www.ietdl.org

Published in IET Information Security
Received on 4th August 2012
Revised on 3rd September 2013
Accepted on 1st November 2013
doi: 10.1049/iet-ifs.2012.0241

ISSN 1751-8709

Group-based key array authentication protocol in radio frequency identification systems

Yi Jiang[1], *Wei Cheng*[1], *Xiaojiang Du*[2]

[1]*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, People's Republic of China*
[2]*College of Science and Technology, Temple University, Philadelphia, PA, USA*
E-mail: jiangyiv88@nwpu.edu.cn

Abstract: For the purposes of information security and privacy between readers and tags, identity authentication is a significant issue for radio frequency identification (RFID) systems. In this study, the authors propose a novel security group-based key array authentication protocol, which is suitable for a large scale RFID environment. Based on a key array, this protocol can generate an authentication key for each pair of reader and tag with lower storage. Adding an identifier update phase, they design the authentication process passing the formal analysis from GNY. The security and performance analysis results show that the protocol they present can achieve better security than previous protocols in resisting external and internal attacks, with lower storage and acceptable communication and computation load.

1 Introduction

Radio frequency identification (RFID) systems is a contactless automatic identification system that provide contact-free communication between a reader and a tag over a radio link. Recently, the wide deployment of RFID systems in a variety of applications has raised many concerns about security and privacy. RFID systems must ensure the security of communication data as well as solve the identity authentication issue among entities.

To resolve the security issue successfully, several authentication protocols have been proposed, most of which focus on the threats from the external illegal attackers, but ignore the attacks from the internal legal entities. Hash chain protocol [1], which uses two different hash functions to confirm identity, can be used to achieve increased security. challenge-respond based RFID authentication protocol (HIDVP) [2] keeps track of its session number and is designed to prevent eavesdropping and replay attacks by diversifying values. Moessner and Gul [3] offered a high level of security through the combination of a random key scheme with a strong cryptography. Zhou *et al.* [4] proposed a lightweight anti-desynchronisation privacy preserving authentication protocol which is suitable for the low-cost environment. However, all the previous papers [1–4] do not consider the internal attacks. The method suggested by Karthikeyan and Nesterenko [5], based on simple exclusive-or (XOR) operation and matrix operation, cannot resist the external attacks. It does not support the authentication of multiple readers. Another method, used by Yang *et al.* [6], in which tags only have hash function and exclusive-or operation, can improve the abilities of the forgery and anonymity attacks. However, it was pointed out that the scheme cannot protect privacy. Chien and Chen [7] proposed a mutual authentication scheme appropriate for low cost tags, but it cannot resist denial of service (DoS) attacks if the synchronisation between the tag and reader is lost. Kolias *et al.* [8] improved the protocol [7] by enabling readers and tags to communicate securely and by providing resistance against DoS attack. Ding *et al.* [9] proposed a method which shares a key from key array to resolve the internal attacks issue, but they do not provide a proof for correctness. Key array authentication protocol (KAAP) [10] is an extension to the protocol [9], performing informal security analysis, but it cannot solve the internal attack in the same group of readers or tags, and does not include an update operation.

In this paper, we propose a novel security authentication protocol GKAAP (group-based KAAP), which is suitable for large scale RFID systems. It has the best capacity to authenticate communication. To deal with the internal attacks, we design an authentication key generation method using a key array based on [11], which differs from the ones proposed in [9–10]. Using this method, we define the authentication process to prevent both the external and internal attacks. Then, to increase the ability to protect against attacks, the pseudorandom identifier update method is used. Considering the correctness of our protocol, we use GNY logic [12] to carry out the formal analysis. The security and performance analysis results show that the protocol we present outperforms previous protocols in security with lower storage and acceptable communication and computation load, while also resisting various attacks such as forgery, replaying, tracking, DoS and internal attacks stemming from both the same group and between different groups.

The remainder of the paper is organised as follows. We describe our proposed protocol in Section 2. Section 3 analyses formally the correctness of our protocol with GNY logic. Section 4 analyses the performance of our protocol. Finally, our concluding remarks are stated in Section 5.

2 Proposed GKAAP protocol

In order to resolve such security and privacy problems from both the external and internal attacks, especially in the same group, the GKAAP protocol is proposed. It is an extension to the protocol KAAP [10] which has a key in the same group to resist the attacks among groups.

In our protocol, two types of keys are used for encryption: the shared key and the authentication key. A unique shared key k_u is given to legal readers and tags preventing external attacks. A key array D composed of authentication keys in the DB is used in security authentication between internal tags and readers.

