International Journal of Computational Intelligence Systems, Vol. 5, No. 5 (September, 2012), 824-833

A Forward-secure Grouping-proof Protocol for Multiple RFID Tags

LIU Ya-li
College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics
Nanjing, 210016, China
College of Computer Science & Technology, Jiangsu Normal University
Xuzhou, 221116, China

QIN Xiao-lin*, LI Bo-han, LIU Liang
College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics
Nanjing, 210016, China

Received 1 November 2011
Accepted 15 June 2012

Abstract

Designing secure and robust grouping-proof protocols based on RFID characteristics becomes a hotspot in the research of security in Internet of Things (IOT). The proposed grouping-proof protocols recently have security and/or privacy omission and these schemes afford order-dependence by relaying message among tags through an RFID reader. In consequence, aiming at enhancing the robustness, improving scalability, reducing the computation costs on resource-constrained devices, and meanwhile combing Computational Intelligence (CI) with Secure Multi-party Communication (SMC), a Forward-Secure Grouping-Proof Protocol (FSGP) for multiple RFID tags based on Shamir's (*n*, *n*) secret sharing is proposed. In comparison with the previous grouping-proof protocols, FSGP has the characteristics of forward-security and order-independence addressing the scalability issue by avoiding relaying message. Our protocol provides security enhancement, performance improvement, and meanwhile controls the computation cost, which equilibrates both security and low cost requirements for RFID tags.

Keywords: RFID; Grouping-proof; Forward-secure; Order-independent; Secret Sharing

1. Introduction

With the wide spread of RFID tags and its cheap implementations, the need for providing secure and privacy-preserving authentication protocols in extremely resource-constrained environments is evident. Ari Juels first introduced yoking-proof[1], which involves generating evidence of the simultaneous presence of two tags in the range of an RFID reader. The proof can then be verified by a verifier which holds all the secret keys of tags. Then he extended this notion and envisioned the concept of grouping-proofs[2-4], which allows multiple RFID tags to provide evidence that they are scanned simultaneously in an identification session by one or more readers within its broadcast range. Other improved variants of yoking-proof were also proposed in[3,6,7]. As Juels[1] already pointed out, there are several practical scenarios where grouping-proofs could significantly expand the capabilities of RFID-based systems, such as manufacturing, supply chains, access control, e-ticketing and counterfeit prevention, *etc*. Motivated by the potential applications, several grouping-proofs for RFID tags are developed in recent years[2-8].

* Corresponding author: qinxcs@nuaa.edu.cn

RFID tags are severely constrained in terms of storage resources, computational capabilities and power supply, and therefore the protocols that involve high computational and storage burdens are not attractive. Computational intelligence (CI) has been successfully used in recent years to address various challenges such as data aggregation, security, optimal deployment and localization, which brings about broad applicability, flexibility, self optimization capability and robustness against malicious attacks in dynamic environments. CI is an area of fundamental and applied research[9,10] involving numerical information processing in contrast to the symbolic information processing techniques of artificial intelligence (AI), which is defined as[11] the computational models and tools of intelligence capable of inputting raw numerical sensory data directly, processing them by exploiting the representational parallelism and pipelining the problems, generating reliable and timely responses and withstanding high fault tolerance. CI studies adaptive mechanisms that enable or facilitate intelligent behavior in complex and changing environments[12,13], which encompasses neural networks, genetic algorithms, reinforcement learning, swarm intelligence, evolutionary algorithms, fuzzy logic and artificial immune systems, *etc*.

Different from the common techniques of CI that is addressed above, we focus on the applications of CI into this rapidly growing area of grouping-proof protocols for RFID tags by combing CI with Secure Multi-party Communication (SMC). In this paper we first evaluate these proposed grouping-proofs[1-8] recently to observe security demand and analyze security weakness. Then we propose a lightweight forward-secure grouping-proof protocol based on Shamir's (n, n) secret sharing to improve scalability, robustness, especially order-independence by avoiding relaying message. In comparison with the previous grouping-proofs, our contributions can be summarized as follows:

(i) Guarantees session unlinkability with forward security by auto-update mechanism for secret and state information within a tag.
(ii) Ensures the protocol order-independent by avoiding relaying message through RFID reader using Shamir's (n, n) secret sharing.
(iii) Addresses the scalability issue which makes a single authentication protocol in combination with a grouping-proof protocol properly by controlling round-trip time of a challenge-response cycle and applying the technique of CI properly.
(iv) Enhances the robustness, which makes the protocol thwart man-in-the-middle attack, replay attack, counterfeit attack in a formal security framework and meanwhile possess tag anonymity and untraceability.
(v) Meets the requirement of lightweight on resource-constrained devices by only using MAC and PRNG operation on RFID tags.

The remainder of this paper is organized as follows. We present a critical review of the related work in Section 2. In Section 3 we then review some preliminaries briefly. Next our forward-secure grouping-proof protocol (FSGP) is described in Section 4. The Section 5 addresses the presentation of security and performance analysis. Finally, Section 6 concludes this paper.

