Grouping-Proofs-Based Authentication Protocol for Distributed RFID Systems

Hong Liu, *Student Member*, *IEEE*, Huansheng Ning, *Senior Member*, *IEEE*, Yan Zhang, *Senior Member*, *IEEE*, Daojing He, *Member*, *IEEE*, Qingxu Xiong, *Member*, *IEEE*, and Laurence T. Yang, *Member*, *IEEE*

Abstract—Along with radio frequency identification (RFID) becoming ubiquitous, security issues have attracted extensive attentions. Most studies focus on the single-reader and single-tag case to provide security protection, which leads to certain limitations for diverse applications. This paper proposes a grouping-proofs-based authentication protocol (GUPA) to address the security issue for multiple readers and tags simultaneous identification in distributed RFID systems. In GUPA, distributed authentication mode with independent subgrouping proofs is adopted to enhance hierarchical protection; an asymmetric denial scheme is applied to grant fault-tolerance capabilities against an illegal reader or tag; and a sequence-based odd-even alternation group subscript is presented to define a function for secret updating. Meanwhile, GUPA is analyzed to be robust enough to resist major attacks such as replay, forgery, tracking, and denial of proof. Furthermore, performance analysis shows that compared with the known grouping-proof or yoking-proof-based protocols, GUPA has lower communication overhead and computation load. It indicates that GUPA realizing both secure and simultaneous identification is efficient for resource-constrained distributed RFID systems.

Index Terms—RFID, security, authentication protocol, grouping proof, distributed system

1 Introduction

Radio frequency identification (RFID) as an emerging sensor technique has been developed in various applications. Due to the limited communication resources and computation capabilities, several problems restrict its extensive development. Particularly, security issues are increasingly concerned in recent studies [1], [2], and are also confronting with severe challenges. Conventional cryptographic primitives have low portability on low-cost tags with inadequate power and storage, which may make security issue more formidable.

Different techniques have been proposed to strengthen security protection for RFID systems, including physical mechanism, authentication protocol, access control, and encryption algorithm. Thereinto, authentication is the principal scheme that owns ubiquitous applicability [3], [4], [5]. For instance, ultralightweight protocols mainly apply bitwise operations to achieve the tag-reader air interface security. Lightweight protocols mainly adopt hash function, cyclic redundancy code (CRC), message authentication code (MAC) and pseudo-random number generator (PRNG) for authentication. Middleweight protocols mainly use full-fledged cryptographic primitives such as symmetric encryption to satisfy high security requirements. However, most RFID security protocols focus on the case of single reader and single tag while ignoring the simultaneous identification among multiple readers and tags. In practical applications, there are many scenarios that need multiple entities' interactions. For instance, 1) *In the inventory management*, a number of goods should to be associated with an authorized user; 2) *In the valuables traceability service*, an evidence is needed to provide that a valuable article has been present in the multiple readers' overlay areas; 3) *In the supply chain management*, quick entry identification is needed by diverse interest groups. Generally, time-division multiple access is the mainstream algorithm to solve the mentioned multiple objects identification problem in RFID systems. Some examples are aloha-based protocol, tree-based protocol, and their variations [6], [7]. Such schemes mainly realize batch identification in the data link layer without considering security protection. Most previous researches consider authentication and multiple objects simultaneous identification as separated research areas since the former belongs to the application layer while the latter belongs to the data link layer. Hence, it becomes significant to design a new scheme to provide coexistence proofs for realizing both secure and simultaneous identification in distributed RFID systems.

The concept of generating an evidence to achieve two tags secure identification was first introduced by Juels [8]. He presented a distinctive *yoking-proofs* protocol to deal

- *H. Liu, H. Ning, and Q. Xiong are with the School of Electronic and Information Engineering, Beihang University, Box 205, Xueyuan Road No.37, Haidian District, Beijing 100191, China. E-mail: liuhongler@ee.buaa.edu.cn, {ninghuansheng, qxxiong}@buaa.edu.cn.*
- *Y. Zhang is with the Simula Research Laboratory of Norway and the Department of Informatics, University of Oslo, Martin Linges v17, Fornebu, PO Box 134, 1325 Lysaker, Norway. E-mail: yanzhang@simula.no.*
- *D. He is with the College of Computer Science, Zhejiang University, Yuquan Campus, Zheda Road #38, Hangzhou 310027, China. E-mail: hedaojinghit@gmail.com.*
- *L.T. Yang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, China, and the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada. E-mail: ltyang@stfx.ca.*

Manuscript received 13 Nov. 2011; revised 18 June 2012; accepted 9 July 2012; published online 20 July 2012.
Recommended for acceptance by M. Guo.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-11-0832.
Digital Object Identifier no. 10.1109/TPDS.2012.218.

with the problem that two tags are verified and scanned within a reader's interrogation range. Saito and Sakurai [9] proposed a grouping-proofs protocol which is extended from the yoking proofs. Burmester et al. [10] focused on an anonymous grouping proof for two tags. Thereafter, several protocols based on yoking proofs or grouping proofs are designed [11], [12], [13], [14], [15], [16], [17], in which simultaneous existences of multiple tags are regarded as a pair or a group to be verified by a reader. Grouping proof is an evidence that multiple tags can be simultaneously authenticated by a single reader, or multiple readers can be simultaneously authenticated by a single tag. We take the supply chain management as an example. *For a group of multiple readers*, when a number of goods are transferred from a material supplier to a carrier, it is necessary to perform independent identification by both the material supplier and the carrier. The reason is that the material supplier and the carrier may concern different tag fields and have private information towards the same tag. In this case, the grouping proof is able to prove that multiple readers of different interest groups have transmitted the same batch of goods. *For a group of multiple tags*, the grouping proof can prove that multiple tagged goods have been transmitted via a certain intermediate link. It is necessary to perform secure and simultaneous identification on the multiple tags by different interest groups (e.g., material supplier, carrier, and retailer).

However, most existing grouping-proofs-based protocols mainly consider the case that a single reader accesses multiple tags, but ignore the other case that a single tag (or multiple tags) may be simultaneously scanned by multiple readers. Meanwhile, several defects exist in the previous protocols. For instance, the reader and tags are based on centralized structure, which may lead to low scalability. The reader obtains the grouping proofs for final verification but discounts the intermediate verifications. The tags perform the nonlightweight secret updating, which may limit the protocol's applications.

In this paper, we propose a grouping-proofs-based authentication protocol (GUPA) for readers and tags secure and simultaneous identification for distributed RFID systems. The main contributions are as follows: We build a distributed authentication mode to make the subgrouping proofs relatively independent, and apply different tag/reader groups to achieve hierarchical identification. We adopt the asymmetric denial mechanism to grant diverse fault-tolerance capabilities against attackers. Such scheme can resist the denial-of-proof (DoP) attack in which the attacker aims to disturb the normal identification on legal tags. We design a lightweight secret updating algorithm, in which the readers and tags use specific extendible functions to realize random accessing replacing additional update module and redundant workloads.

The remainder of this paper is organized as follows: Section 2 presents a review of related works. Section 3 introduces the detailed phases of the proposed GUPA in three cases. Then, attack analysis and performance analysis are given in Sections 4 and 5, respectively. Finally, Section 6 draws a conclusion.

2 Related Works

Burmester et al. [10] proposed three protocols: 1) A nonanonymous protocol uses a counter to realize state update, and applies group keys to avoid useless proofs. A tag computes a pseudorandom function to prove that another tag belongs to a certain group. The grouping proofs are applied to realize simultaneous scanning, and to confirm the correctness of tags. 2) An anonymous protocol achieves anonymity by the randomized pseudonyms instead of the former group identifier. The current value and the previous value are introduced to guarantee unlinkability. 3) An anonymous and forward-security protocol updates the secret keys and the group keys after each season, where the tag stores the secret/group key and the group pseudonyms to enhance security. Burmester's robust grouping proof is proved to be vulnerable against an impersonation attack [11].

Lo et al. [12] introduced two types of coexistence-proofs-based protocols to protect tag privacy, forward secrecy, and sequential authentication. Specifically, the online verifier-based protocol (OVBP) is designed for a trusted online database that stores necessary and relative data for all the legal tags. In OVBP, DoP attack is resisted by the multiple-tag authentication which also prevents the generation of invalid coexistence-proofs. Moreover, a data redundancy mechanism is adopted to defend against the denial of service attack. The offline time stamp server-based protocol (OTSBP) is designed in which a tamper-resistant time stamp module is equipped since the backend database may be temporarily unavailable. In OTSBP, the proof involving the shared secret key and the tag identifier is used for authentication. Tag identifiers and derived keys are used to check the validity of coexistence proofs.

Cho et al. [13] focused on the replay attack and proposed an enhanced yoking-proof protocol for multiple tags' simultaneous scanning. The main functions used in the protocol are MAC and PRNG functions. Meanwhile, Lamport signature scheme is presented in the MAC function to realize error code check. The proof obtained with an encrypted value may potentially increase additional computation, and is not suitable for passive RFID tags.

Huang and Ku [14] designed a grouping-proofs-based protocol for passive tags with the EPCglobal C1G2 standard. The protocol based on PRNG and CRC functions is designed for medication safety applications. The protocol uses the CRC checksum code to detect error and to verify the integrity of transmitted data, which may result in the protocol vulnerability for DoS attack due to the linear properties of the CRC function. Chien et al. [15] proved that Huang's protocol is vulnerable to replay attack, and then proposed two protocols (online and offline) to enhance inpatient medication safety. In the online protocol, the reader and tags share a secret, and the reader associates the tags by checking the correctness of the received tuples. Additionally, the reader establishes an evidence for the offline protocol. Both protocols cannot resist the forgery and replay attack.