2.1 Generation of the authentication key based on group

Suppose that tags are expressed as $T_i, i \in 0, 1, 2, \dots, m$ and the readers are expressed as $R_j, j \in 0, 1, 2, \dots, n$. Then we define D as the key generation array, A as the generation array of tags with size $m \times n$ and G as the generation array of readers with size $n \times n$. Suppose that $D = A \times G$, where the dimension of array D is $m \times n$. The expressions of A, G and D are

$$A = \begin{bmatrix} 1 & a_1 & {a_1}^2 & \dots & {a_1}^{n-1} \\ 1 & a_2 & {a_2}^2 & \dots & {a_2}^{n-1} \\ & & \dots & & \\ 1 & a_m & {a_m}^2 & \dots & {a_m}^{n-1} \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 1 & & 1 \\ g_1 & g_2 & & g_n \\ {g_1}^2 & {g_2}^2 & \dots & {g_n}^2 \\ \dots & \dots & & \dots \\ {g_1}^{n-1} & {g_2}^{n-1} & & {g_n}^{n-1} \end{bmatrix}$$

$$D = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2n} \\ & & \dots & & \\ k_{m1} & k_{m2} & k_{m3} & \dots & k_{mn} \end{bmatrix}$$

T_i stores the ith row in A and R_j stores the jth column in G. The authentication key is k_{ij} in D. As the generation of rows in A is regular, T_i can only store a_i, which can decrease the storage space greatly. It is the same as R_j which stores g_j. D in database DB is used to prove the correctness of the authentication key by computing between T_i and R_j. If some tags and readers are not permitted to communicate, the corresponding key in D is null. The generation of the authentication key between T_i and R_j is described by the following equation

$$k_{ij} = 1 + a_i g_j + (a_i g_j)^2 + \cdots + (a_i g_j)^{n-1} \qquad (1)$$

Based on the different functions, all tags and readers are divided into S and T groups, respectively, in which T_i belongs to the sth tag group A_s, $s \in 1, 2, \dots, S$ and R_j belongs to the tth read group G_t, $t \in 1, 2, \dots, T$. Different read groups own relative independent authorities for each tag groups, using different authentication key array D_{st} in DB. If one of the reader groups is not permitted to access a tag group, the corresponding key array is non-existent in DB, so the authentication cannot be carried out.

Overall, the different tags in the same group have different authentication keys with the different readers in the same group, which can defend against the internal attacks in groups. By group classification, it is fit for multiple applications against unauthorised access. The authentication key must be computed with the corresponding a and g stored in the tag and reader, which can prevent losing keys if an attacker captures a tag and obtains all of its sensitive message, including the authentication key.

2.2 Group-based key array authentication protocol

In the initialisation of GKAAP, it will allot an ID to each tag and reader in order to make communication between the two more convenient. The true ID can be utilised by replay and tracking attacks, so we use pseudorandom identifier preID = CRC(ID) to substitute it.

In order to describe the authentication phases between a tag and a reader, we use T_i and R_j to illustrate the process of message exchanges. They are as follows:

1. *Phase 1:* R_j generates a random number r_{R_j}, then concatenates r_{R_j}, preID_{R_j}, G_t and g_j to form $r_{R_j} \| \text{preID}_{R_j} \| G_t \| g_j$. Encrypting using the shared key k_u, R_j sends the message $\{r_{R_j} \| \text{preID}_{R_j} \| G_t \| g_j\}_{k_u}$ to T_i as an initial query.
2. *Phase 2*: On receiving the query, T_i decrypts the ciphertext $\{r_{R_j} \| \text{preID}_{R_j} \| G_t \| g_j\}_{k_u}$ using k_u. When T_i obtains r_{R_j}, preID_{R_j}, G_t and g_j, it will do several jobs as follows:
   - T_i verifies R_j by checking G_t in the list of permitted access reader groups which is prestored in its memory. If it is valid, T_i will compute the authentication key k_{ij}. Otherwise, it will stop the authentication process with an error code.
   - T_i uses a_i belonging to the tag group A_s and g_j to compute the authentication key k_{ij} with formula (1).
   - T_i generates a random number r_{T_i}, and encrypts $r_{T_i} \| r_{R_j}$ by k_{ij} to obtain $\{r_{T_i} \| r_{R_j}\}_{k_{ij}}$. Then, T_i concatenates preID_{T_i}, A_s and $\{r_{T_i} \| r_{R_j}\}_{k_{ij}}$ to encrypt by k_u.

   T_i sends the message $\left\{\text{preID}_{T_i} \| A_s \| \left\{r_{T_i} \| r_{R_j}\right\}_{k_{ij}}\right\}_{k_u}$ to R_j as an response.