2. Related Works: RFID Grouping-Proofs

2.1. *Review of existing protocols*

The idea of grouping proofs originated from Juels[1] in 2004. His proposal for this type of identification protocol, so-called yoking proof, relies on interleaving MACs of two tags using a reader as a communication medium and utilizing a timeout mechanism to guarantee the validity of yoking proof generated at each session.

(1) Yoking-proof[1] attack and improvements

Nevertheless, Saito & Sakurai [2] were the first to point out the weaknesses in the work of Juels[1]. They indicated that yoking-proof is not immune to replay attacks. Yoking-proof has been extended to prove simultaneous presence of a group of tags in the range of an RFID reader in Saito & Sakurai[2]. They called this kind of proof a grouping-proofs. Burmester *et al.*[6] pointed out two additional weaknesses in Saito & Sakurai[2]: Denial-of-Service (DOS) and impersonation attacks. In addition, in 2006 Piramuthu[3] showed Saito's protocol[2] with timestamps is also vulnerable to replay attack. Accordingly, he proposed another variant of yoking-proof which does not use timestamp to prevent replay attack. But Piramuthu[3] did not resolve security threats such as privacy disclosure, forward secrecy divulgence, authentication sequence disorder and DOP attack.

(2) Anonymous grouping-proof schemes

The idea of anonymous grouping-proofs was first introduced by Bolotnyy and Robins[4] in 2006, which accommodated a group of RFID tags by extending Juels' yoking proof, so-called Generalized Yoking-proofs. Unfortunately, this proposed Anonymous Yoking scheme suffers from forward secrecy disclosure and tag privacy divulgence. In 2007 Peris-Lopez *et al*.[5] discovered Piramuthu[3] that it can't resist tag-tracking attack and replay attack, which is a variant of counterfeit proof attack. To solve these security threats, Peris-Lopez *et al*. developed a clumping-proof[5] which is privacy-preserving of anonymous. However, this protocol is not reliable when defending against DOP attack, forward secrecy disclosure and authentication sequence disorder. In 2008 Burmester *et al*.[6] pointed out weaknesses in Bolotnyy's scheme[4] and presented a security model based on the Universal Composability framework. In 2009 Chien and Liu[14] proposed an anonymous tree based yoking protocols. However, this protocol is vulnerable to malicious tracking.

(3) Order-independent grouping-proof schemes

The order-independent grouping-proof was introduced by Lien *et al*.[8], which had resolved several security pitfalls in Piramuthu's protocol[3] such as reading order dependence and authentication sequence disorder. Lin *et al*.[7] pointed out that Piramuthu's protocol[3] suffers from interference problem when multiple readers are represented. To counter the problem, Lin *et al*. proposed both online and offline grouping-proof protocols[7] which are variants of timestamp-based yoking proof to avoid race conditions. In both schemes tag anonymity and forward security cannot be guaranteed and they cannot defend against DOP attack. Chien *et al*.[15] also proposed an offline grouping-proof protocol. Unfortunately it is vulnerable to replay attacks.

2.2. *Common weakness of existing RFID Grouping-proofs*

In the previous section, we show the analysis to recent schemes[1-8]. Then we summarize the common weakness of those schemes as follows.

(1) Afford order-dependence by relaying message among tags transferring through an RFID reader.
(2) Reduce the efficiency, improve failure rates and especially reject verification in one proof session.
(3) Suffer from the weak scalability issue which isolates a single authentication from a grouping-proof.
(4) Address weak secure and privacy properties, which sometimes cannot defend against malicious attacks and do not have unlinkability.

3. Preliminary

3.1. *RFID Deployments and Assumption*[6]

A typical deployment of an RFID system involves three types of legitimate entities: Tags, Readers and a Verifier (or a Backend Server). Throughout this paper, we assume the following characteristic of grouping-proof:

(1) The tags are passive and have very limited computation and communication capabilities. It is common assumed that they are able to perform basic cryptographic operations such as generating pseudo-random numbers and evaluating pseudo-random functions. The tags do not maintain clocks while the verifier controls a challenge-response cycle.
(2) The readers establish communication channels that link the tags to manage the interrogation of tags and keep a record of proofs for each session which cannot be manipulated by the adversary.
(3) The verifier is the only trusted entity that may share some secret information with the tags such as cryptographic keys. The verifier has a secure channel that links to the readers. In contrast, the channels between tags and the reader are considered insecure.
(4) A qualified RFID grouping-proof protocol should comply with several essential security and privacy requirements[16], such as data confidentiality, tag anonymity, forward security, defending against malicious attacks and untraceability.
(5) RFID grouping-proof protocols are mainly concerned with security issues[16] at the protocol layer and not with physical or link layer issues.

3.2. *Shamir's* (*t, n*)*-SS* [17]

Secret sharing schemes were originally introduced by Blakley[18] and Shamir[19] independently as a solution for safeguarding secret keys. Shamir's (*t, n*) secret sharing is denoted as (*t, n*)-SS. In particular, (*t, n*)-SS is called (*n, n*)-SS when $t = n$. (*t, n*)-SS is based on Lagrange interpolating polynomial and is information-theoretically secure without any computational assumption. (*t, n*)-SS consists of two algorithms:

(1) **Share generation algorithm**: The mutually trusted dealer *D* first selects a random polynomial $f(x)$ of

degree t-1: $f(x) = a_0+a_1x+\ldots a_{t-1}x^{t-1}$, such that s = a_0 and all coefficients a_0, a_1, … , a_{t-1} are in a finite field F_p = $GF(p)$ with p elements. D computes n shares (s_1, s_2,…, s_n) as $s_1 = f(1)$, $s_2 = f(2)$,…, $s_n = f(n)$. D distributes each share s_i to corresponding shareholder P_i secretly.