Peris-Lopez et al. [11] performed further studies to review the security flaws in the above protocols, and proposed Kazahaya protocol. Kazahaya is designed for two tags simultaneous and secure scanning, and it is only based on an unilateral authentication mode without

TABLE 1
Notations

Notation	Description
R_x, T_y	The x-th reader, and the y-th tag.
G_r, g_t	The r-th reader group, and the t-th tag group.
$\hat{R}_{\mathcal{A}}$, $\hat{T}_{\mathcal{A}}$	The illegal reader and tag.
PID_{R_x}, PID_{T_y}	The pseudonym of R_x and T_y.
$\widehat{PID}_{\hat{R}}$, $\widehat{PID}_{\hat{T}}$	The imitative identifier of $\hat{R}$ and $\hat{T}$.
GID_r, gid_t	The pseudo group identifier of the reader group G_r, and the tag group g_t.
F_{R_x}, F_{T_y}	The pseudo-random flag of R_x and T_y, which act as a label with timestamp.
L_R, L_T	The reader access list, and tag access list.
r	The pseudo-random number.
S_{xy}/S_{yx}	The l-bit length secrets owned by R_x/T_y respectively, and are pre-shared with T_y/R_x, $(S_{xy} = S_{yx})$.
M^{ℓ}	The locally derived value M.
$\widehat{M}$	The forged value of M.

verifying the reader by the tags. Such vulnerability may be utilized by a malicious attacker whose purpose is not to obtain the tag identifier, but to disturb the communication among legal entities.

The previous studies mainly focus on multiple tags identification by a single reader, ignoring other scenarios that multiple readers may concurrently identify a single tag or multiple tags. In this paper, we apply lightweight bitwise logical operators to achieve such multiple readers and tags secure and simultaneous identification.

3 Authentication Protocol Description

3.1 System Initialization

Suppose that the RFID system comprises readers $\{R_1, R_2, \ldots, R_x\}$, tags $\{T_1, T_2, \ldots, T_y\}$, and the backend system DB. The readers and tags store their own pseudonym, group identifier, identity flag, access list, and a set of preshared secrets $\{\mathcal{S}\}$. Here, each reader and each tag share a corresponding private secret. The legal tags are divided into z groups which are identified by the specific group identifiers $\{gid_1, \ldots, gid_z\}$, and the legal readers belong to z' groups which are identified by $\{GID_1, \ldots, GID_{z'}\}$. Different tag/reader groups are granted independent authorities to realize hierarchical access control. The notations are introduced in Table 1.

When a new reader joins the system, the uncertain reader should be authenticated by DB. In GUPA, a ring signature is introduced for the anonymous authentication, which is inspired by [18], [19]. Considering the reader's hardware condition, the ring signature scheme can be designed by the lightweight cryptographic algorithm such as elliptic curve cryptography, here DB acts as a verifier to perform authentication on an uncertain reader. Suppose that the new uncertain reader R_j is in the reader set $G_v = \{R_1, \ldots, R_j, \ldots, R_J\}$ with the ring size J, in which the pairwise public key Y_R and privacy key x_R satisfy the function that $Y_R = \log_\rho x_R$, $(\rho \in \{0,1\}^*, x_R \in \{0,1\}^*, Y_R \in \mathbb{Z}_q^*)$. Two hash functions are defined: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, and $H_2 : \mathbb{Z}_q^* \times \mathbb{Z}_q^* \rightarrow \{0,1\}^*$, in which q is a large prime number. R_j generates a ring signature of m_R on behalf of other readers in G_v.

R_j randomly chooses $\xi_\alpha \in \{0,1\}^*$, $(\alpha = 1, \ldots, J)$, and computes $P_{R_\alpha} = \log_\rho \xi_\alpha$, $(\alpha \neq j)$. Thereafter, R_j randomly chooses $\xi \in \{0,1\}^*$ to compute P_{R_j} and s_P:

$$P_{R_j} = \log_\rho \xi - \sum_{\alpha=1,\alpha\neq j}^{J} (H_2(H_1(m_R), H_1(P_{R_\alpha})) + Y_{R_\alpha}),$$

$$s_P = \rho^{H_2(H_1(m_R),H_1(P_{R_j}))} x_{R_j} \xi \prod_{\alpha=1,\alpha\neq j}^{J} \xi_\alpha \pmod q.$$

The signature $\delta(m_R)$ is established as that $\{P_{R_1}, \ldots, P_{R_j}, Y_{R_1}, \ldots, Y_{R_j}, s_P\}$, and R_j transmits $\delta(m_R)\|m_R$ to DB for verification. DB first extracts $\{P_{R_\alpha}, s_P\}$, and computes $h_\alpha = H_2(H_1(m_R), H_1(P_{R_\alpha}))$. Afterward, DB computes $\delta(m_R)$, and performs the verification by comparing $\sum_{\alpha=1}^{J}(P_{R_\alpha} + h_\alpha + Y_{R_\alpha})$ with $\log_\rho s_P$. If $\sum_{\alpha=1}^{J}(P_{R_\alpha} + h_\alpha + Y_{R_\alpha}) = \log_\rho s_P$ holds, R_j will be authenticated by DB. Thereafter, DB assigns the new reader R_j with the corresponding preshared values. Subsequently, the tags $\{T_1, T_2, \ldots, T_y\}$ update the locally stored reader access list L_R, and we consider T_y as an example to describe the access list updating. Note that the reader access list should also be updated when a reader leaves the system:

1. DB generates a pseudorandom number r_{DB}, extracts the updating command $Comd$, and transmits the cascaded value $r_{DB}\|Comd$ to challenge T_y.
2. When T_y detects the updating command $Comd$, it generates a pseudorandom number r_{T_y}, extracts its local access list L_R, and computes $H_1(L_R\|r_{DB})$. Thereafter, T_y replies $r_{T_y}\|H_1(L_R\|r_{DB})$ to DB.
3. DB recomputes $H_1(L_R\|r_{DB})$ by its locally stored $\{L_R, r_{DB}\}$ to verify T_y. If the recomputed hash value equals the received one, T_y will be regarded as a legal tag. Otherwise, the protocol will terminate. Afterward, DB further extracts the new reader R_j's information Δ_{R_j} to compute $H_1(\Delta_{R_j}\|L_R\|r_{T_y})$, and applies the private PRNG function to compute $PRNG(\Delta_{R_j})$. Afterward, DB transmits
$$H_1(\Delta_{R_j}\|L_R\|r_{T_y})\|PRNG(\Delta_{R_j}) \quad \text{to} \quad T_y.$$
4. Upon receiving the message, T_y first performs an inverse function $PRNG^{-1}()$ to derive Δ_{R_j}, and then re-computes $H_1(\Delta_{R_j}\|L_R\|r_{T_y})$ to verify DB. If the re-computed hash value equals the received one, DB will be regarded as a legal backend system, and T_y will update the reader access list by adding Δ_{R_j} into L_R. Otherwise, the protocol will terminate.

Till now, T_y's reader access list L_R has been updated. When a new tag joins or leaves the system, the readers $\{R_1, R_2, \ldots, R_x\}$ update the tag access list L_T according to the similar approach. The access list updating algorithm is mainly based on the lightweight PRNG and hash functions, which have low computation loads (CLs).

In GUPA, we also define a sequence $Sequ = [G_R g_t]$ and a function $G \cdot g(X)$ to perform the secret updating. Let G_R/g_t denote the reader/tag group; x/y denote the number of the reader/tag groups. Toward $Sequ$, the subscripts of $\{G_R, g_t\}$ are arrayed in an odd-even alternation mode. Metaphorically, $Sequ$ is regarded as a chain: From the front to the half, $\{G_R, g_t\}$

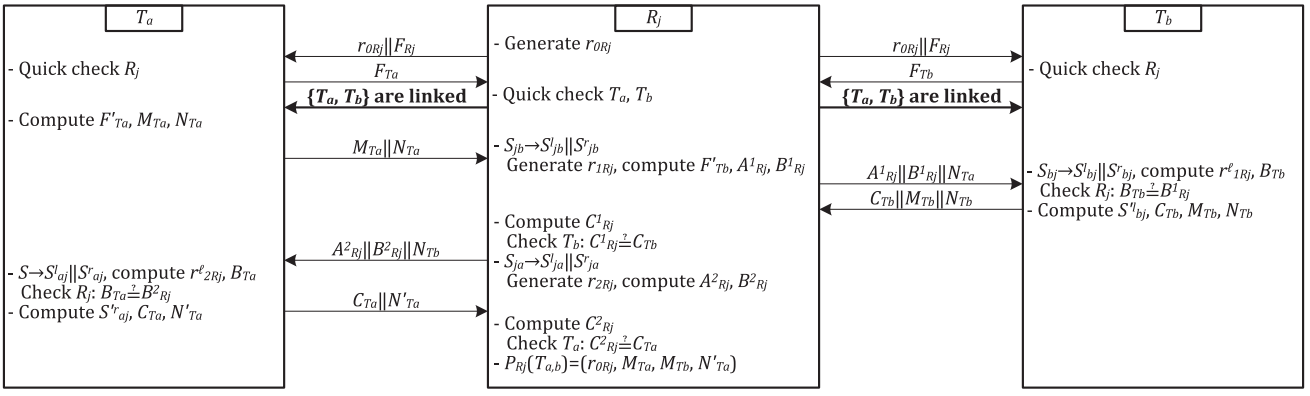


Fig. 1. GUPA: two-tag and single-reader case (2T-R).

are arrayed in the link form of $[G_{odd}g_{even}]$, and from the back to the half, $\{G_R, g_t\}$ are arrayed in the link form of $[G_{even}g_{odd}]$. The formal function definition $Sequ$ is as follows:

- $\exists\, x < y,\ \{n, n'_1, n''_1\} \in \mathbb{N},\ 1 < 2(n + n'_1) - 1 \le x$, and $1 < 2(n + n''_1) \le x$. Here, $y = 2m$ or $y = 2m + 1$:

$$\begin{aligned} Sequ_{(x<y)} = {} & G_1 g_2, \ldots, G_{2n-1} g_{2n} G_1 g_{2(n+1)}, \ldots, \\ & G_{2(n+n'_1)-1} g_{2m} G_{2(n+n''_1)} g_{2m+1}, \ldots, \\ & G_2 g_{2n+1} G_{2n} g_{2n-1}, \ldots, G_2 g_1. \end{aligned}$$