3. *Phase 3:* When R_j receives the response, it extracts preID_{T_i} and A_s from $\left\{\text{preID}_{T_i} \| A_s \| \left\{r_{T_i} \| r_{R_j}\right\}_{k_{ij}}\right\}_{k_u}$ with k_u. Then, R_j forwards $\text{preID}_{T_i} || A_s$ to DB for verifying the identity of T_i and obtains k_{ij}.
4. *Phase 4:* When receiving $\text{preID}_{T_i} || A_s$, DB first extracts preID_{T_i} to verify whether the message was generated from a legal tag of the corresponding A_s. If preID_{T_i} is correct, DB will deliver the corresponding authentication key k_{ij} to R_j from the key array D_{st}.

When R_j receives k_{ij}, it will check whether the current computed r_{R_j} equals the previous generated r_{R_j} in Phase 1. If the two values are identical, T_i will be successfully authenticated by R_j. Otherwise, it will stop the authentication process with an error code.

Phase 5: Reversing r_{T_i}, we obtain $\overline{r_{T_i}}$. R_j encrypts $\overline{r_{T_i}}$ by k_{ij} and forwards the ciphertext $\left\{\overline{r_{T_i}}\right\}_{k_{ij}}$ to T_i. To avoid being intercepted directly by attackers from the latter part of $\{r_{T_i} \| r_{R_j}\}_{k_{ij}}$ which will be used for spoofing, it chooses $\overline{r_{T_i}}$ to transmit.

When receiving $\{\overline{r_{T_i}}\}_{k_{ij}}$, T_i extracts r_{T_i} by decrypting and reversing. T_i checks whether the current computed r_{T_i} equals the previous generated r_{T_i} in Phase 2. If the two values are identical, R_j will be successfully authenticated by T_i. Otherwise, it will stop the authentication process with an error code.

The authentication process is illustrated in Fig. 1, and the phases can be described as follows

$$P1(R_j \to T_i):\quad \left\{r_{R_j} \| \text{preID}_{R_j} \| G_t \| g_j\right\}_{k_u}$$

$$P2(T_i \to R_j):\quad \left\{\text{preID}_{T_i} \| A_s \| \left\{r_{T_i} \| r_{R_j}\right\}_{k_{ij}}\right\}_{k_u}$$

$$P3(R_j \to \text{DB}):\quad \text{preID}_{T_i} \| A_s$$

$$P4(\text{DB} \to R_j):\quad k_{ij},\ \left(r_{R_j}\right)_{\text{com}} \triangleq \left(r_{R_j}\right)_{\text{ori}}$$

$$P5(R_j \to T_i):\quad \left\{\overline{r_{T_i}}\right\}_{k_{ij}},\ \left(r_{T_i}\right)_{\text{com}} \triangleq \left(r_{T_i}\right)_{\text{ori}}$$

2.3 Updating for pseudorandom identifier

Using the same preID to replace a same tag ID or a same reader ID for long time is unsafe for RFID systems. Tags and readers can easily suffer from replay, tracking and forgery attacks. We will update preID for every tag and reader when they authenticate with each other. To maintain consistency, DB will store the new preID and the old preID for a tag or a reader simultaneously, which can prevent the DoS attack from not updating the preID simultaneously between a tag (reader) and DB. If an error occurred during updating the preID to a tag or a reader, DB will use the old preID to attend the authentication process.

The generation of the new preID of a tag is described by (2), and the generation of the new preID of a reader is described by the following equation

$$\text{preID}_{T_i(\text{new})} = \text{PRNG}\left(\text{preID}_{T_i}\right) \tag{2}$$

$$\text{preID}_{R_j(\text{new})} = \text{PRNG}\left(\text{preID}_{R_j}\right) \tag{3}$$

PRNG is a unidirectional random number generator, which is irreversible. DB stores $\text{preID}_{T_i(\text{new})}$, preID_{T_i}, $\text{preID}_{R_j(\text{new})}$ and preID_{R_j} simultaneously.

The new preID can be transmitted and verified during Phases 4 and 5. The authentication process of updating is described as follows

- *Phase 4:* DB will compute the $\text{preID}_{T_i(\text{new})}$ and $\text{preID}_{R_j(\text{new})}$. It will concatenate the corresponding authentication key k_{ij}, $\text{preID}_{T_i(\text{new})}$ and $\text{preID}_{R_j(\text{new})}$. Then, DB will deliver $\left\{\text{preID}_{T_i(\text{new})} \| \text{preID}_{R_j(\text{new})} \| k_{ij}\right\}$ to R_j. R_j extracts and checks $\text{preID}_{R_j(\text{new})}$ to decide whether the preID_{R_j} should be updated. If $\left(\text{preID}_{R_j(\text{new})}\right)_{\text{com}} \triangleq \text{PRNG}\left(\text{preID}_{R_j}\right)$, R_j will update its preID.
- *Phase 5:* R_j encrypts $\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})}$ by k_{ij} and forwards the ciphertext $\left\{\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})}\right\}_{k_{ij}}$ to T_i. When receiving $\left\{\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})}\right\}_{k_{ij}}$, T_i extracts $\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})}$ by decryption and checks the correctness of the r_{T_i} and its new predID. If $\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})} \triangleq \overline{\left(r_{T_i}\right)_{\text{ori}}} \oplus \text{PRNG}\left(\text{preID}_{T_i}\right)$, R_j will be successfully authenticated by T_i. Otherwise, it will stop the authentication process with an error code. T_i will then update its preID by formula (2). DB, readers and tags store the same PRNG function.

