Ultralightweight RFID Authentication Protocol Based on Random Partitions of Pseudorandom Identifier and Pre-shared Secret Value*

NING Huansheng¹, LIU Hong¹ and YANG Chen²

(1.School of Electronic and Information Engineering, Beihang University, Beijing 100191, China)
(2.China Electronics Standardization Institute, Beijing 100007, China)

Abstract — As open and wireless RFID air interfaces suffer from severe threats, concerns with respect to the security and privacy problems are increasingly becoming noteworthy issues. The paper proposes an ultralightweight RFID authentication protocol based on Random partition (RPAP) to achieve security and efficiency. The protocol adopts multiple mechanisms (i.e. random partition, dynamic update, and mutual authentication) as safeguards in the air interface. Specifically, random partition mechanism is used to divide the pseudorandom identifier and pre-shared secret value for bitwise operations; dynamic update mechanism enhances the freshness during identifications; double-entity-round mutual authentication mechanism provides stratified access control. Meanwhile, attack models are established to analyze the resistance to typical attacks (i.e. replay, spoofing, tracking, and tampering) in the primary and further authentication. Furthermore, security and performance are analyzed to prove that the protocol owns high security, high efficiency, and low complexity. The protocol is practical for the low-cost and resource-limited RFID applications.

Key words — Radio frequency identification (RFID), Authentication, Protocols, Security.

I. Introduction

Radio frequency identification (RFID) as a sensor technology uses the electromagnetic wave for object identification. Due to the open wireless communication environments, the RFID system, particularly the reader-tag air interface, is suffering from several security threats[1]. While the backward link between readers and the back-end database should be well safeguarded, and the forward air interface between readers and tags even needs more considerations. It is necessary to propose an effective scheme to improve robustness, reliability and security to resist major active and passive attacks. Therefore, security and privacy have become critical issues.

In order to guard against the unauthorized access to sensitive tag data, several security schemes have been proposed to address potential security problems[2–9], which apply bitwise Boolean operator, hash function, Cyclic redundancy code (CRC) function, Pseudo random number generator (PRNG), and also full-fledged cryptographic primitives for security and privacy protection. However, some complicated protocols may be limited by the tag hardware requirements such as power consumption, storage space, computational capacity etc. Whereas considering the cost, most RFID applications adopt passive tags with lower operational capabilities. Hence, it is reasonable to design an ultralightweight RFID authentication protocol required less operations and fewer exchanged messages to achieve acceptable security and efficiency.

In this paper, we focus on low-cost RFID systems. An ultralightweight Authentication protocol based on Random partition identifier (RPAP) is proposed for low-cost RFID applications. The scheme could resist major potential active and passive attacks. Main contributions in this paper are as follows. Firstly, random partition mechanism is adopted in two aspects. One is dividing the pseudorandom identifiers to extract the first certain bits for quick search and the primary authentication. The other is dividing the pre-shared secret value into three partial fields for further authentication. The partition is self-refreshed in each session to avoid additional update modules and workloads. Secondly, dynamic update mechanism is applied to produce random numbers and index-pseudonyms, which are introduced into the bitwise logical operations to refresh the values dynamically. It avoids desynchronization among all the tags, readers and database efficiently. Finally, double-entity-round mutual authentication mechanism is used between the reader and tag to ensure high security. The proposed mutual authentication mode has merits of providing stratified access control and improving efficiency.

The paper is organized as follows. In Section II, related works on lightweight RFID security protocols are reviewed. The progress of the proposed protocol is introduced in Section III. Then in Section IV, attack models in primary and further authentications are built to analyze resistance to the typical attacks. In Section V and Section VI, the implementation of security and performance are analyzed respectively. Finally, a conclusion is drawn in Section VII.

II. Related Works

In the section, related works on the lightweight RFID authentication protocols are presented. A series of ultralightweight schemes with bitwise operators and other simple functions have been proposed for low-cost RFID systems.

Peris-Lopez et al. proposed LMAP in Ref.[2], the protocol is efficient and requires less logic gates, which uses index-pseudonyms and bitwise operation to realize tag anonymity and data integrity.

*Manuscript Received Dec. 2010; Accepted Feb. 2011. This work is supported by the National High Technology Research and Development Program of China (863 Program) (No.2008AA04A101).
However, it has been proved vulnerable to the de-synchronization and full-disclosure attacks, in which an adversary could interfere with both entities during the authentication rounds to disclose all sensitive secret tag identifiers.

Chien's SASI in Ref.[4] is a typical ultra-lightweight protocol in which public sub-messages are built via bitwise operations. Exclusive-or (XOR) operation is the main functional component that is needed, and pseudonym is pre-shared as the search index to determine a matched record in the database. The possible de-synchronization attack can be resisted due to the freshness and dynamic update mechanism which are applied for storing the old key and the potential key to resist the de-synchronous attack. Meanwhile, the use of addition mod 2^n is realistic for low-cost and low-power applications. Furthermore, Phan[13] and Cao et al.[12] pointed out that SASI with limited integrity protection does not satisfy the desired objective of untraceability, and the protocol does not achieve the resistance to DoS and tracking attacks.