- $\exists\, x = y$, and $n \in \mathbb{N}$. Let $\{x, y\}$ equal $2n - 1$ or $2n$:

$$\begin{aligned} Sequ_{(x=y)_{odd}} &= G_1 g_2, \ldots, G_{2n-1} g_2 G_2 g_{2n-1}, \ldots, G_2 g_1. \\ Sequ_{(x=y)_{even}} &= G_1 g_2, \ldots, G_{2n-1} g_{2n} G_{2n} g_{2n-1}, \ldots, G_2 g_1. \end{aligned}$$

- $\exists\, x > y,\ \{n, n'_2, n''_2\} \in \mathbb{N},\ 1 < 2(n + n'_2) \le y$, and $1 < 2(n + n''_2) - 1 \le y$. Here, $x = 2m - 1$ or $x = 2m$:

$$\begin{aligned} Sequ_{(x>y)} = {} & G_1 g_2, \ldots, G_{2n-1} g_{2n} G_{2n+1} g_2, \ldots, \\ & G_{2m-1} g_{2(n+n'_2)} G_{2m} g_{2(n+n''_2)-1}, \ldots, \\ & G_{2(n+1)} g_1 G_{2n} g_{2n-1}, \ldots, G_2 g_1. \end{aligned}$$

According to $Sequ$, $G \cdot g(X)$ is defined as follows:

$$G \cdot g(X) = G_R \vee (g_{(r+1)\pm 2n_R} \oplus X \oplus G_{(t+1)\pm 2n_t}) \vee g_t,$$

in which, $r, t, n_R, n_t \in \mathbb{N}$; $1 \le r \le x$; $1 \le t \le y$; $1 < (r+1) \pm 2n_R \le y$; $1 < (t+1) \pm 2n_t \le x$.

Note that the subscripts of G_r ($r \in \{1, \ldots, z'\}$) and g_t ($t \in \{1, \ldots, z\}$) may repeatedly emerge due to the unequaled group number of the readers and tags. For instance, there are four reader groups and seven tag groups in the system, thereinto the readers in G_1 and the tags in g_5 have two optional addressable paths $[G_1 g_2 G_3 g_4 G_1 g_6 G_4 g_7 G_2 g_5]$ and $[G_1 g_6 G_4 g_7 G_2 g_5]$ for G_1 and g_5. Here, G_1 and g_5 apply the latter segment of $Sequ$ and $G \cdot g(X)$ to update the secret X:

$$\begin{aligned} Sequ_{(4<7)} &= G_1 g_2 G_3 g_4 [G_1 g_6 G_4 g_7 G_2 g_5] G_4 g_3 G_2 g_1, \\ G \cdot g(X) &= (G \cdot g)_{1 \cdot 5}(X) = G_1 \vee (g_6 \oplus X \oplus G_2) \vee g_5. \end{aligned}$$

3.2 Two-or-Multiple-Tag and Single-Reader Case

Fig. 1 shows interactions of R_j and $\{T_a, T_b\}$. Thereinto, R_j belongs to G_v, and $\{T_a, T_b\}$ belong to $\{g_m, g_n\}$.

Phase 1. Preliminary authentication between R_j and $\{T_a, T_b\}$: The reader R_j generates a pseudorandom number r_{0R_j}, and cascades r_{0R_j} and its identity flag F_{R_j}. R_j transmits $r_{0R_j} \| F_{R_j}$ to $\{T_a, T_b\}$ as a query to initiate a new session. Upon receiving the message, $\{T_a, T_b\}$ search F_{R_j} in L_R, and check the correctness of F_{R_j}. If there is a nonmatching flag or the flag with wrong time stamp, R_j will be regarded as an illegal reader and the protocol will terminate. Otherwise, the two tags will reply $\{F_{T_a}, F_{T_b}\}$ to R_j, respectively. Afterward, R_j performs quick check on $\{T_a, T_b\}$, and judges whether the tags are legal and which group they belong to. If both tags pass the quick check, $\{T_a, T_b\}$ will be linked via the channels to R_j, and the messages between the two tags will be sequentially exchanged.

Phase 2. T_a challenges R_j: When $\{T_a, T_b\}$ are linked, an initiator tag T_a updates F_{T_a} into F'_{T_a} by $(G \cdot g)_{v \cdot m}(F_{T_a})$ for $\{n_{v_1}, n_{m_1}\} \in \mathbb{N}$, and computes M_{T_a} and N_{T_a}

$$\begin{aligned} F'_{T_a} &= G_v \vee (g_{(v+1)\pm 2n_{v_1}} \oplus F_{T_a} \oplus G_{(m+1)\pm 2n_{m_1}}) \vee g_m, \\ M_{T_a} &= (PID_{T_a} \oplus F'_{T_a}) \vee r_{0R_j}, \\ N_{T_a} &= PRNG(F_{R_j} \vee PID_{T_a}). \end{aligned}$$

T_a transmits $M_{T_a} \| N_{T_a}$ to R_j. Upon receiving the message, R_j divides a preshared secret S_{jb} into $S^l_{jb} \| S^r_{jb}$ by r_{0R_j}. The partition method is as follows: 1) perform modulo operation on r_{0R_j} by l to obtain $d_0 = r_{0R_j} \pmod l$; 2) extract the higher and lower d_0 bits of S_{jb} as two partial fields S^l_{jb} and S^r_{jb}. Note that underflow should be considered, and zero is padded to the higher bits. Hereafter, R_j generates r_{1R_j}, obtains the updated F'_{T_b} by $(G \cdot g)_{v \cdot n}(F_{T_b})$ for $\{n_{v_2}, n_{n_1}\} \in \mathbb{N}$, and computes $A^1_{R_j}$ and $B^1_{R_j}$:

$$\begin{aligned} F'_{T_b} &= G_v \vee (g_{(v+1)\pm 2n_{v_2}} \oplus F_{T_b} \oplus G_{(n+1)\pm 2n_{n_1}}) \vee g_n, \\ A^1_{R_j} &= (PID_{T_b} \vee F_{R_j}) \oplus (S^l_{jb} + r_{1R_j}), \\ B^1_{R_j} &= (gid_n \oplus F'_{T_b}) \vee r_{1R_j}. \end{aligned}$$

Phase 3. Further authentication between R_j and T_b: R_j transmits $A^1_{R_j} \| B^1_{R_j} \| N_{T_a}$ to T_b. Afterward, T_b performs a partition operation on the preshared secret S_{bj} to obtain S^l_{bj} and S^r_{bj} by d_0. Note that $S_{jb} = S_{bj}$ theoretically holds; therefore, $S^l_{jb} \| S^r_{jb}$ should equal $S^l_{bj} \| S^r_{bj}$. Thereafter, T_b obtains the updated F'_{T_b}, performs an inverse operation to derive $r^{\ell}_{1R_j}$, and computes B_{T_b}:

$$r^{\ell}_{1R_j} = A^1_{R_j} \oplus (PID_{T_b} \vee F_{R_j}) - S^l_{bj},$$
$$B_{T_b} = (gid_n \oplus (G \cdot g)_{v \cdot n}(F_{T_b})) \vee r^{\ell}_{1R_j}.$$

T_b verifies R_j by checking $B_{T_b} \stackrel{?}{=} B^1_{R_j}$. If it does not hold, R_j will be regarded as an illegal reader and the protocol will terminate. Otherwise, T_b will compute S'^l_{bj}, C_{T_b}, M_{T_b}, and N_{T_b}:

$$S'^l_{bj} = (G \cdot g)_{v \cdot n}(S^l_{bj}) \oplus r^{\ell}_{1R_j},$$
$$C_{T_b} = S'^l_{bj} \vee PID_{T_b},$$
$$M_{T_b} = (PID_{T_b} \oplus F'_{T_b}) \vee r_{0R_j},$$
$$N_{T_b} = N_{T_a} \vee PRNG(PID_{T_b} \vee gid_n).$$

T_b transmits $C_{T_b} \| M_{T_b} \| N_{T_b}$ to R_j. Upon receiving the message, R_j computes $C^1_{R_j}$:

$$C^1_{R_j} = (G \cdot g)_{v \cdot n}(S^l_{jb}) \oplus r_{1R_j} \vee PID_{T_b}.$$

R_j verifies T_b by checking $C^1_{R_j} \stackrel{?}{=} C_{T_b}$. If it does not hold, R_j will regard T_b as an illegal tag and eliminate T_b from the authentication. Otherwise, R_j will continue to divide S_{ja} into $S^l_{ja} \| S^r_{ja}$ by r_{0R_j}, generate r_{2R_j}, and compute $A^2_{R_j}$ and $B^2_{R_j}$:

$$A^2_{R_j} = (PID_{T_a} \vee F_{R_j}) \oplus (S^r_{ja} + r_{2R_j}),$$
$$B^2_{R_j} = (gid_m \oplus (G \cdot g)_{v \cdot m}(F_{T_a})) \vee r_{2R_j}.$$

Phase 4. Further authentication between R_j and T_a: R_j transmits $A^2_{R_j} \| B^2_{R_j} \| N_{T_b}$ to T_a for further authentication. Similarly, T_a divides S_{aj} into $S^l_{aj} \| S^r_{aj}$, and derives $r^{\ell}_{2R_j}$ to compute B_{T_a}:

$$r^{\ell}_{2R_j} = A^2_{R_j} \oplus (PID_{T_a} \vee F_{R_j}) - S^r_{aj},$$
$$B_{T_a} = (gid_m \oplus F'_{T_a}) \vee r^{\ell}_{2R_j}.$$

T_a verifies R_j by checking $B_{T_a} \stackrel{?}{=} B^2_{R_j}$. If it does not hold, R_j will be regarded as an illegal reader and the protocol will terminate. Otherwise, T_a may consider that R_j is authorized, and compute S'^r_{aj}, C_{T_a}, and N'_{T_a}:

$$S'^r_{aj} = (G \cdot g)_{v \cdot m}(S^r_{aj}) \oplus r^{\ell}_{2R_j},$$
$$C_{T_a} = S'^r_{aj} \vee PID_{T_a},$$
$$N'_{T_a} = N_{T_b} \vee gid_m.$$