The authentication processes 4 and 5 are illustrated in Fig. 2, and the Phases 4 and 5 can be described as follows

$$P4(\text{DB} \to R_j):\quad \left\{\text{preID}_{T_i(\text{new})} \| \text{preID}_{R_j(new)} \| k_{ij}\right\}$$

$$\left(r_{R_j}\right)_{\text{com}} \triangleq \left(r_{R_j}\right)_{\text{ori}},\ \left(\text{preID}_{R_j(\text{new})}\right)_{\text{com}} \triangleq \text{PRNG}\left(\text{preID}_{R_j}\right)$$

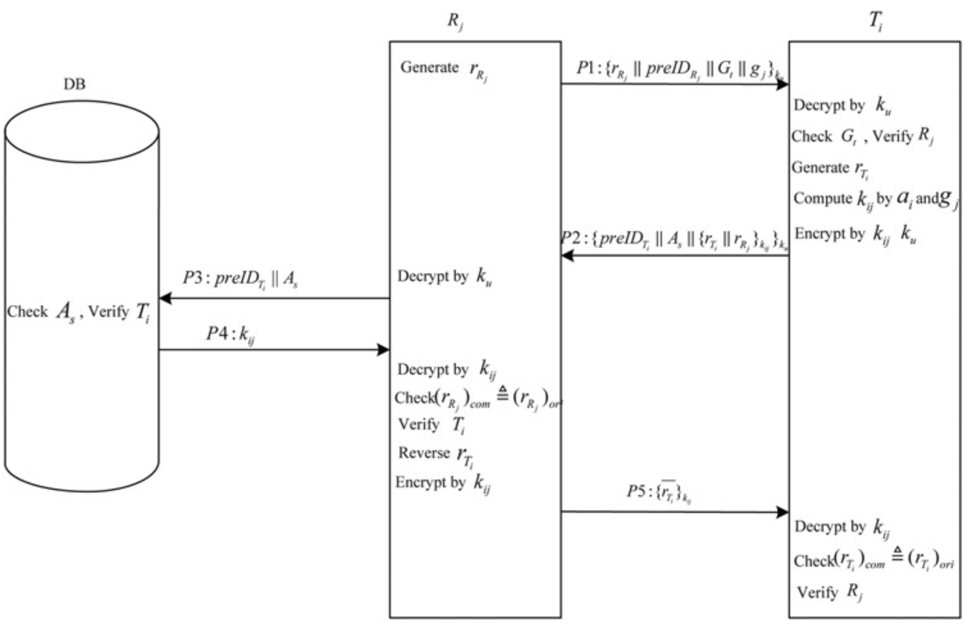

Fig. 1 *Authentication process of GKAAP*

doi: 10.1049/iet-ifs.2012.0241

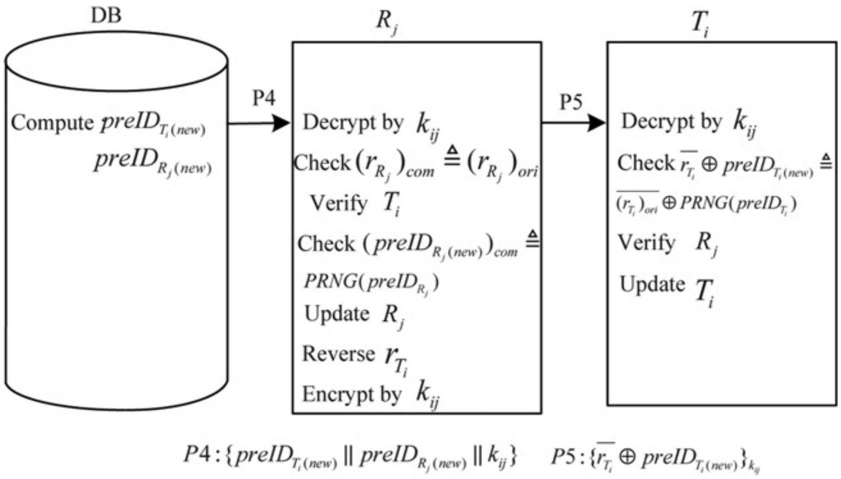

Fig. 2 *Authentication process adding pseudorandom identifier update*

$$P5(R_j \rightarrow T_i):\quad \left\{\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})}\right\}_{k_{ij}},$$

$$\overline{r_{T_i}} \oplus \text{preID}_{T_i(\text{new})} \triangleq \overline{\left(r_{T_i}\right)_{\text{ori}}} \oplus \text{PRNG}\left(\text{preID}_{T_i}\right)$$

3 Formal analysis of authentication protocol with GNY logic

In this section, GNY logic is applied to analyse the design correctness of GKAAP. We will analyse our protocol using four steps: proposing the initial assumptions, setting up an ideal model, confirming the security goals and verifying by GNY logic rules [12]. Table 1 shows notations to facilitate the formal descriptions.