(2) **Secret reconstruction algorithm**: For any t share $(s_{i1}, s_{i2},..., s_{it})$ where $(i_1, i_2,..., i_t) \subset \{1,2,...,n\}$, the secrets can be reconstructed using Lagrange interpolating formula.

4. Our Protocol FSGP

To enhance the robustness, reduce the computation costs and avoid order-dependent, we propose a forward-secure grouping-proof protocol for multiple RFID tags based on Shamir's (n, n)-SS, which is the new application of combing CI with SMC to construct Grouping-proof protocol. FSGP addresses the scalability issue properly by avoiding relaying message among tags through RFID reader and realizes the direct challenge-response among readers and tags. Moreover the protocol guarantees session unlinkability by adding forward security. Our protocol FSGP sets a single authentication proof being a typical example of a grouping-proof properly in one challenge-response session by applying the technique of CI properly. In addition, FSGP is light weight by reducing the computation costs on resource-constrained devices because tags operations are limited to the invocation of a PRNG function and MAC operation.

4.1. *Notations*

We use the notations base on Juel [1] for entities and operations as summarized in Table 1 to simplify description.

4.2. *Forward-Secure Grouping-Proof* (*FSGP*)

In the following, we will construct FSGP based on Shamir's (n, n)-SS, which can recover the original secret x by collecting n legitimate (or not forging) sub-secret. Our protocol controls a challenge-response session cycle by timestamp TS and △T. FSGP is invalid when the time of challenge-response exceeds one session cycle. The procedure is described as follows:

1. Initial Setup Phase

TDS selects a PRNG $g:\{0,1\}^k \rightarrow \{0,1\}^{2k}$ based on k and sets ID_i as the initial seed of g. All tags in Grouping-Proof have the ability of computing g and meanwhile TDS initializes the current state $s_{i0} = g(ID_i)$ of T_i. After that TDS stores the triples (ID_i, s_{i0}, K_{i0}) of T_i.

2. Challenge-Response Phase

(1) V selects a main-random-number x using g and constructs a polynomial $f(x)=a_0+a_1x+\ldots+a_{n-1}x^{n-1} \bmod p \in Z_p[x]$ of degree n-1 based on (n, n)-SS described in section 3.3, in which all coefficients a_0, a_1, …,a_{n-1} $\in GF(p)$. V sets $x=a_0= f(0)$ and keeps $f(x)$ secret.

(2) V→R: V generates n couples of sub-random-number (x_i, y_i) by $f(x)$ and sends them to R (i=1, 2,…, n). V stores TS and x of this grouping-proof session in

Table 1. Notations of FSGP

R	Reader	x	main-random-number based on (n, n)-SS
T_i	Tags	(x_i, y_i)	sub-random-number based on (n, n)-SS
V	Trusted Verifier which connects trusted database TDS	N	Set of all tags in Grouping-Proof
A	Adversary	n	Number of all tags in Grouping-Proof
PRNG	Pseudo-random Number Generator[20]	l	Number of response tags in △T
TS	Timestamp	MAC	Message Authentication Code
△T	Time of one authentication session	$MAC_{Ki_j}[m]$	MAC of message m with key K_{i_j}
ID_i	Identity of T_i	TDS	Trusted timestamp database, which contains TS, ID_i and message authentication codes of all legitimate tags T_i, s_{i_j} and K_{ij} between T_i and V
s_{i_j}	State of T_i in period j	H	Set of the legitimate tags
K_{ij}	Shared secret key of T_i in period j	S	Set of the illegitimate tags
P	Grouping-Proof evidence for T_i	h	Number of the legitimate tags

TDS. The set of all tags in Grouping-Proof is denoted as N={T_1,T_2,…,T_n}.

(3) R → T_i: R queries T_i by sending sub-random-number x_i (i=1, 2,…, n).

(4) T_i→R: T_i computes $m_i = \mathrm{MAC}_{Kij}(x_i)$ and sends the response (s_{ij}, m_i, x_i) for period j to R. (j=TS)

(5) R→V: R combines (s_{ij}, m_i, x_i) with another sub-random-number y_i to form $(s_{ij}, m_i, (x_i, y_i))$ and forwards $(s_{ij}, m_i, (x_i, y_i))$ to V.

(6) V: V stores l responses $(s_{ij}, m_i, (x_i, y_i))$ (l<=n) for period j in the time of △T and meanwhile forms the grouping-proof P of this session $P = (s_{1j}, m_1, (x_1, y_1), s_{2j}, m_2, (x_2, y_2), ..., s_{ij}, m_i, (x_i, y_i))$, which proceeds with Validity Authentication Phase.