T_a transmits $C_{T_a} \| N'_{T_a}$ to R_j for authentication. When R_j receives the message, it computes $C^2_{R_j}$:

$$C^2_{R_j} = (G \cdot g)_{v \cdot m}(S^r_{ja}) \oplus r_{2R_j} \vee PID_{T_a}.$$

R_j continues to verify T_a by checking $C^2_{R_j} \stackrel{?}{=} C_{T_a}$. If it does not hold, R_j will regard T_a as an illegal tag and eliminate T_a from the authentication. Otherwise, R_j will establish the grouping proofs $P_{R_j}(T_{a,b}) = (r_{0R_j}, M_{T_a}, M_{T_b}, N'_{T_a})$. When R_j receives the subgrouping proofs from $\{T_a, T_b\}$, R_j invokes the final authentication to validate the grouping proofs $P_{R_j}(T_{a,b})$. Till now, $\{T_a, T_b\}$ have been simultaneously accessed by R_j, and the grouping proofs is verified as follows:

1. Verify $\{M_{T_a}, M_{T_b}\}$ by the tag pseudonyms $\{PID_{T_a}, PID_{T_b}\}$, and the updated tag flags $\{F'_{T_a}, F'_{T_b}\}$;
2. Verify N'_{T_a} by $\{PID_{T_a}, PID_{T_b}\}$, the tag group identifiers $\{gid_m, gid_n\}$, and the reader flag F_{R_j}.

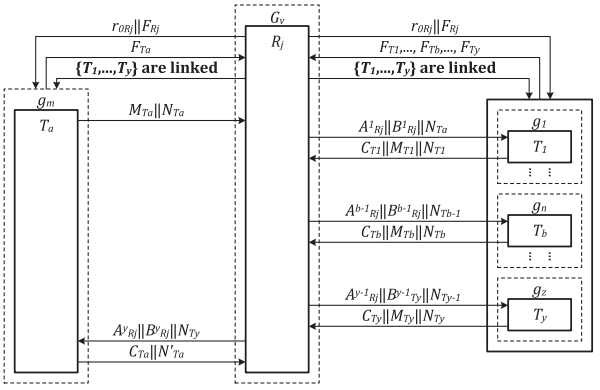


Fig. 2. GUPA: multiple-tag and single-reader case (nT-R).

Fig. 2 shows the extension of GUPA in the case of multiple tag and single reader (nT-R). Suppose that the reader R_j first challenges the tags $\{T_1, \ldots, T_y\}$ by $r_{0R_j} \| F_{R_j}$. After passing the mutual quick check, $\{T_1, \ldots, T_a, T_b, \ldots, T_y\}$ are linked together. T_a acts as an initiator, and transmits $M_{T_a} \| N_{T_a}$ to R_j. R_j proceeds to sequentially access the distributed tags $\{T_1, \ldots, T_b, \ldots, T_x\}$. During the quick check phase, R_j stores the flags $\{F_{T_a}, F_{T_1}, \ldots, F_{T_b}, \ldots, F_{T_y}\}$ into a temp queue; therefore, R_j challenges the tags $\{T_1, \ldots, T_b, \ldots, T_y\}$ by their corresponding flags in a first-in-first-out (FIFO) mode. R_j transmits $A^1_{R_j} \| B^1_{R_j} \| N_{T_a}$ to T_1, and T_1 replies $C_{T_1} \| M_{T_1} \| N_{T_1}$ to R_j, and so forth. R_j continues to transmit $A^{b-1}_{R_j} \| B^{b-1}_{R_j} \| N_{b-1}$ to T_b, and T_b replies $C_{T_b} \| M_{T_b} \| N_{T_b}$ to R_j. In the last round, R_j transmits $A^{y-1}_{R_j} \| B^{y-1}_{R_j} \| N_{T_{y-1}}$ to T_y, and T_y replies $C_{T_y} \| M_{T_y} \| N_{T_y}$ to R_j. Thereafter, R_j transmits $A^y_{R_j} \| B^y_{R_j} \| N_{T_y}$ to T_a, then T_a replies $C_{T_a} \| N'_{T_a}$ to R_j. The grouping proofs $P_{R_j}(T_{1,\ldots,a,b,\ldots,y})$ can be established:

$$P_{R_j}(T_{1,\ldots,a,b,\ldots,y}) = (r_{0R_j}, M_{T_1}, \ldots, M_{T_y}, N'_{T_a}).$$

3.3 Two-or-Multiple-Reader and Single-Tag Case

Fig. 3 shows interactions of T_a and $\{R_i, R_j\}$. Thereinto, $\{R_i, R_j\}$ belong to $\{G_u, G_v\}$, and T_a belongs to g_m.

Phase 1. Preliminary authentication between T_a and $\{R_i, R_j\}$: The readers $\{R_i, R_j\}$ generate pseudorandom numbers $\{r_{0R_i}, r_{0R_j}\}$, and transmit $r_{0R_i} \| F_{R_i}$ and $r_{0R_j} \| F_{R_j}$ to T_a to initiate a new session. Upon receiving the queries, T_a searches $\{F_{R_i}, F_{R_j}\}$ in L_R, and checks the correctness. If R_i or R_j has unmatched flag, R_i or R_j will be regarded as an illegal reader, and T_a will eliminate the illegal reader from the authentication. Otherwise, T_a will reply F_{T_a} to $\{R_i, R_j\}$, and link the two readers $\{R_i, R_j\}$ together. Afterward, $\{R_i, R_j\}$ perform quick check on T_a by its flag, and judge whether the tag is legal and which group it belongs to. If T_a passes the quick check, the protocol will continue. Note that "link" does not mean to establish direct communication channel between the two readers $\{R_i, R_j\}$, and the two readers do not need to know which reader they are linking with. The fact is that $\{R_i, R_j\}$ are linked by T_a which acts as the middleman to exchange the messages.

Phase 2. Further authentication on $\{R_i, R_j\}$: When $\{R_i, R_j\}$ are linked, R_i acts as a proof initiator, and extracts the higher and lower d_1 bits of S_{ia} as $S^l_{ia} \| S^r_{ia}$, in which

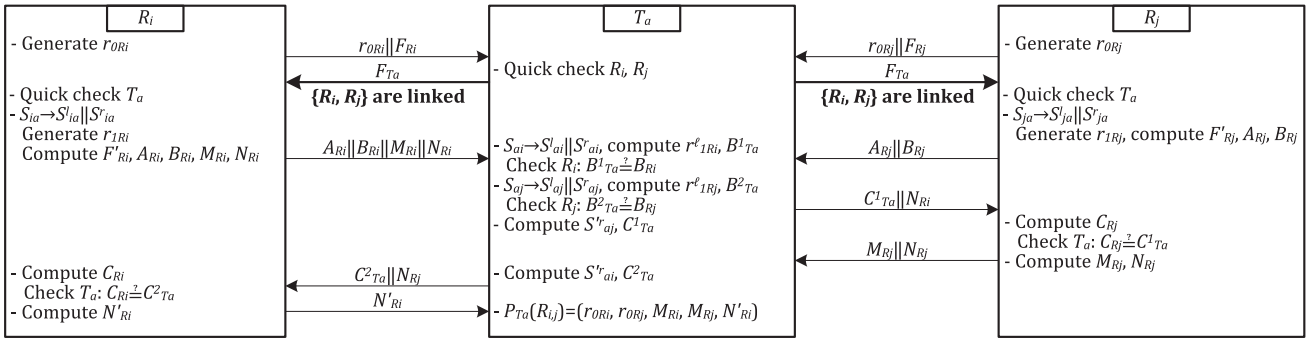


Fig. 3. GUPA: two-reader and single-tag case (2R-T).

$d_1 = r_{0R_i} \pmod l$. R_i generates a pseudorandom number r_{1R_i}, updates F_{R_i} by $(G \cdot g)_{u \cdot m}(F_{R_i})$ for $\{n_{u_1}, n_{m_2}\} \in \mathbb{N}$, and computes A_{R_i}, B_{R_i}, M_{R_i}, and N_{R_i}:

$$F'_{R_i} = G_u \vee (g_{(u+1)\pm 2n_{u_1}} \oplus F_{R_i} \oplus G_{(m+1)\pm 2n_{m_2}}) \vee g_m,$$
$$A_{R_i} = (PID_{T_a} \vee F_{T_a}) \oplus (S^l_{ia} + r_{1R_i}),$$
$$B_{R_i} = (GID_u \oplus F'_{R_i}) \vee r_{1R_i},$$
$$M_{R_i} = (PID_{R_i} \oplus F'_{R_i}) \vee r_{0R_i},$$
$$N_{R_i} = PRNG(F_{T_a} \vee PID_{R_i}).$$

R_j divides S_{ja} into $S^l_{ja}\|S^r_{ja}$ by $d'_0 = r_{0R_j} \pmod l$. R_j generates r_{1R_j}, updates F'_{R_j} by $(G \cdot g)_{v \cdot m}(F_{R_j})$ for $\{n_{v_3}, n_{m_3}\} \in \mathbb{N}$, and computes A_{R_j} and B_{R_j}:

$$F'_{R_j} = G_v \vee (g_{(v+1)\pm 2n_{v_3}} \oplus F_{R_j} \oplus G_{(m+1)\pm 2n_{m_3}}) \vee g_m,$$
$$A_{R_j} = (PID_{T_a} \vee F_{T_a}) \oplus (S^r_{ja} + r_{1R_j}),$$
$$B_{R_j} = (GID_v \oplus F'_{R_j}) \vee r_{1R_j}.$$