3.1 Initial assumptions

We assume that the following holds at the beginning of every run of the protocol.

1. The assumptions for T_i

$$r_{T_i} \in T_i, \quad \text{preID}_{T_i} \in T_i, \quad A_s \in T_i, \quad T_i| \equiv \sharp\text{preID}_{R_j}$$

$$k_u \in T_i, \quad T_i| \equiv \sharp k_u, \quad k_{ij} \in T_i, \quad T_i| \equiv \sharp k_{ij}$$

$$T_i| \equiv T_i \overset{k_u,k_{ij}}{\leftrightarrow} R_j, \quad T_i| \equiv T_i \overset{k_{ij}}{\leftrightarrow} \text{DB}$$

Table 1 Symbol notations

Notation	Description
$P \triangleleft X$	P receives a message containing X, P can read and repeat X
$P \triangleleft *X$	P receives X, X is a not-originated-here formula
$X \in P$	P possesses, or is capable of possessing X
$P \mid\sim X$	P once conveyed X
$P \mid\equiv \# X$	P believes, or is entitled to believe that X is fresh
$P \mid\equiv \phi X$	P believes, or is entitled to believe that X is recognisable
$P \mid\equiv C$	P believes, or would be entitled to believe, that statement C holds
$P \mid\Rightarrow C$	P is an authority on statement C, and has jurisdiction over C
$P \mid\equiv P \overset{K}{\longleftrightarrow} Q$	P believes, or is entitled to believe, that K is a suitable secret for P and Q
$\{X\}_K$	symmetric encryption

These expressions indicate that: T_i possesses r_{T_i}, preID_{T_i}, A_s, k_u and k_{ij}; T_i believes that k_u and k_{ij} are fresh and T_i is entitled to believe that preID_{R_j} is fresh; T_i believes k_u and k_{ij} are suitable secrets for T_i and R_j; T_i believes k_{ij} is a suitable secret for T_i and DB.

2. The assumptions for R_j

$$r_{R_j} \in R_j, \quad \text{preID}_{R_j} \in R_j, \quad G_t \in R_j, \quad g_j \in R_j,$$

$$R_j| \equiv \sharp\text{preID}_{T_i}$$

$$k_u \in R_j, \quad R_j| \equiv \sharp k_u, \quad R_j| \equiv R_j \overset{k_u,k_{ij}}{\leftrightarrow} T_i$$

These expressions indicate that: R_j possesses r_{R_j}, preID_{R_j}, G_t, g_j and k_u; R_j believes that k_u is fresh and R_j is entitled to believe that preID_{T_i} is fresh; R_j believes that k_u and k_{ij} are suitable secrets for T_i and R_j.

3. The assumptions for DB

$$\text{DB}| \equiv \sharp\text{preID}_{T_i}, \quad \text{DB}| \equiv \sharp A_s$$

$$k_{ij} \in \text{DB}, \quad \text{DB}| \equiv \sharp k_{ij}, \quad \text{DB}| \equiv \text{DB} \overset{k_{ij}}{\leftrightarrow} T_i$$

These expressions indicate that: DB possesses k_{ij}; DB believes that A_s and k_{ij} are fresh, and DB is entitled to believe that preID_{T_i} is fresh; DB believes that k_{ij} is a suitable secret for DB and T_i.

3.2 Ideal model

According to the authentication phases, the formal messages delivered between T_i R_j, and DB can be expressed as follows

(1) $T_i \triangleleft: *\{r_{R_j}\}_{k_u}, *\{\text{preID}_{R_j}\}_{k_u}, *\{G_t\}_{k_u}, *\{g_j\}_{k_u}$

(2) $R_j \triangleleft: *\{\text{preID}_{T_i}\}_{k_u}, *\{A_s\}_{k_u}, *\{r_{T_i}\}_{k_u,k_{ij}}, *\{r_{R_j}\}_{k_u,k_{ij}}$

(3) $\text{DB} \triangleleft: *\text{preID}_{T_i}, *A_s$

(4) $R_j \triangleleft: *k_{ij}, *\text{preID}_{T_i(\text{new})}, *\text{preID}_{R_j(\text{new})}$