Hopper and Blum's HB protocol in Ref.[5] is suitable for pervasive computing environments since it only requires scalar dot product operation of binary vectors. The security of this scheme relies on the hardness of the computational Learning parity with noise (LPN) problem. While the HB protocol focus on the major passive attacks, a series of modified variants HB+, HB++[13,14] have been improvised to resist active attacks, along with preserving HB's advantages of low requirements for tag resources to be implemented. The family of protocols is proved to be secure against active attacks, and can be implemented with so few resources on an RFID tag.

Zhou et al. proposed a lightweight anti-desynchronization preserving authentication protocol in Ref.[6], which is suitable for pervasive computing environments since only the capacity of hash function and XOR operation are required. In the protocol, the backend database keeps the former records of the random key update to prevent the active attackers who forge prevailing tags.

Chien and Chen proposed a mutual authentication scheme in Ref.[7] on EPCglobal Class1 Gen2 tag, which uses CRC checksum code to detect error and verify the integrity of transmitted data. Meanwhile, access and kill commands are used to detect cloned tags, withstand the malicious eavesdropping readers, and a manufacturer can also implicitly keep track of tagged items. The updated authentication key and access key are adapted to enhance forward security. The random numbers are integrated to defend the tracing and replay attacks. However, due to the linear properties of the CRC function, the protocol is proved to be vulnerable for DoS attack[15]. Moreover, persistent certification by the brute search mode may increasingly burden the server and the overall performance will be reduced.

Kulseng et al.'s protocol in Ref.[9] is designed with mutual authentication to secure the ownership transfer of RFID tags. The minimalistic cryptography such as physically unclonable functions and linear feedback shift registers are used to realize the ownership transfer. The protocol is efficient in hardware and particularly suitable for the low-cost systems.

In summary, we propose an RFID authentication protocol absorbing the merits of former schemes based on lightweight bitwise operations. Compared with previous researches[2,4-7,9], the proposed RPAP based on random partition differs from the conventional security scheme applying complex hash function and cryptographic algorithms. Taking the limitations of tags into consideration, the proposed protocol based on bitwise operations is suitable for ubiquitous systems in pervasive computing environments. The main novelty of RPAP is applying the random partitioned the pseudorandom identifier and pre-shared secret value for dynamic update and mutual authentication. The combination of random partition mechanism, dynamic update mechanism, and mutual authentication mechanism has not received as much attention from previous researches. The general schemes use static secret values with additional update modules or provide one-side authentication where only one entity involved authenticates the other one.

III. Protocol Description

In the RFID systems, there are m readers, n tags and the backend database. The backward link between a reader and the database is assumed to be secure, but the forward link between a reader and a tag is suffering from various active and passive attacks. We suppose that each tag/reader owns its identifier (ID), and pre-shares a pseudonym (IDS) and a secret value (S) with the database. Here, the index-pseudonym storing a list of pseudorandom identifiers is used to retrieve a certain tag or reader. In our system structure, the readers and the database are integrated as a whole entity for protocol description. Furthermore, an attacker can not replicate a reader or a tag, which is a reasonable assumption since it is always possible to resistant tamper by hardware.

1. System parameters

The notations are introduced as follows.

- R_i: The i-th reader in the RFID systems, i = 1, 2, · · · , m.
- T_j: The j-th tag in the RFID systems, j = 1, 2, · · · , n.
- DB: The backend database.
- α: The attacker in the RFID systems.
- R_a, T_a: The reader and tag imitated by α.
- IDR_i, IDT_j: The identifier of R_i and T_j.
- IDSpa, IDSpb: The pseudorandom identifier of R_i and T_j.
- IDS_i′, IDS_j′: The imitated identifier of R_a and T_a.
- S: The secret value pre-shared by each legal entity.
- d: The random integer used for random partition, d ∈ (0, 1/2).
- r_1, r_2: The random number.

\[ \lfloor \cdot \rceil: The first d-bit fields of a value. \]
\[ \Xi, U, +, - , Rot(\cdot): The bitwise logical operators that denote XOR, OR, addition mod 2^n, subtraction mod 2^n, and bit rotation. \]
\[ P: Concatenate operator. \]
\[ \rightarrow: Translation operator. \]

2. Authentication progress

The interactions between a particular reader R_i and a tag T_j are introduced to describe the protocol progress. Fig.1 shows the proposed RFID authentication protocol based on random partition, and the details are given as follows.

\[ \begin{align*}
R_i &\quad 1. d \quad \text{Generate } d \\
&\quad 2. [IDS_i′, S] \quad \text{Verify } T_j \quad \text{[IDS_i′ =IDS_j′?]} \\
&\quad 3. [IDS_j, S] \quad \text{Extract [IDS_j′= IDS_i′?]} \\
&\quad 4. D \quad \text{Verify } R_i \quad \text{[IDS_i′=IDS_j′?]} \\
&\quad \text{Compute: } S′=Rot((S⊕r_1), d) \quad S′=Rot((S⊕r_2), d) \\
&\quad D′=S′(d(1)S′)⊕S_i′ \quad D′=S_i′(d(1)S_i′)⊕S. \\
&\quad \text{Verify } T_j \quad \text{[IDS_i′=IDS_j′?]} \\
&\quad C′=C \quad \text{Verify } R_i \quad \text{[IDS_i′=IDS_j′?]} \\
&\quad \text{Compute: } S′=Rot((S⊕r_1), d) \quad S′=Rot((S⊕r_2), d) \\
&\quad D′=S′(d(1)S′)⊕S_i′. \end{align*} \]