R_i transmits $A_{R_i}\|B_{R_i}\|M_{R_i}\|N_{R_i}$ to T_a, and R_j transmits $A_{R_j}\|B_{R_j}$ to T_a. Upon receiving the message, T_a divides S_{ai} into $S^l_{ai}\|S^r_{ai}$ by d_1. Hereafter, T_a derives $r^\ell_{1R_i}$, and computes $B^1_{T_a}$:

$$r^\ell_{1R_i} = A_{R_i} \oplus (PID_{T_a} \vee F_{T_a}) - S^l_{ai},$$
$$B^1_{T_a} = (GID_u \oplus (G \cdot g)_{u \cdot m}(F_{R_i})) \vee r^\ell_{1R_i}.$$

T_a verifies R_i by checking $B^1_{T_a} \stackrel{?}{=} B_{R_i}$. If it does not hold, T_a will regard R_i as an illegal reader and eliminate R_i from the authentication. Otherwise, T_a will extract $S^l_{aj}\|S^r_{aj}$ by d'_0. T_a derives $r^\ell_{1R_j}$, and computes $B^2_{T_a}$:

$$r^\ell_{1R_j} = A_{R_j} \oplus (PID_{T_a} \vee F_{T_a}) - S^r_{aj},$$
$$B^2_{T_a} = (GID_v \oplus (G \cdot g)_{v \cdot m}(F_{R_j})) \vee r^\ell_{1R_j}.$$

Similarly, T_a verifies R_j by comparing the computed $B^2_{T_a}$ with the received B_{R_j}. If they are not identical, T_a will regard R_j as an illegal reader and eliminates R_j from the authentication. Otherwise, T_a will obtain the updated S''^r_{aj} and compute $C^1_{T_a}$:

$$S''^r_{aj} = (G \cdot g)_{v \cdot m}(S^r_{aj}) \oplus r^\ell_{1R_j},$$
$$C^1_{T_a} = S''^r_{aj} \oplus PID_{T_a}.$$

Phase 3. Further authentication on T_a: T_a transmits $C^1_{T_a}\|N_{R_i}$ to R_j, thereafter, R_j computes C_{R_j}:

$$C_{R_j} = (G \cdot g)_{v \cdot m}(S^l_{ja}) \oplus r_{1R_j} \oplus PID_{T_a}.$$

R_j verifies T_a by checking $C_{R_j} \stackrel{?}{=} C^1_{T_a}$. If it does not hold, R_j will regard T_a as an illegal tag and terminates the protocol. Otherwise, R_j will compute M_{R_j} and N_{R_j}:

$$M_{R_j} = (PID_{R_j} \oplus F'_{R_j}) \vee r_{0R_j},$$
$$N_{R_j} = N_{R_i} \vee PRNG(PID_{R_j} \vee GID_v).$$

R_j transmits $M_{R_j}\|N_{R_j}$ to T_a. Hereafter, T_a updates S^r_{ai} into S''^r_{ai}, and computes $C^2_{T_a}$:

$$S''^r_{ai} = (G \cdot g)_{u \cdot m}(S^r_{ai}) \oplus r^\ell_{1R_i},$$
$$C^2_{T_a} = S''^r_{ai} \oplus PID_{T_a}.$$

Phase 4. Further authentication on T_a: T_a transmits $C^2_{T_a}\|N_{R_j}$ to R_i, and R_i computes C_{R_i}:

$$C_{R_i} = (G \cdot g)_{u \cdot m}(S^r_{ia}) \oplus r_{1R_i} \oplus PID_{T_a}.$$

R_i verifies T_a by checking $C_{R_i} \stackrel{?}{=} C^2_{T_a}$. If it does not hold, R_i will regard T_a as an illegal tag and terminate the protocol. Otherwise, R_i will compute N'_{R_i}:

$$N'_{R_i} = N_{R_j} \vee GID_u.$$

Thereafter, R_i transmits N'_{R_i} to T_a, and T_a establishes the grouping proofs $P_{T_a}(R_{i,j}) = (r_{0R_i}, r_{0R_j}, M_{R_i}, M_{R_j}, N'_{R_i})$. When T_a receives all the subgrouping proofs from $\{R_i, R_j\}$, T_a invokes the final authentication to validate the grouping proofs $P_{T_a}(R_{i,j})$. Till now, $\{R_i, R_j\}$ have simultaneously access T_a, and the grouping proofs is verified as follows:

1. Verify $\{r_{0R_i}, r_{0R_j}\}$ by checking whether the received random numbers appear in the former sessions within a certain time threshold.
2. Verify $\{M_{R_i}, M_{R_j}\}$ by $\{PID_{R_i}, PID_{R_j}\}$, and the updated reader flags $\{F'_{R_i}, F'_{R_j}\}$;
3. Verify N'_{R_i} by the reader pseudonyms $\{PID_{R_i}, PID_{R_j}\}$, the reader group identifiers $\{GID_u, GID_v\}$, and the tag flag F_{T_a}.

Fig. 4 shows the extension of GUPA in the case of multiple-reader and single-tag (nR-T), note that the available amount of the readers is limited by the channel resources. Suppose that the readers $\{R_1, \ldots, R_x\}$ concurrently challenge T_a by $\{r_{0R_1}\|F_{R_1}, \ldots, r_{0R_x}\|F_{R_x}\}$. After passing T_a's quick check, $\{R_1, \ldots, R_i, R_j, \ldots, R_x\}$ are linked together. R_i acts as a proof initiator, and transmits $A_{R_i}\|B_{R_i}\|M_{R_i}\|N_{R_i}$ to T_a. T_a replies the distributed readers $\{R_1, R_2, \ldots, R_x\}$ according to the FIFO mode. R_1 first transmits $A_{R_1}\|B_{R_1}$ to T_a, T_a replies $C^1_{T_a}\|N_{R_i}$ to R_1, and R_1

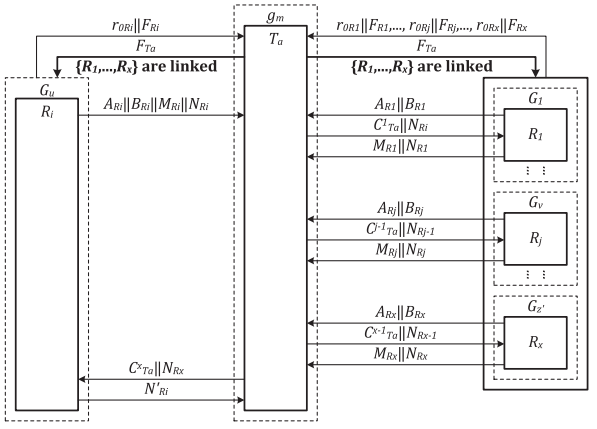


Fig. 4. GUPA: multiple-reader and single-tag case (nR-T).

transmits $M_{R_1}\|N_{R_1}$ to T_a, and so forth. R_j continues to transmit $A_{R_j}\|B_{R_j}$ to T_a, T_a replies $C_{T_a}^{j-1}\|N_{R_{j-1}}$ to R_j, then R_j replies $M_{R_j}\|N_{R_j}$ to T_a. In the last round, R_x transmits $A_{R_x}\|B_{R_x}$ to T_a, T_a replies $C_{T_a}^{x-1}\|N_{R_{x-1}}$ to R_x, and then R_x replies $M_{R_x}\|N_{R_x}$ to T_a. Thereafter, T_a transmits $C_{T_a}^{x}\|N_{R_x}$ to R_i, then R_i replies N'_{R_i} to T_a. The grouping proofs $P_{T_a}(R_1, \ldots, i, j, \ldots, x)$ can be established:

$$P_{T_a}(R_{1,\ldots,i,j,\ldots,x}) = (r_{0R_1}, \ldots, r_{0R_x}, M_{R_1}, \ldots, M_{R_x}, N'_{R_i}).$$

3.4 Multiple-Tag and Multiple-Reader Case

Fig. 5 shows the extension of GUPA in the case of multiple-tag and multiple reader (nR-nT), which is an infrequent communication case. Suppose that readers $\{R_1, \ldots, R_x\}$ and tags $\{T_1, \ldots, T_y\}$ are addressed in the system. $\{R_1, \ldots, R_x\}$ challenge $\{T_1, \ldots, T_y\}$ with the corresponding $\{r_{0R_1}\|F_{R_1}, \ldots, r_{0R_x}\|F_{R_x}\}$, then $\{T_1, \ldots, T_y\}$ respond with $\{F_{T_1}, \ldots, F_{T_y}\}$. After passing the quick check, $\{R_1, \ldots, R_x\}$ and $\{T_1, \ldots, T_y\}$ are independently linked together. We consider T_a and R_j to introduce the authentication. Let R_i and T_a act the initiators of the corresponding grouping proofs. R_i transmits $A_{R_i}\|B_{R_i}\|M_{R_i}\|N_{R_i}$ to T_a. T_a proceeds to reply the distributed readers $\{R_1, R_j, \ldots, R_x\}$ as the case descriptions of nR-T, and the grouping proofs $P_{T_a}(R_{1,\ldots,i,j,\ldots,x})$ is obtained by T_a. Meanwhile, T_a transmits $M_{T_a}\|N_{T_a}$ to R_j. R_j proceeds to access the distributed tags $\{T_1, \ldots, T_b, \ldots, T_x\}$ as the case descriptions of nT-R, and the grouping proofs $P_{R_j}(T_{1,\ldots,a,b,\ldots,y})$ can be obtained by R_j. In GUPA, $P_{T_*}(R_{1,\ldots,i,j,\ldots,x})$ and $P_{R_*}(T_{1,\ldots,a,b,\ldots,y})$ are established for secure and simultaneous identification among multiple readers and tags, in which $\{A_{R_*}, B_{R_*}, C_{T_*}\}$ are adopted with different functions:

- $\{A_{R_*}\}$ is used by the tag to derive the reader generated random number via the inverse operations.
- $\{B_{R_*}, C_{T_*}\}$ are used by the tag and reader to perform mutual authentication.
- $\{M_{R_*}\}$ is computed by the pseudonyms and flags for final verification.
- $\{N_{R_*,T_*}, N'_{R_*,T_*}\}$ are applied to correlate each pair of tags or readers. For instance, N_{R_i} computed by R_i is transmitted to R_j to obtain N_{R_j}. N_{R_j} computed by R_j is transmitted to R_i to obtain N'_{R_i}.