(5) $T_i \triangleleft: *\left\{\overline{r_{T_i}}\right\}_{k_{ij}}, *\left\{\text{preID}_{T_i(\text{new})}\right\}_{k_{ij}}$

3.3 Security goals

The security goals for T_i, R_j and DB can be expressed as follows

(1) $T_i|\equiv R_j|\sim r_{R_j}$, $T_i|\equiv R_j|\sim \text{preID}_{R_j}$, $T_i|\equiv R_j|\sim G_t$, $T_i|\equiv R_j|\sim g_j$; $T_i|\equiv \sharp\left\{\text{preID}_{R_j}\right\}_{k_u}$

R_j conveyed r_{R_j}, preID_{R_j}, G_t and g_j; T_i believes that R_j conveyed r_{R_j}, $\{\text{preID}_{R_j}\}_{k_u}$ is fresh

(2) $R_j\Big|\equiv T_i\Big|\sim r_{T_i}$, $R_j\Big|\equiv T_i\Big|\sim \text{preID}_{T_i}$, $R_j\Big|\equiv T_i|\sim A_s$ $R_j\Big|\equiv DB\overset{k_{ij}}{\leftrightarrow}T_i$, $R_j\Big|\equiv \sharp\left\{\text{preID}_{T_i}\right\}_{k_u}$

R_j believes that T_i conveyed r_{T_i}, preID_{T_i}, A_s; R_j believes the authentication key k_{ij} between DB and T_i; R_j believes that $\{\text{preID}_{T_i}\}_{k_u}$ is fresh

(3) $\text{DB}\big|\equiv T_i\big|\sim \text{preID}_{T_i}$

DB believes that T_i conveyed preID_{T_i}.

3.4 Verification

The security goals can be verified by the initial assumptions and GNY logic rules in the ideal model. The process is similar to the protocol [10].

1. *For C(1): $T_i|\equiv R_j|\sim r_{R_j}$*

From $B(1)$, it has $T_i\lhd *\{r_{R_j}\}_{k_u}$. Applying rule $T1$: $(P\lhd(*X))/(P\lhd X)$, we obtain $T_i\lhd\{r_{R_j}\}_{k_u}$. From the assumptions for T_i, it has $k_u\in T_i$. Applying rule $T3$: $(P\lhd\{X\}_K, K\in P)/(P\lhd X)$, we obtain $T_i\lhd r_{R_j}$. Then, applying rule $P1$: $(P\lhd X)/(X\in P)$, we obtain $r_{R_j}\in T_i$. Applying rule $P3$: $(X\in P)/(H(X)\in P)$ and $R6$: $(H(X)\in P)/(P|\equiv\phi(X))$, we obtain $T_i|\equiv\phi(r_{R_j})$. Thus, T_i is entitled to believe that r_{R_j} is recognisable. Applying the assumption $T_i|\equiv\sharp k_u$ and rule $F1$: $(P|\equiv(X))/(P|\equiv\sharp(X,Y), P|\equiv\sharp F(X))$, we obtain $T_i|\equiv\sharp(r_{R_j},k_u)$. Applying rule $I1$

$$\frac{P\lhd *\{X\}_K, K\in P, P|\equiv P\overset{K}{\leftrightarrow}Q, P|\equiv\phi(X), P|\equiv\sharp(X,K)}{P|\equiv Q|\sim X, P|\equiv Q|\sim\{X\}_K, P|\equiv K\in Q}$$

yields $T_i|\equiv R_j|\sim r_{R_j}$. As a consequence, T_i believes that R_j once conveyed r_{R_j}.

2. For *C(1): $T_i|\equiv\sharp\left\{\text{preID}_{R_j}\right\}_{k_u}$*

From the assumptions for T_i, it has $T_i|\equiv\sharp\text{preID}_{R_j}, k_u\in T_i$. Applying rule $F2$: $(P|\equiv\sharp(X), K\in P)/\left(P|\equiv\sharp\{X\}_K, P|\equiv\sharp\{X\}_{K^{-1}}\right)$, we obtain $T_i|\equiv\sharp\left\{\text{preID}_{R_j}\right\}_{k_u}$. As a consequence, T_i believes that $\left\{\text{preID}_{R_j}\right\}_{k_u}$ is fresh.

3. *For C(2): $R_j|\equiv DB\overset{k_{ij}}{\leftrightarrow}T_i$*

From the assumptions for DB, it has $\text{DB}|\equiv DB\overset{k_{ij}}{\leftrightarrow}T_i$ and the communication channel between R_j and DB is secure, so $R_j|\equiv\text{DB}$. Applying rule J1:$(P|\equiv Q|\Rightarrow C, P|\equiv Q|\equiv C)/(P|\equiv C)$, we obtain $R_j|\equiv DB\overset{k_{ij}}{\leftrightarrow}T_i$. As a consequence, R_j believes that k_{ij} used in DB and T_i is credible.