3. Validity Authentication Phase

V searches the triples (ID_i, s_{ij}, K_{ij}) in TDS by s_{ij} and checks whether m_i is a valid MAC or not as follows:

If m_i is valid, V keeps $(s_{ij}, m_i, (x_i, y_i))$ in P. Then P will proceed with Legitimacy Authentication Phase.

Otherwise, V removes $(s_{ij}, m_i, (x_i, y_i))$ from P and puts T_i into S, which shows that T_i is attacked in the form of forging or interpolating the challenge-response by A. Then T_i in S will return the second phase of Challenge-Response and wait for proceeding with the next grouping-proof session.

4. Legitimacy Authentication Phase

If $P \neq \phi$, the third phase is valid and then V will proceed with this phase to authenticate legitimacy.

V gets l couples of sub-random-number (x_i, y_i) from P and proceeds with the following steps:

(1) If l=n, V recovers the secret x' based on (n, n)-SS and compares x' with the main-random-number x.

If x'= x, P is valid, and that means all of the tags (T_1, T_2,…, T_n) are legitimate, and V puts (T_1,T_2,…, T_n) into H.

Otherwise P is invalid, and that means there are suspicious tags in (T_1,T_2,…, T_n) and V puts (T_1,T_2,…, T_n) into S. Then T_i in S will return the second phase of Challenge-Response and wait for proceeding with the next grouping-proof session.

(2) If l<n, V checks l couples of sub-random-number (x_i, y_i) by the polynomial $f(x)$ as follows:

If $f(x)$ is equal, T_i is legitimate and V puts T_i into H.

Otherwise T_i is a suspicious tag and V puts T_i into S. Then T_i in S will proceed with the same to (1).

Note: This case indicates that the grouping-proof will convert into a single authentication proof.

5. State and Key Updating Phase

(1) If $H \neq \phi$, V computes every T_i in H by $D_i^{'} = g(K_{ij} + s_{ij})$ and after V sends D'_i to T_i by R, V will update s_{ij} and K_{ij} by $s_{ij+\Delta T} = g(s_{ij})$ and $K_{ij+\Delta T} = g(K_{ij})$. TDS stores $(ID_i, s_{ij+\Delta T}, K_{ij+\Delta T})$ and sets TS = j+△T.

After receiving D'_i, T_i computes $D_i = g(K_{ij} + s_{ij})$ and checks the relation between D'_i and D_i as follows:

If D'_i is equal to D_i, T_i will update s_{ij} and K_{ij} by $s_{ij+\Delta T} = g(s_{ij})$ and $K_{ij+\Delta T} = g(K_{ij})$. After that T_i deletes s_{ij} and K_{ij}.

Otherwise, T_i will keep s_{ij} and K_{ij} unchanged.

(2) If $S \neq \phi$, V computes every T_i in S by $E_i^{'} = g(s_{ij})$ and then sends it to T_i by R. After receiving E'_i, T_i computes $D_i = g(K_{ij} + s_{ij})$. Because E'_i is not equal to D_i, T_i and V will keep s_{ij} and K_{ij} unchanged.

(3) If h=n, P is a valid grouping-proof and that means all of the tags are simultaneously scanned and FSGP terminates.

Otherwise, there are suspicious tags in this session. V sets n=n-h and returns the second Challenge-Response Phase to proceed with the next grouping-proof session.

Notes: I. The Validity Authentication result of the third phase is described as follows:

(1) The case of m_i being valid MAC

① Legitimate T_i with K_{ij} and of not being attacked. Its response is $(s_{ij}, \mathrm{MAC}_{Kij}(x_i), (x_i, y_i))$.

② Legitimate T_i with K_{ij} and of being interpolated x_i in the *Query* command from R to T_i. Its response is $(s_{ij}, \mathrm{MAC}_{Kij}(x'_i), (x'_i, y_i))$.

③ Illegitimate T_i without K_{ij} and of being forged because it is out of the broadcast range of readers by eavesdropping K_{ij} in an illegitimate way or legitimate T_i with K_{ij} and of being forged by forging x_i in *Query* command from R to T_i because of x_i being blocked by A. Its response is $(s_{ij}, \mathrm{MAC}_{Kij}(x'_i), (x'_i, y_i))$.

(2) The case of m_i being invalid MAC

① Illegitimate T_i without K_{ij} and of not being attacked. Its response is $(s_{ij}, \mathrm{MAC}_{K'ij}(x_i), (x_i, y_i))$.

② Legitimate T_i with K_{ij} and of being interpolated *Response* message from T_i to R. Its response $(s_{ij}, \mathrm{MAC}_{Kij}(x_i), (x'_i, y_i))$ or $(s_{ij}, \mathrm{MAC}'_{Kij}(x_i), (x_i, y_i))$.

③ Illegitimate T_i without K_{ij} and of being interpolated *Query* command or *Response* message between T_i and R. Its response is $(s_{ij}, \mathrm{MAC}_{K'ij}(x_i), (x'_i, y_i))$ *etc*.

II. The result of Legitimacy Authentication Phase is described as follows:

V puts the legitimate T_i with K_{ij} and those haven't been attacked into H, but others are viewed as suspicious tags and are put into S. Then T_i in S will proceed with the same to (1) of Legitimacy Authentication Phase.