In the nR-nT case, the reader and tag have identical denial capability. It means that if any reader (or tag) regards the verified tag (or reader) as an illegal entity, the reader (or tag) will eliminate the suspect entity from the authentication and the protocol will continue.

4 Attack Analysis

4.1 Replay Attack

In replay attack, assume that $\mathcal{A}$ has learned all the exchanged messages of $\{R_i, T_a, \{R_1, \ldots, R_j, \ldots, R_x\}\}$ and $\{T_a, R_j, \{T_1, \ldots, T_b, \ldots, T_y\}\}$ in a former session. In another session, $\mathcal{A}$ may replay the intercepted messages to interfere with the ongoing session. Thereinto, $\mathcal{A}$ may act as two types of identities (i.e., an initiator entity, and a generic entity). In the former case, $\mathcal{A}$ acts as a tag or reader to challenge the verifier

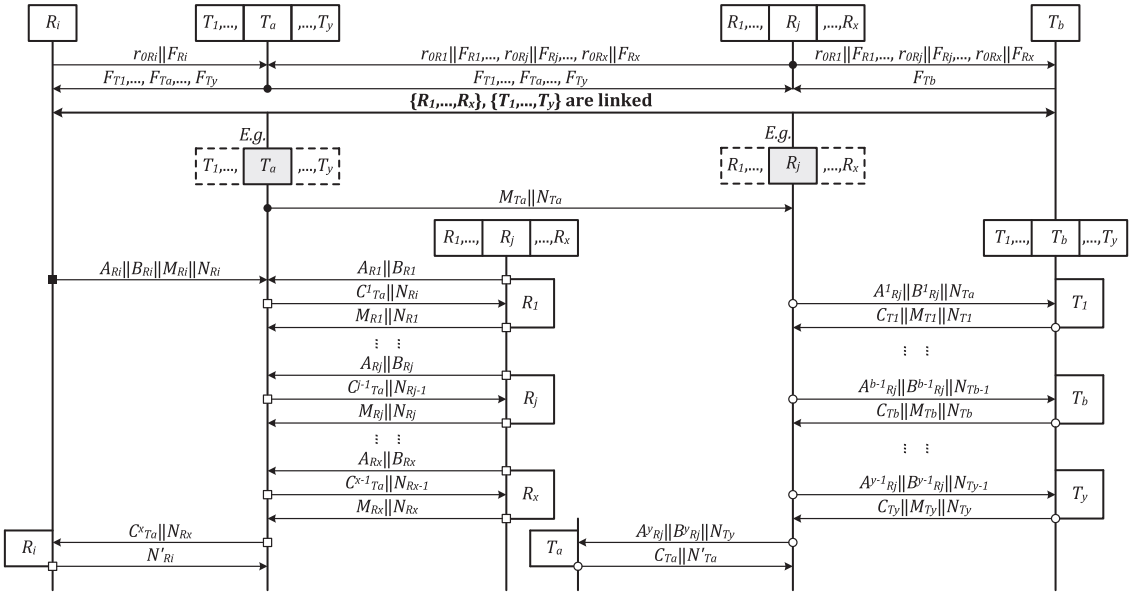


Fig. 5. GUPA: multiple-tag and multiple-reader case (nR-nT).

as a proof initiator. In the latter case, $\mathcal{A}$ acts as an unfeatured tag or reader that is mingled with other legal entities.

The identity flag with built-in time stamp is introduced to enhance session freshness. $\mathcal{A}$ acts as an illegal reader $\hat{R}_{\mathcal{A}}$ to challenge $\{T_1, \ldots, T_y\}$ with the outdated $r_{0R_*}^{old} \| F_{R_*}^{old}$, or acts as an illegal tag $\hat{T}_{\mathcal{A}}$ to challenge $\{R_1, \ldots, R_x\}$ with the outdated $F_{T_*}^{old}$. Upon receiving the messages, $\{T_1, \ldots, T_y\}$ or $\{R_1, \ldots, R_x\}$, first check the correctness of the identity flag. They will find that $F_{R_*}^{old}$ or $F_{T_*}^{old}$ has an unmatched time stamp (i.e., out of allowable time range), and eliminate $\mathcal{A}$ from the authentication. In a bad condition, the legal entities may ignore the error, and the protocol continues.

In the case of $\{R_i, T_a, \{R_1, \ldots, R_j, \ldots, R_x\}\}$, $\hat{R}_{\mathcal{A}}$ may replay an initiator reader R_i's $A_{R_i}^{old} \| B_{R_i}^{old} \| M_{R_i}^{old} \| N_{R_i}^{old}$ to T_a, and T_a derives $r_{1R_i}^{\ell new}$ to compute $B_{T_a}^{1new}$:

$$r_{1R_i}^{\ell new} = A_{R_i}^{old} \oplus (PID_{T_a} \vee F_{T_a}^{new}) - S_{ai}^{lold},$$
$$B_{T_a}^{1new} = (GID_u \oplus F'^{old}_{\hat{R}_i}) \vee r_{1R_i}^{\ell new}.$$

T_a finds that $B_{T_a}^{1new} \neq B_{\hat{R}_i}^{old} = (GID_u \oplus F'^{old}_{\hat{R}_i}) \vee r_{1R_i}^{old}$ since the updated $F_{T_a}^{new}$ is introduced to derive $r_{1R_i}^{\ell new}$, and the probability that $r_{1R_i}^{old}$ equals $r_{1R_i}^{\ell new}$ is negligible.

Meanwhile, $\hat{T}_{\mathcal{A}}$ may replay T_a's $C_{T_a}^{(j-1)old} \| N_{R_{j-1}}^{old}$ to R_j, and R_j computes

$$C_{R_j}^{new} = (G \cdot g)_{v \cdot m}(S_{ja}^{lnew}) \oplus r_{1R_j}^{new} \vee PID_{T_a}.R_j$$

find that $C_{R_j}^{new} \neq C_{T_a}^{(j-1)old}$ since $r_{1R_j}^{new}$ is introduced to update S_{ja}^{lnew}, and the probability that S_{ja}^{lnew} equals S_{ja}^{lold} is negligible. Similarly, in the case of $\{T_a, R_j, \{T_1, \ldots, T_b, \ldots, T_y\}\}$, inconsistencies will be deduced by introducing the updated identity flags and pseudorandom numbers.

4.2 Forgery Attack

In forgery attack, $\mathcal{A}$ may impersonate as a forged entity (i.e., $\hat{R}_{\mathcal{A}}$ or $\hat{T}_{\mathcal{A}}$) to utilize the forged messages to access system. Upon receiving $\{F_{R_{\mathcal{A}}}, F_{T_{\mathcal{A}}}\}$, $\{T_1, \ldots, T_y\}$ and $\{R_1, \ldots, R_x\}$, search the identify flags in their own access lists $\{L_R, L_T\}$ to perform the quick check, respectively, and it turns out that there are no unmatched flags, and $\{\hat{R}_{\mathcal{A}}, \hat{R}_{\mathcal{A}}\}$ are eliminated from the authentication.

Reader forging attack: An illegal reader $\hat{R}_{\mathcal{A}}$ tries to access the legal tags, in which the forged secrets $\{\widehat{PID}_T, \hat{S}\}$ are used to derive $r_{1\hat{R}_{\mathcal{A}}}^{\ell}$. For instance, $\hat{R}_{\mathcal{A}}$ acts as an initiator reader $\hat{R}_i$ to challenge T_a, and $\hat{R}_i$ computes $\hat{B}_{\hat{R}_i}$ for verification. Upon receiving $\hat{A}_{\hat{R}_i} \| \hat{B}_{\hat{R}_i} \| \hat{M}_{\hat{R}_i} \| \hat{N}_{\hat{R}_i}$, T_a derives $r_{1\hat{R}_i}^{\ell}$ to compute $B_{T_a}^{1}$:

$$r_{1\hat{R}_i}^{\ell} = \hat{A}_{\hat{R}_i} \oplus (PID_{T_a} \vee F_{T_a}) - S_{ai}^{l} = (\widehat{PID}_{T_a} \vee F_{T_a}) \oplus (\hat{S}_{ia}^{l} + r_{1\hat{R}_i}) \oplus (PID_{T_a} \vee F_{T_a}) - S_{ai}^{l}.$$

T_a finds that $\hat{B}_{\hat{R}_i} = (\widehat{GID}_u \oplus \hat{F}'_{\hat{R}_i}) \vee r_{1\hat{R}_i} \neq B_{T_a}^{1}$ since $B_{T_a}^{1}$ is computed by $\{GID_u, r_{1\hat{R}_i}^{\ell}\}$.

Tag forging attack: An illegal tag $\hat{T}_{\mathcal{A}}$ tries to cheat the legal readers, and $\{\widehat{PID}_{\hat{T}_b}, \hat{S}\}$ are also applied for verification. For instance, $\hat{T}_{\mathcal{A}}$ acts as a generic tag $\hat{T}_b$, which is queried by R_j. Upon receiving R_j's $A_{R_j}^{b-1} \| B_{R_j}^{b-1} \| N_{T_{b-1}}$, $\hat{T}_b$ directly computes $\hat{C}_{\hat{T}_b}$, and transmits $\hat{C}_{\hat{T}_b} \| \hat{M}_{\hat{T}_b} \| \hat{N}_{\hat{T}_b}$ to R_j. Thereafter, R_j computes $C_{R_j}^{b-1}$ for authentication. R_j finds that $\hat{C}_{\hat{T}_b} = \hat{S}_{bj}^{nl} \vee \widehat{PID}_{\hat{T}_b} \neq C_{R_j}^{b-1}$ since $C_{R_j}^{b-1}$ is obtained by $S_{jb}^{nl} \vee PID_{T_b}$, which are never exposed.