The remainder of the security goals can be verified based on the above corresponding processes. GKKAP protocol uses a key array based on group to set up authentication key, which is different from [9, 10]. It also has a different authentication process and an update method, but the formal analysis results are similar to the protocol [10]. Thus, our protocol has the correctness and security performance based on this formal analysis.

4 Evaluation

4.1 Security analysis

The presented authentication approach offers a high resistance to most common attacks relating to RFID systems. We analyse its security as follows:

1. *Forgery attack resistance:* The forgery attack is an attacker disguised as a reader or a tag which tries to obtain valid responses to cheat the legal entities. Our protocol can resist the forgery attack. The solution can be discussed by two cases.
   - *Forge T_i:* The attacker does not have the right a_i and preID_{T_i}, so it cannot compute a correct k_{ij}, thereby failing to be authenticated by DB.
   - *Forge R_j:* The attacker does not have a correct G_t that belongs to the permissible group list of T_i, so it cannot pass the authentication of T_i. The incorrect g_j cannot compute a correct k_{ij}, thereby failing to be authenticated by DB.
2. *Replay attack resistance:* Replay attack means that an attacker impersonates a reader or a tag by replaying the previous query or response, in order to complete the authentication process. The random number generated by readers and tags can resolve the problem, because the random number is different in every transmission. We will compare the received and original random numbers to see whether they are the same or not. If they are same, the authentication is successful. In Phase 4, the reader verifies the random number directly, and in Phase 5, the reader chooses $\overline{r_{T_i}}$ to transmit, which can protect against replay attack by intercepting directly from the latter part of $\{r_{T_i}\|r_{R_j}\}_{k_{ij}}$. The tag must reverse the received random number to verify.
3. *Tracking attack resistance:* Multiple malicious readers in fixed locations transmit the same query to a tag. If the tags response remains invariant in all transmissions, the reader may track the tag passing by. The response of a tag to the same query must be changed. So the following measures can be employed to solve the above problem.
   - There are variable random numbers at different time.
   - The true ID of tags can be substituted by preID, and the preID must be updated after every authentication process.
4. *DoS attack resistance:* Owing to the blocking and proofing attack, the updating message cannot reach the tag, so the updating of preID is prevented. The following authentication cannot be carried out smoothly. DB stores the new preID and the old preID simultaneously. If the tag does not receive the updating message, it will use the old one which can still pass the verification of DB.

IET Inf. Secur., 2014, Vol. 8, Iss. 6, pp. 290–296
doi: 10.1049/iet-ifs.2012.0241

Table 2 Security features of various protocols

	Hash chain	HIDVP	Karthikeyan *et al.*	Yang *et al.*	Chien *et al.*	KAAP	Our protocol
mutual authentication	N	Y	Y	Y	Y	Y	Y
forgery attack resistance	N	Y	N	Y	Y	Y	Y
replay attack resistance	N	Y	N	Y	N	Y	Y
tracking attack resistance	Y	Y	N	Y	Y	Y	Y
DoS attack resistance	N	N	N	N	N	Y	Y
internal attack resistance	N	N	N	N	N	△	Y

N, the protocol does not have ability to resist the attacks; *Y*, the protocol has ability to resist the attacks; Δ, the protocol has partially ability to resist the attacks

5. *Internal attack resistance:* The protocol from [10] can prevent internal attack from different groups of readers or tags. Readers and tags of the internal system impersonate another reader and tag from other groups, which can be discovered by DB quickly, because different groups use different k_{ij}. This method can resolve internal attack between groups, but it cannot solve the internal attack in groups, which can be solved by having a different k_{ij} for each reader and tag in the same group. Each tag and reader has a different authentication key in our protocol, so our protocol can resist internal attack, not only among the different groups but also within the same group.

6. *Mutual authentication:* Our protocol provides mutual authentication between the communicating entities. The preID_{R_j} of reader and r_{T_i} are authenticated by the tag in Phases 1 and 5. The preID_{T_i} of tag, r_{R_j} and $k_{i,j}$ are authenticated by the reader with the aid of DB in Phases 3 and 4. The shared key k_u is mainly used to provide the confidential authentication to protect against the external attacks.

Table 2 summarises and compares the security analysis of various recent protocols as well as our novel protocol presented in this paper. It can be concluded that the proposed protocol provides a much higher level of security than the previous presented protocols. The compared protocols are typical and interrelated with GKAAP. It can resist both the external and internal attacks effectively.

