto, Japan. New York, NY, USA: ACM, 2007: 8p

Aug 27–31, 2007, Kyoto, Japan. New York, NY, USA: ACM, 2007: 8p

6. Baracca P, Boccardi F, Braun V. A dynamic joint clustering scheduling algorithm for downlink CoMP systems with limited CSI. Proceedings of the 9th International Symposium on Wireless Communication Systems (ISWCS'12), Aug 28–31, 2012, Paris, France. Piscataway, NJ, USA: IEEE, 2012: 830–834
7. Gong J, Zhou S, Niu Z S, et al. Joint scheduling and dynamic clustering in downlink cellular networks. Proceedings of the IEEE Global Communications Conference (GLOBECOM'11), Dec 5–9, 2011, Houston, TX, USA. Piscataway, NJ, USA: IEEE, 2011: 2011: 5p
8. Sadek M, Tarighat A, Sayed A H. A leakage-based precoding scheme for downlink multi-user MIMO channels. IEEE Transactions on Wireless Communications, 2007, 6(5): 1711–1721
9. Zhao J, Lei Z D. Clustering methods for base station cooperation. Proceedings of the Wireless Communications and Networking Conference (WCNC'12), Apr 1–4, 2012, Paris, France. Piscataway, NJ, USA: IEEE, 2012: 946–951
10. Abdelaal R A, Ismail M H, Elsayed K. Resource allocation strategies based on the signal-to-leakage-plus-noise ratio in LTE-A CoMP systems. Proceedings of the Wireless Communications and Networking Conference (WCNC'12), Apr 1–4, 2012, Paris, France. Piscataway, NJ, USA: IEEE, 2012: 1590–1595

[redacted]