The generation process of FSGP is shown in Fig.1. The outputs of FSGP are the legitimate T_i in H and the suspicious T_i in S.

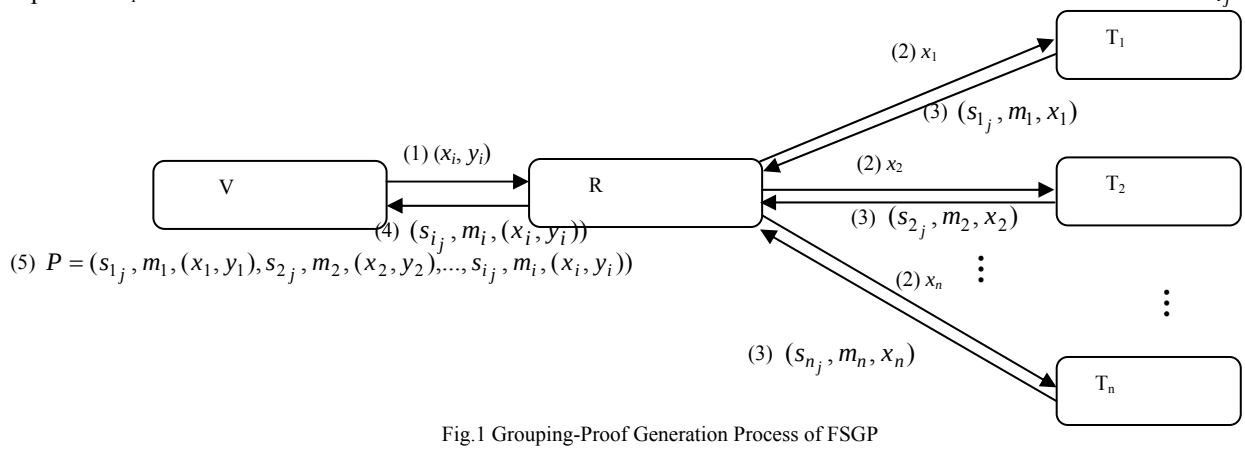

Fig.1 Grouping-Proof Generation Process of FSGP

5. Evaluation

In this section, we present the security and performance analysis of our protocol FSGP. In addition, we compare FSGP with previous research works based on the typical characteristics of the existing grouping-proof protocols.

5.1. *Security analysis*

(1) Tag Anonymity: Instead of transmitting static tag T_i identity in plaintext over R-T insecure communication channel, FSGP utilizes a dynamically generated random number x_i by using PRNG and a constructed polynomial $f(x)$ in (n, n)-SS to challenge each tag T_i directly during one session to achieve tag anonymity. The response of T_i is the triples (s_{ij}, m_i, x_i) which combine MAC of the challenge message x_i with the current state s_{ij} of T_i and only V has the right to verify this session. The responses computed by the tags do not leak any information interrelated with ID_i to any third-party who does not know the private key of T_i in the whole authentication session. Even though the transmitted message x_i or $\mathrm{MAC}_{K_{ij}}(x_i)$ over the insecure wireless channel can be eavesdropped, it is impossible to obtain the relevant information about ID_i of the legitimate tag T_i. The security robustness of ID_i embedded in transmitted messages x_i will not be compromised. Hence, tag anonymity can be guaranteed in our scheme.

(2) Untraceability: On concern of the privacy, FSGP randomizes the direct challenge-response among readers and tags in one session. Since FSGP offers privacy protection against an adversary, the transmitted message over R-T channel and the current state value s_{ij} of T_i depend on the dynamically generated random number x_i which is randomized in different proof sessions. Moreover V changes every Timestamp TS after a successful verification, which adds the difficulties for an attacker to trace tags. On account of the triples (s_{ij}, m_i, x_i) not being the same in different sessions, the adversary cannot obtain the same responses from the same tag T_i by interfering with two or more dependent challenge-response. So the adversary cannot track the legitimate tag T_i and untraceability can be guaranteed. This feature ensures location privacy protection of the tagged objects.

(3) Forward Security: In FSGP, forward security is naturally embedded because K_{ij} shared between Verifier and T_i and s_{ij} of the legitimate tags T_i in H will be automatically updated after each valid grouping-proof session. V sends the message D_i to T_i in H and after T_i checking D_i valid, T_i and V will update s_{ij} and K_{ij} by the updating algorithm $s_{ij+\Delta T} = g(s_{ij})$ and $K_{ij+\Delta T} = g(K_{ij})$. Meanwhile TDS stores the updated triple $(ID_i, s_{ij+\Delta T}, K_{ij+\Delta T})$. s_{ij} and K_{ij} are generated by PRNG and $(s_{ij}, m_i, x_i, K_{ij})$ are updated according to the different sessions. Even if $(s_{ij}, m_i, x_i, K_{ij})$ is eavesdropped, the adversary is not able to obtain the transmitted valid message between reader and tags for the prior period. It is known from $m_i = \mathrm{MAC}_{K_{ij}}(x_i)$ that m_i is constructed by K_{ij} and the challenge x_i which depends on the sub-random-

number (x_i, y_i) according to the different sessions. It is obvious that $(s_{ij}, m_i, x_i, K_{ij})$ has the characteristic of random and periodic and the adversary cannot obtain $(s_{ij-\Delta T}, m_{i-\Delta T}, x_{i-\Delta T}, K_{ij-\Delta T})$ for period j-$\triangle$T by $(s_{ij}, m_i, x_i, K_{ij})$ for period j. The difficulty of obtaining $s_{ij-\Delta T}, K_{ij-\Delta T}$ by $(s_{ij}, m_i, x_i, K_{ij})$ is equivalent to attacking PRNG. Therefore, even if T_i was compromised in period j, the grouping-proof before period j is valid. Hence, the evolutions of K_{ij}, s_{ij} and the grouping-proof all have forward security which protects past communications of a compromised tag.