4.3 Tracking Attack

The tracking attack is a passive attack that the attacker traces an entity's location by multiple malicious devices.

Scenario 1: In a series of sessions, $\mathcal{A}$ disguises as a set of malicious readers $\hat{R} = \{\hat{R}^1, \hat{R}^2, \ldots\}$ to continuously challenge $\{T_1, \ldots, T_y\}$ with the queries $\{r_{0\hat{R}^1} \| \hat{F}_{\hat{R}^1}, r_{0\hat{R}^2} \| \hat{F}_{\hat{R}^2} \ldots\}$, to monitor traffic flows, and tries to analyze their location information. The protocol will terminate since $\{T_1, \ldots, T_y\}$ cannot recognize $\hat{R}$ for nonmatching flags $\{\hat{F}_{\hat{R}^1}, \hat{F}_{\hat{R}^2} \ldots\}$,. In a bad condition, $\{T_1, \ldots, T_y\}$ may ignore the error, and the protocol will continue. In one site, $\{T_1, \ldots, T_y\}$ respond with $\{F_{T_1^1}, \ldots, F_{T_y^1}\}$. In another site, $\{T_1, \ldots, T_y\}$ respond with $\{F_{T_1^2}, \ldots, F_{T_y^2}\}$, and so forth. Any two responses are independent since the flags are randomly chosen from the pseudonym index. $\hat{R}$ cannot confirm which tag the response belongs to since the tags' responses will be updated in each session.

Scenario 2: In another series of sessions, $\mathcal{A}$ disguises as malicious tags $\hat{T} = \{\hat{T}^1, \hat{T}^2, \ldots\}$, and $\{R_1, \ldots, R_x\}$ challenge the tag set $\hat{T}$. In one site, $\{R_1, \ldots, R_x\}$ transmit $\{r_{0R_1^1} \| F_{R_1^1}, \ldots, r_{0R_x^1} \| F_{R_x^1}\}$ to $\hat{T}$. In another site, $\{R_1, \ldots, R_x\}$ transmit $\{r_{0R_1^2} \| F_{R_1^2}, \ldots, r_{0R_x^2} \| F_{R_x^2}\}$ to $\hat{T}$, and so forth. Similarly, any two queries are independent since the pseudorandom numbers and flags are introduced. $\hat{T}$ cannot confirm which reader challenges the specific query. Thereafter, $\{\hat{T}^1, \hat{T}^2 \ldots\}$, respond with $\{\hat{F}_{\hat{T}^1}, \hat{F}_{\hat{T}^2}, \ldots\}$ to $\{R_1, \ldots, R_x\}$. The protocol will terminate since $\{R_1, \ldots, R_x\}$ cannot recognize $\hat{T}$ for nonmatching flags $\{\hat{F}_{\hat{T}^1}, \hat{F}_{\hat{T}^2}, \ldots\}$ in L_T. The attacker is incapable of tracking a specific tag according to the pseudorandom responses.

4.4 DoP

The DoP attack is executed by injecting illegal entities into the communication among legal entities, which may cause the grouping proofs invalid. Suppose that an attacker $\mathcal{A}$ (i.e., $\hat{R}_{\mathcal{A}}$, $\hat{T}_{\mathcal{A}}$) could pass the quick check, and all the legal/illegal entities can be linked together.

Scenario 1: T_a acts as an initiator tag, and transmits $M_{T_a} \| N_{T_a}$ to $\hat{R}_{\mathcal{A}}$. $\hat{R}_{\mathcal{A}}$ proceeds to transmit $\hat{A}_{\hat{R}_{\mathcal{A}}}^{1} \| \hat{B}_{\hat{R}_{\mathcal{A}}}^{1} \| N_{T_a}$ to T_1, and T_1 finds $B_{T_1} \neq \hat{B}_{\hat{R}_j}^{1}$. Similarly, the other tags $\{T_2, \ldots, T_a, T_b, \ldots, T_y\}$ will deduce the inconsistency. In another condition, $\hat{R}_{\mathcal{A}}$ may access a certain tag T_a along with other legal readers $\{R_1, \ldots, R_j, \ldots, R_x\}$ as follows: 1) $\hat{R}_{\mathcal{A}}$ acts as an initiator reader, and transmits $\hat{A}_{\hat{R}_{\mathcal{A}}} \| \hat{B}_{\hat{R}_{\mathcal{A}}} \| \hat{M}_{\hat{R}_{\mathcal{A}}} \| \hat{N}_{\hat{R}_{\mathcal{A}}}$ to T_a. T_a finds $B_{T_a}^{1} \neq \hat{B}_{\hat{R}_{\mathcal{A}}}$. Thereafter, another reader R_1 acts as an initiator, and the protocol continues; or 2) R_i acts as an initiator reader to challenge T_a; thereafter, T_a and $\{R_1, \ldots, \hat{R}_{\mathcal{A}}, \ldots, R_x\}$ orderly perform mutual verifications. Toward $\hat{R}_{\mathcal{A}}$, T_a receives $\hat{A}_{\hat{R}_{\mathcal{A}}} \| \hat{B}_{\hat{R}_{\mathcal{A}}}$, and finds $B_{T_a}^{j} \neq \hat{B}_{\hat{R}_{\mathcal{A}}}$.

Scenario 2: R_i acts as an initiator reader to challenge $\hat{T}_{\mathcal{A}}$, and a generic tag R_1 transmits $A_{R_1} \| B_{R_1}$ to $\hat{T}_{\mathcal{A}}$. $\hat{T}_{\mathcal{A}}$ skips the verification, and replies $C_{\hat{T}_{\mathcal{A}}}^{1} \| N_{R_i}$. R_1 finds $C_{R_1} \neq \hat{C}_{\hat{T}_{\mathcal{A}}}^{1}$. Similarly, $\{R_2, \ldots, R_x\}$ will also find the inconsistency. In another condition, $\hat{T}_{\mathcal{A}}$ may establish communication with R_j along with other legal tags $\{T_1, \ldots, T_b, \ldots, T_y\}$ as follows: 1) $\hat{T}_{\mathcal{A}}$ acts as an initiator tag, and transmits $\hat{M}_{\hat{T}_{\mathcal{A}}} \| \hat{N}_{\hat{T}_{\mathcal{A}}}$ to R_j. R_j proceeds to transmit $A_{R_j}^{1} \| B_{R_j}^{1} \| N_{T_a}$ to T_1, and T_1 replies $C_{T_1} \| M_{T_1} \| N_{T_1}$ to R_j, and so forth. During the message delivery, R_j and $\{T_1, \ldots, T_b, \ldots, T_y\}$ complete mutual verifications. R_j transmits $A_{R_j}^{y} \| B_{R_j}^{y} \| N_{T_y}$ to $\hat{T}_{\mathcal{A}}$, and $\hat{T}_{\mathcal{A}}$ replies $\hat{C}_{\hat{T}_{\mathcal{A}}} \| \hat{N}'_{\hat{T}_{\mathcal{A}}}$ to R_j. R_j finds $C_{R_j}^{y} \neq \hat{C}_{\hat{T}_{\mathcal{A}}}$; or 2) T_a acts as an

TABLE 2
Performance Comparison among Related Protocols

		Burmester's P3 [10]	Kazahaya [11]	OTSBP [12]	Cho's [13]	GUPA: 2T-R	GUPA: 2R-T
SR	T	$6l$	$4l$	$3l$	$2l$	$(3+2z')l$	
	$R(DB)$	$(2+y)l$	$1l$	$1l$	$(1+y)l$	$(3+2z+y)l$	
CO	$T \rightarrow R(DB)$	4	3	3	3	5	4
	$R(DB) \rightarrow T$	5	3	3	2	4	6
	Total Packet	14	15	17	9	19	19
	Round Number	6	6	8	6	8	8
CL	T_a	2R+3H	13B+13R	4B+4R+2M	1R+1M+1E	19B+R	30B
	T_b	2R+3H	7B+10R	4B+4R+2M	1M	19B+R	–
	$R_i(DB)$	1R	12B+13R	1H	2R+2E	30B+3R	19B+3R
	$R_j(DB)$	–	–	–	–	–	19B+3R

B: Bitwise function; R: PRNG function; M: MAC function; H: Hash function; E: Encryption; l: Length of identifier.
x: The reader number; y: The tag number; z': The reader group number; z: The tag group number, $\{z', z\} \ll x \ll y$.

initiator tag, and transmits $M_{T_a} \| N_{T_a}$ to R_j. R_j transmits $A^1_{R_j} \| B^1_{R_j} \| N_{T_a}$ to T_1, and T_1 replies $C_{T_1} \| M_{T_1} \| N_{T_1}$ to R_j, and so forth. Toward $\hat{T}_{\mathcal{A}}$, R_j transmits $A^{b-1}_{R_j} \| B^{b-1}_{R_j} \| N_{T_{b-1}}$ to $\hat{T}_{\mathcal{A}}$, and $\hat{T}_{\mathcal{A}}$ replies $\hat{C}_{\hat{T}_{\mathcal{A}}} \| \hat{M}_{\hat{T}_{\mathcal{A}}} \| \hat{N}_{\hat{T}_{\mathcal{A}}}$. R_j finds $C^{b-1}_{R_j} \neq \hat{C}_{\hat{T}_{\mathcal{A}}}$.

In GUPA, the subgrouping proof is verified independent, which is not affected by each other based on the distributed structure. Even if the illegal proof exists, it will be eliminated from the authentication and will not influence the legal entities' proofs.

5 Performance Analysis

Table 2 shows the performance comparison between the related protocols.

Toward storage requirement (SR), a tag mainly stores $\{PID_T, F_T, gid, GID_{1,\ldots,z'}, \{S\}_{z'}\}$, and a reader mainly stores $\{PID_R, F_R, GID, PID_{T_{1,\ldots,y}}, gid_{1,\ldots,z}, \{S\}_z\}$. Compared with other protocols, GUPA needs $(3+2z')l$ units tag storage, which is a little lager due to the additional reader group identifiers. The reader SR is $(3+2z+y)l$, which is comparable with protocols in [10], [13]. Toward [11] and [12], the least reader storage is defined, while the lightweight storage is achieved by ignoring the intermediate verification.