4.2 Performance analysis

During the performance analysis, the length of keys and random numbers are ignored for the sake of simplicity. In our protocol, the access list of tag includes the reader group ID, the number of which is T. The tag does not store the authentication key, but only a_i, the number of which is also T. Then, the tag must store its identifier, the length of which is L. The storage of a tag is $2T + L$. The value T is lower than L which is also defined as the length of the access list in KAAP. The communication load and computation load is similar to the protocol presented in paper [10], but the delivery message increase L to update from a reader to a tag. A reader connects with DB directly, so its ability to transmit additional data are very strong. Adding the updating part, tags and readers must carry out random number operation (PRNG) to update preID in our protocol, so the computation load is $2(R + E)$. We compare performance between GKAAP and KAAP because KAAP [10] is most closely related to our protocol. Table 3 shows the performance comparison between KAAP and our protocol. From it, we can see that our protocol has lower storage requirements, and slightly higher communication load and computation load. PRNG has less cost, so computation load is acceptable. Hence, GKAAP can be used in the applications requiring high security with proper cost. Thus, the generation method of authentication key by key array in our protocol did not increase tag consumption, but unexpectedly reduced the consumption.

Table 3 Performance comparison between related protocols

	Storage	Communication load		Computaion load	
		$T \to R$	$R \to T$	T	R
KAAP	$3L$	L	L	$R + 2E$	$R + 2E$
our protocol	$2t + L$	L	$2L$	$2(R + E)$	$2(R + E)$

L, length of identifier/access list; *R*, random number generation (RNG) operation; *E*, encryption-decryption operation

5 Conclusions

We describe a novel security authentication protocol GAKKP, which is suitable for large scale RFID systems. To strengthen the resistance to internal attack in groups, we propose a method to generate authentication key by key array, which can reduce storage of tags effectively. During the course of identity authentication, we introduce the identifier updating phase, which can enhance the performance to prevent external attacks. From GNY analysis, the design correctness of our protocol is proved. The security and performance analysis results show that the protocol we present can achieve better security than previous protocols with proper cost. Based on the comprehensive analysis and comparison, our protocol can be used in applications requiring high security.

6 Acknowledgments

The authors acknowledge support from the Nature Science Foundation of China (nos. 61301092, 61202394) and the Nature Science Foundation of Shanxi Province of China (no. 2012JQ8005). The authors would also like to thank all of the anonymous reviewers for their comments and suggestions.

7 References

1 Ohkubo, M., Suzuki, K., Kinoshita, S.: 'Hash-chain based forward secure privacy protection scheme for low-cost RFID'. Proc. 2004 Symp. on Cryptography and Information Security (SCIS 2004), Sendai, 2004, pp. 719–724

2 Rhee, K., Kwak, J., Kim, S.: 'Challenge-response based RFID authentication protocol for distributed database environment'. Proc.

Second Int. Conf. on Security in Pervasive Computing (SPC 2005), (*LNCS*, **3450**), Berlin, 2005, pp. 70–84
3 Moessner, M., Gul, N.K.: 'Secure authentication scheme for passive C1G2 RFID tags', *Comput. Netw.*, 2012, **56**, (1), pp. 273–286
4 Zhou, S., Zhang, Z., Luo, Z. *et al.*: 'A lightweight anti-desynchronizetion RFID authentication protocol', *Inf. Syst. Front*, 2010, **12**, pp. 521–528
5 Karthikeyan, S., Nesterenko, M.: 'RFID security without extensive cryptography'. Proc. Third ACM Workshop on Security of AdHoc and Sensor Networks, 2005, pp. 63–67
6 Yang, J., Ren, K., Kim, K.: 'Security and privacy on authentication protocol for low-cost radio'. Proc. 2005 Symp. on Cryptography and Information Security, 2005
7 Chien, H., Chen, C.: 'Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards', *Comput. Stand. Interfaces*, 2007, **29**, (2), pp. 254–259
8 Kolias, C., Kolias, V., Kambourakis, G.: 'A secure and efficient authentication protocol for passive RFID tags'. Proc. Sixth Int. Symp. on Wireless Communication Systems (ISWCS), 2009, pp. 36–40
9 Ding, Z., Guo, L., Wang, Y.: 'An authentication protocol based on key array for RFID', *Electron. Inf. Technol.*, 2009, **31**, (3), pp. 722–726
10 Ning, H., Liu, H., Mao, J., Zhang, Y.: 'Scalable and distributed key array authentication protocol in radio frequency identification-based sensor systems', *IET Commun.*, 2011, **5**, (12), pp. 1755–1768
11 Blom, R.: 'An optimal class of symmetric key generation system'. Proc EUROCRYPT84, 1984, pp. 335–338
12 Gong, L., Needham, R., Yahalom, R.: 'Reasoning about belief in cryptographic protocols'. Proc. IEEE Computer Society Symp. Research in Security and Privacy, California, USA, May 1990, pp. 234–248

doi: 10.1049/iet-ifs.2012.0241