(4) Resistance to Replay Attack: It was a specific design feature of FSGP that only the trusted verifier can check the correctness of the grouping-proof. FSGP uses the randomized direct challenge-response, MAC computation of the dynamically generated random number x_i with K_{ij} to defend against replay attack and meanwhile V stores $(s_{ij}, m_i, (x_i, y_i))$ in TDS. Because of this feature, V will accept the response only when the two responses $(s_{ij}, m_i, (x_i, y_i))$ of T_i are different. That is if two or more responses of T_i are the same in one session, V will refuse to accept. Additional, even if the triples (s_{ij}, m_i, x_i) are eavesdropped, an adversary cannot impersonate the legitimate tag T_i by replaying the response of T_i and V can identify the adversary. Hence, our protocol can resist replay attack.

(5) Resistance to Man-In-The-Middle Attack (MITM): In order to obtain valid message of tags in a successful session, an adversary tries to interpolate the transmitted message over R-T channel and interfere in the challenge-response but V-R-T cannot detect.

Case① Supposing that an adversary eavesdrops one or more challenges x_i during the *challenge* phase and interpolates it into x'_i. To this case, MITM attack is described as follows:

(I) If the adversary sends x'_i to legitimate T_i, the response of T_i is $(s_{ij}, \mathrm{MAC}_{K_{ij}}(x'_i), x'_i)$ and then R sends $(s_{ij}, \mathrm{MAC}_{K_{ij}}(x'_i), (x'_i, y_i))$ to V. According to Validity Authentication Phase, it is clear that $\mathrm{MAC}_{K_{ij}}(x'_i)$ is a valid MAC by x'_i and K_{ij}. But in Legitimacy Authentication Phase, the main-random-number x' that V reconstructs by l couples of sub-random-number (x'_i, y_i) based on (n, n)-SS is not equal to the main-random-number x in this session. Therefore, Legitimacy Authentication Phase will not be allowed, that means MITM attack cannot succeed.

(II) If the adversary sends x'_i to illegitimate T_i, the response of T_i is $(s_{ij}, \mathrm{MAC}_{K'_{ij}}(x'_i), x'_i)$ and then R sends $(s_{ij}, \mathrm{MAC}_{K'_{ij}}(x'_i), (x'_i, y_i))$ to V. According to Validity Authentication Phase, it is clear that $\mathrm{MAC}_{K'_{ij}}(x'_i)$ is invalid by x'_i and K_{ij}. Therefore, Validity Authentication Phase will not be allowed, that means MITM attack cannot succeed.

Case② Supposing that an adversary eavesdrops one or more responses of T_i during the *response* phase and interpolates them. To this case, MITM attack is described as follows:

(I) If the adversary interpolates the response of legitimate T_i into $(s_{ij}, \mathrm{MAC}_{Kij}(x_i), x'_i)$ or $(s_{ij}, \mathrm{MAC}'_{Kij}(x_i), x_i)$ and then R sends $(s_{ij}, \mathrm{MAC}_{Kij}(x_i), (x'_i, y_i))$ or $(s_{ij}, \mathrm{MAC}'_{Kij}(x_i), (x_i, y_i))$ to V. According to Validity Authentication Phase, it is clear that $\mathrm{MAC}_{Kij}(x_i)$ and $\mathrm{MAC}'_{Kij}(x_i)$ are invalid MAC by K_{ij}. Therefore, Validity Authentication Phase will not be allowed, that means MITM cannot succeed.

(II) If the adversary interpolates the response of illegitimate T_i into $(s_{ij}, \mathrm{MAC}_{K'ij}(x_i), x'_i)$ *etc.* and then R sends $(s_{ij}, \mathrm{MAC}_{K'ij}(x_i), (x'_i, y_i))$ *etc.* to V. Similarly, Validity Authentication Phase will not be allowed, that means MITM attack cannot succeed.

Based on the analysis of Case①②, the interference of FSGP will not be successful. Furthermore, even if the adversary eavesdrops all of n challenge x_i of this current session, he cannot be able to reconstruct the main-random-number x of this current session without knowing y_i. Hence, FSGP are immune to MITM attack.