Considering communication overhead (CO), the average exchanged data packets are 19 units in a session. Thereinto, $\{T_a, T_b\}$ transmit messages to the reader in five steps in the two-tag and single-reader case (2T-R), and $\{R_i, R_j\}$ to the tag via 4 steps; $\{T_a, T_b\}$ transmit messages to the reader in four steps in the two-reader and single-tag case (2R-T), and $\{R_i, R_j\}$ to the tag via six steps. The purpose of compromise on CO is introducing the flags $\{F_R, F_T\}$ for preliminary authentications. The mutual authentication completes in eight rounds, which is considered as a moderate number.

During the initialization phase, the main CL is brought by the ring signature, which is used to verify the new readers. Besides, the access list updating is performed based on PRNG and hash functions. During the protocol execution phase, grouping proofs are used for verification. In the two-tag and single-reader case, T_a/T_b performs 19 bitwise functions (i.e., XOR, OR) and 1 PRNG function, and R_j performs 30 bitwise functions and three PRNG functions. In the two-reader and single-tag case, R_i/R_j performs 19 bitwise functions and three PRNG functions, and T_a performs 30 bitwise functions. According to [20], the PRNG function needs about $10K$ logic gates, secure hash algorithm (SHA-1) needs less than $9K$ logic gates, and advanced encryption standard needs about $12K$ logic gates. Compared with the protocols, which are based on hash function in [10], MAC function in [12], [13], and encryption [13], the CL of GUPA is lightweight for a pervasive computing environment. Note that the protocols in [10], [11], [12], [13] are not competent for the cases when two or more readers perform secure and simultaneous identification on the single tag. Furthermore, the allocation proportion of workload for the readers and tags is more reasonable than the protocols in [10], [11], [12]. In these protocols, the reader mainly acts as a middle transmitter, and it actually executes less CL than tags, which is not reasonable from the hardware consideration. It turns out that GUPA applies the lightweight bitwise and PRNG functions to achieve security without using complicated algorithms, which makes GUPA is appropriate for resource-constrained systems.

6 Conclusion

In the paper, we have presented an authentication protocol (GUPA) for distributed RFID systems. The protocol applies grouping proofs to realize multiple readers and tags secure and simultaneous identification. The distributed authentication mode assigns tags into diverse groups to enhance hierarchical protection, and to achieve independent subgrouping proofs. The asymmetric denial scheme grants the entity diverse denial capabilities. It indicates that GUPA owns substantial advantages for lightweight RFID applications.

Acknowledgments

This work is jointly funded by National Natural Science Foundation of China (NSFC) and Civil Aviation Administration of China (CAAC) (61079019), and is partly supported by the National Natural Science Foundation of China (NSFC) (61071071). This work is also supported by the Fundamental Research Funds for the Central Universities.

References

[1] Y. Zuo, "Survivable RFID Systems: Issues, Challenges, and Techniques," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Rev.*, vol. 40, no. 4, pp. 406-418, July 2010.

[2] K. Ren, S. Yu, W. Lou, and Y. Zhang, "PEACE: A Novel Privacy-Enhanced yet Accountable Security Framework for Metropolitan Wireless Mesh Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 2, pp. 203-215, Feb. 2010.

[3] H.Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," *IEEE Trans. Dependable and Secure Computing,* vol. 4, no. 4, pp. 337-340, Oct.-Dec. 2007.

[4] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight Mutual Authentication and Ownership Transfer for RFID Systems," *Proc. IEEE INFOCOM,* pp. 1-5, 2010.

[5] M. Burmester and J. Munilla, "Lightweight RFID Authentication with Forward and Backward Security," *ACM Trans. Information and System Security,* vol. 14, no. 1, 2011.

[6] L. Zhu and T.S.P. Yum, "Optimal Framed Aloha Based Anti-Collision Algorithms for RFID Systems," *IEEE Trans. Comm.,* vol. 58, no. 12, pp. 3583-3592, Dec. 2010.

[7] T.F.L. Porta, G. Maselli, and C. Petrioli, "Anti-Collision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization," *IEEE Trans. Mobile Computing,* vol. 10, no. 2, pp. 267-279, Feb. 2011.

[8] A. Juels, "'Yoking-Proofs' for RFID Tags," *Proc. Second IEEE Ann. Conf. Pervasive Computing and Comm. Workshops,* pp. 138-143, 2004.

[9] J. Saito and K. Sakurai, "Grouping Proof for RFID Tags," *Proc. 19th Int'l Conf. Advanced Information Networking and Applications (AINA '05),* pp. 621-624, 2005.

[10] M. Burmester, B. Medeiros, and R. Motta, "Provably Secure Grouping-Proofs for RFID Tags," *Proc. Eighth IFIP WG 8.8/11.2 Int'l Conf. Smart Card Research and Advanced Applications,* Sept. 2008.

[11] P. Peris-Lopez, A. Orifila, J.C. Hernandez-Castro, and J.C.A. Lubbe, "Flaws on RFID Grouping-Proofs. Guidelines for Future Sound Protocols," *J. Network and Computer Applications,* vol. 34, pp. 833-845, 2011.

[12] N.W. Lo and K.H. Yeh, "Anonymous Coexistence Proofs for RFID Tags," *J. Information Science and Eng.,* vol. 26, no. 4, pp. 1213-1230, 2010.

[13] J.S. Cho, S.S. Yeo, S. Hwang, S.Y. Rhee, and S.K. Kim, "Enhanced Yoking Proof Protocols for RFID Tags and Tag Groups," *Proc. 22nd Int'l Conf. Advanced Information Networking and Applications Workshops (AINAW '08),* pp. 1591-1596, 2008.

[14] H.H. Huang and C.Y. Ku, "A RFID Grouping Proof Protocol for Medication Safety of Inpatient," *J. Medical Systems,* vol. 33, no. 6, pp. 467-474, 2009.

[15] H.Y. Chien, C.C. Yang, T.C. Wu, and C.F. Lee, "Two RFID-Based Solutions to Enhance Inpatient Medication Safety," *J. Medical Systems,* vol. 35, no. 3, pp. 369-375, 2009.

[16] P. Peris-Lopez, A. Orfilaa, A. Mitrokotsaa, and J.C. van der Lubbea, "A Comprehensive RFID Solution to Enhance Inpatient Medication Safety," *Int'l J. Medical Informatics,* vol. 80, pp. 13-24, 2011.

[17] Y. Lien, X. Leng, K. Mayes, and J. Chiu, "Reading Order Independent Grouping Proof for RFID Tags," *Proc. IEEE Int'l Conf. Intelligence and Security Informatics (ISI '08),* pp. 128-136, June 2008.

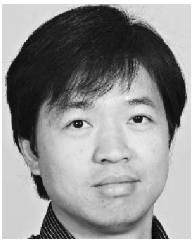
[18] X. Lin, R. Lu, H. Zhu, P.H. Ho, X. Shen, and Z. Cao, "ASRPAKE: An Anonymous Secure Routing Protocol with Authenticated Key Exchange for Wireless Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '07),* 2007.

[19] D. He, J. Bu, S. Zhu, S. Chan, and C. Chen, "Distributed Access Control with Privacy Support in Wireless Sensor Networks," *IEEE Trans. Wireless Comm.,* vol. 10, no. 10, pp. 3472-3481, Oct. 2011.

[20] Security Solutions Based on Top-Level Security IP and Software Applications - Products: Crypto Cores, http://www.intrinsic-id.com/cryptocores.htm, 2013.

Hong Liu is currently working toward the PhD degree at the School of Electronic and Information Engineering, Beihang University, China. She focuses on the security and privacy issues in radio frequency identification, vehicle-to-grid, and wireless machine-to-machine networks. Her research interests include authentication protocol design, and security formal modeling and analysis. She is a student member of the IEEE.

Huansheng Ning received the BS degree from Anhui University in 1996 and the PhD degree from Beihang University in 2001. He is currently an associate professor at the School of Electronic and Information Engineering, Beihang University, China. His current research focuses on Internet of things, aviation security, electromagnetic sensing and computing. He has published more than 30 papers in journals, international conferences/workshops. He is a senior member of the IEEE and the IEEE Computer Society.



Yan Zhang received the PhD degree from Nanyang Technological University, Singapore. From August 2006, he has been with Simula Research Laboratory, Norway. He is currently a senior research scientist at Simula Research Laboratory, Norway. He is an adjunct associate professor at the University of Oslo, Norway. He is a regional editor, associate editor, on the editorial board, or guest editor of a number of international journals. He is currently serving the Book Series Editor for the book series on *Wireless Networks and Mobile Communications* (Auerbach Publications, CRC Press, Taylor & Francis Group). He serves as organizing committee chairs for many international conferences. His research interests include resource, mobility, spectrum, energy, data, and security management in wireless communications and networking. He is a senior member of the IEEE and the IEEE Computer Society.

Daojing He received the BEng and MEng degrees in computer science from the Harbin Institute of Technology in 2007 and 2009, respectively. He is currently working toward the PhD degree in the Department of Computer Science, Zhejiang University, P.R. China. His research interests include network and systems security. He is the technical program committee member of many international conferences. He is a member of the IEEE.

Qingxu Xiong received the PhD degree in electrical engineering from Peking University, Beijing, China, in 1994. From 1994 to 1997, he was a postdoctoral researcher in the Information Engineering Department, Beijing University of Posts and Telecommunications. He is currently a professor at the School of Electrical and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing, China. His research interests include scheduling in optical and wireless networks, performance modeling of wireless networks, satellite communication. He is a member of the IEEE and the IEEE Computer Society.

Laurence T. Yang received the PhD degree in computer science from the University of Victoria, Canada. He is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology, Wuhan, China, and in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing. His research is supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation. He is a member of the IEEE and the IEEE Computer Society.