(6) Resistance to Counterfeit Attack: To defend against counterfeit attack, FSGP utilize a timeout mechanism to ensure that all the proof-involved tags coexist at a specific and limited time period. V will not accept the response which is forged or reaches out of TS in the current session. The detailed analysis is described as follows:

Case① T_i Impersonation Resistance: An adversary tries to impersonate a legitimate T_i within the broadcast range of R in this session by forging the legitimate response $(s_{ij}, \mathrm{MAC}_{Kij}(x_i), x_i)$. On account of K_{ij} only shared by V and T_i, even if the legitimate response $\mathrm{MAC}_{Kij}(x_i)$ is eavesdropped from R-T channel, the difficulty of obtaining K_{ij} of the legitimate T_i by $\mathrm{MAC}_{Kij}(x_i)$ is equivalent to attacking MAC [1]. Supposing that the adversary impersonates a tag T_i' with the secret key K'_{ij}, the corresponding response of T_i' is $(s_{ij}, \mathrm{MAC}_{K'ij}(x_i), x_i)$ to the challenge x_i from R

and then R sends $(s_{ij}, \mathrm{MAC}_{K'ij}(x_i), (x_i, y_i))$ to V. Since $K'_{ij} \neq K_{ij}$, V utilizes K_{ij} to proceed with Validity Authentication Phase and then gets the result $\mathrm{MAC}_{K'ij}(x_i) \neq \mathrm{MAC}_{Kij}(x_i)$ which indicates $\mathrm{MAC}_{K'ij}(x_i)$ is invalid MAC. It is clear that Validity Authentication Phase cannot be validated. Even though the adversary tries to modify the received challenge x_i into x'_i to meet the requirement of $\mathrm{MAC}_{K'ij}(x'_i) = \mathrm{MAC}_{Kij}(x_i)$, this difficulty is also equivalent to attacking MAC[1]. Therefore, T_i Impersonation Resistance will not succeed.

Moreover, FSGP can defend against illegitimate T_i of malicious counterfeit which is activated by a malicious R' because T_i is out of the broadcast range of the readers. The detailed analysis process is similar to the above T_i Impersonation Resistance and (5) Case①. Therefore this kind of malicious attack cannot succeed.

Case② R Impersonation Resistance: An adversary tries to impersonate a legitimate R in this session and transmits the challenge-response over V-T by the forged R. Because the transmitted challenge-response over V-T communication channel are not relevant to ID_i of legitimate T_i and moreover s_{ij} is generated by PRNG and updated according to the different sessions, the forged R cannot obtain any information about the privacy of T_i. If the forged R tries to eavesdrop and modify the transmitted challenge-response over V–T, the grouping-proof will not be allowed, that means that attacking to R Impersonation cannot succeed. The detailed process refers to the analysis of (5).

Hence, FSGP can resist both T_i and R impersonation attack and has the property of strong unforgeability.

5.2. *Performance analysis*

Since RFID tags are generally low cost with extremely limited resources, it is necessary for tags to achieve authentication by using the lightweight primitives. According to the requirement of resource-constrained devices, in our grouping-proof FSGP, only simple control commands and three primitive arithmetic operations are required, such as Lagrange interpolating polynomial $f(x)$ construction, random number generator PRNG() and minimalist keyed message authentication code MAC[] of sub-random-number. Moreover, we put the construction operation of $f(x)$ over Verifier and the operations of PRNG and MAC are required at the tag end. Based on the research results in[1-4] and EPC standard specification[21], it is proved that these computation costs can be afforded by resource-constrained tags. Hence, we think that FSGP is very competitive to be a solution candidate on forward-secure grouping-proof for RFID tags.

5.3. *Security comparison*

In the following, we compare FSGP with previous related works in terms of security and privacy aspects in Tables 2, which shows that security robustness of our protocol is superior to the others by supporting tag anonymity, untraceability, forward security, and resisting to security threats such as replay attack, MITM attack and counterfeit attack.

In summary, based on the above analysis and security comparison, our protocol FSGP has the characteristics of forward-security, robustness, order-

Table 2. Security comparison between FSGP and related grouping-proof protocols

	Tag Anonymity	Untraceability	Forward Security	Resistance to Replay Attack	Resistance to MITM	Resistance to Counterfeit Attack
Yoking-proofs[1]	N	N	N	N	N	N
Grouping-proofs[2]	N	N	N	N	N	N
Existences-proofs[3]	N	N	N	Y	Y	Y
Generalized Yoking-proofs[4]	N	N	N	Y	Y	Y
Clumping-proofs[5]	Y	N	N	Y	Y	Y
Provably–secure proofs[6]	Y	Y	Y	Y	Y	N
Coexistence-proofs[7]	N	N	N	Y	Y	N
Order-independent proofs[8]	N	N	N	Y	Y	N
FSGP	Y	Y	Y	Y	Y	Y

independence and efficiency com pared with the previous related protocols.

6. Conclusion

To overcome the weakness of security and/or privacy omission and order-dependence in the previous grouping-proof protocols, in this paper we develop a grouping-proof protocol with forward security for multiple RFID tags based on Shamir's (n, n) secret sharing, called FSGP, which solves the scalability issue properly by avoiding relaying message among tags through RFID reader and meanwhile achieves security enhancement and robust privacy protection. FSGP can defend against malicious attacks a nd possess excellent privacy properties and also realizes a single authentication protocol in combination with a grouping-proof protocol properly by the application of CI. In terms of pr otocol performance measurement, our protocol is lightweight which meets the requirement of resource-constrained RFID tags without increasing much computing burden at both tag end and server end. In the future, as com plexity of technol ogy and networks' services incr ease new challenging m ulti-combinatorial problems are emerging and consequently the CI applications are apt to further enhancement in the environment of Internet of things.

Acknowledgements

This work is suppo rted by the National Natural Science Foundation of China(60673127), the National 863 High Technology Research and Development Program of China(2007AA01Z404), the Fund for the Doctoral Program of Hig her Education of China (20103218110017), the Electronic Development Fund of the M inistry of Information Industry, the Jiangsu Province Science & Technology Pillar Program (BE2008135), the Fund by the Priority Academic Program Development of Jiangsu Higher Education Institutions, the Fundamental Research Funds for the Central Universities: the F unding of Jiangsu Innovation Program for Graduate Education(CX10B_112Z), the Funding for Outstanding Doctoral Dissertation in NUAA(BCXJ10-07), the Natural Science Foundation of Jiangsu Normal University for Grant(11XLA09), the China Postdoctoral Science Foundation(20100481133), the Jiangsu Province Postdoctoral Science Foundation(1001005B) and the National Natural Science Cultivation Foundation of China(NS2012023), under which the present work was possible.

References

1. A. Juels, Yokin g-proofs for RFID tags, in *Proc. 2nd IEEE Annual Conference on PERCOMW'04* (IEEE Press 2004), pp. 138-143.
2. J. Saito, K. Sak urai, Grouping proof for RFID tags, in *Proc. 19th IEEE International Conference on AINA'05* (IEEE Press 2005), pp. 621-624.
3. S. Piramuthu, On existence proofs for multiple RFID tags, in *Proc. IEEE International Conference on ICPS'06* (IEEE Press 2004), pp. 317-320.
4. L. Bolotnyy and G. Robins, Generalized yoking-proofs for a group o f RFID tags, in *Proc. 3rd Annual International Conference on MUS'06* (2006), pp. 1-4.
5. P. Peris-Lopez, J. C. He rnandez-Castro, Estevez-Tapiador J. M., and Ribagorda A., Solving th e simultaneous scanning prob lem anonymously: Clumping proofs for RF ID tags, in *Proc. 3rd International Workshop on SPTPUC'07* (2007), pp. 55-60.
6. M. Burmester, B. D. Medeiros and R. Motta, Provably Secure Grouping-Proofs for RFID Tags, in *Proc. 8th International Conference on CARDIS'08*(Springer-Verlag, LNCS 5189, 2008) , pp.176–190.
7. C. C. Lin, Y. C. Lai, J. D. Tygar, C. K. Yang, and Chiang C. L., Coexistence proof using chain of timestamps for multiple RFID t ags, in *Proc. International Workshop on APWeb/WAIM'07*(Springer-Verlag, LNCS 5189 , 2007), pp.634-643.
8. Y. H. Lien, X. F. Leng, K. Mayes, and J. H. Chiu, Reading order independent grouping proof for RFID tags, in *Proc. IEEE International Conference on ISI'08*(IEEE Press 2004), pp. 128-136.
9. W. Pedrycz, A.V. Vasilakos, *Computational intelligence in telecommunications networks* (CRC Press, Boca Raton, FL, 2000) .
10. A.V. Vasilakos, W. Pedrycz, *Ambient Intelligence, Wireless Networking, Ubiquitous Computing* (Art House, MA, USA, 2006).
11. A. Konar, *Computational Intelligence: Principles, Techniques and applications*(Springer-Verlag, New York, 2005).
12. G. K. Vena yagamoorthy, A successful in terdisciplinary course on computational intelligence, *IEEE Computational Intelligence Mag.* 4(1) (2009) 14–23.
13. A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd edn. (NY, U SA: John Wiley &Sons, 2007).
14. H. Y. Chien, S. B. Liu, Tree-Based RFID Yoking Proof, in *Proc. International Conference on NSWCTC'09* (IEEE Press 2009), pp. 550-553.
15. H. Y. Chien, C. C. Yang, T. C. Wu, C. F. Lee, Two RFID-based Solutions to Enhance Inpatient Medication Safety, *J. Med. Syst.* 35(3) (2011) 369-375.
16. M. Burmester, O. Muni lla, Lightweight RFID authentication with forward and backward security, *ACM Transactions on Information and System Security* (*TISSEC*). 14 (1) (2011) 11-37.
17. H. Lein, L. L. Chang, Strong (n, t, n) verifiable secret sharing scheme, *Information Sciences*. (180) (2010) 3059–3064.
18. G. R. Blakley, Safeguarding cryptographic keys, in *Proc. AFIPS Conf.* (NCC, Arlington, Va. 48, 1979), pp. 313-317.
19. A. Shamir, How to share a secret, *Communications of the ACM*. 22(11) (1979) 612-613.
20. P. Peris-Lopez,, J. C. Hernandez-Castro, Estevez-Tapiador J. M., and Ribagorda A., LAMED – A PRNG for EPC class-1 generation-2 RFID specification, *Computer Standards and Interfaces*. (31) (2009) 88-97.

21. EPCglobal Class 1 Gener ation 2 UHF Air I nterface Protocol Standard Version 1.0.9., Available from: *http://www.EPCglobalinc.org/*.