Fig. 1. The RFID authentication protocol based on random partition

• Phase 1: Reader challenge and tag response

The reader R_i generates a random integer d, and sends it to the tag T_j as a query to initiate a new session. Upon receiving
the query, \( T_j \) extracts the first \( d \) bits of its pseudorandom identifier \( IDS_{T_j} \), to obtain \( |IDS_{T_j}|_d \), and replies it to \( R_i \) as a response.

- **Phase 2: Primary authentication by reader**
  When \( R_i \) receives the response, it re-extracts the first \( d \) bits of \( IDS_{T_j} \), to obtain \( |IDS_{T_j}|_d \), and checks whether it equals the received \( |IDS_{T_j}|_d \). If the two values are consistent, \( T_j \) will pass the primary authentication. Otherwise, \( R_i \) will regard \( T_j \) as illegal and terminate the protocol with an error code. Then, \( R_i \) extracts the first \( d \) bits of its pseudorandom identifier \( IDS_{R_i} \), to gain \( |IDS_{R_i}|_d \), and continues to divide the pre-shared secret value \( S \) into \( S = (S_1, PS, S_2) \) by \( d \). The partitioning method is as that, mark both the higher \( d \)-th bit and lower \( d \)-th bit of \( S \) as two delimiters which divide the whole \( S \) into three partial fields \( S_1, S_2, \) and \( S_3 \) for bitwise operations. During the random partitions, underflow should be considered, and zero is padded to the higher order bits. Meanwhile, \( R_i \) generates two random numbers \( r_1 \) and \( r_2 \), then proceeds with computing \( A, B, S, C \) via bitwise operations given in Fig.1, and sends the concatenation \( |IDS_{R_i}|_d PA PB PC \) to \( T_j \).

- **Phase 3: Primary and further authentication by tag**
  Upon receiving the message, \( T_j \) firstly extracts the first \( d \) bits of \( IDS_{R_i} \), to obtain \( |IDS_{R_i}|_d \) and verifies \( R_i \) by comparing whether the computed \( |IDS_{R_i}|_d \) is equal to the received \( |IDS_{R_i}|_d \). If the two values are consistent, \( R_i \) will pass the primary authentication. Otherwise, \( T_j \) will regard \( R_i \) as illegal and terminate the protocol with an error code. Then, \( T_j \) does the same random partition to obtain \( S_1, S_2, \) and \( S_3 \). Thereafter, \( T_j \) performs inverse operations to derive \( r_1 \) and \( r_2 \), then computes \( S'_i, S'_j, C' \) as the calculation equations in Fig.1. Afterward, \( T_j \) verifies \( R_i \) by comparing whether the computed \( C' \) equals the received \( C \). If the two values are consistent, \( R_i \) will pass the further authentication. Otherwise, \( T_j \) will regard \( R_i \) as illegal and terminate the protocol with an error code.

Then, \( R_i \) computes \( S'_i, S'_j, D \) via bitwise operations given in Fig.1, and sends \( D \) to \( T_j \).

- **Phase 4: Further authentication by reader**
  When \( R_i \) receives the message \( D \), it proceeds with the same bitwise operations to obtain \( S'_i, S'_j, D' \). Then, \( R_i \) verifies \( T_j \) by checking whether the computed \( D' \) equals the received \( D \). If the two values are consistent, \( T_j \) will pass the further authentication. Otherwise, \( R_i \) will regard \( T_j \) as illegal and the protocol will fail with an error code. Then, the protocol runs a whole round.

Alternatively, the authentication progress can be described in the notations. Therein the detailed calculation equations are specified in Fig.1.

\[
\begin{align*}
R_i & \rightarrow T_j : d \\
T_j & \rightarrow R_i : |IDS_{T_j}|_d \\
& \text{Primary authentication: } |IDS_{T_j}|_d == |IDS_{T_j}|_d \? \\
R_i & \rightarrow T_j : |IDS_{R_i}|_d PA PB PC \\
& \text{Primary authentication: } |IDS_{R_i}|_d == |IDS_{R_i}|_d \? \\
& \text{Further authentication: } C' == C' \? \\
T_j & \rightarrow R_i : D \\
& \text{Further authentication: } D' == D' \\
\end{align*}
\]

Above all, the ultralightweight authentication protocol adopts triple mechanisms (random partition mechanism, dynamic update mechanism, and mutual authentication mechanism) to realize security protection. The main approaches include:

1. **Random partition mechanism**: the random integer \( d \) is adopted for random partition mechanism in two aspects. One is dividing the entire pseudorandom identifiers \( IDS_{R_i}, IDS_{T_j} \) to extract the first \( d \) bit fields \( |IDS_{R_i}|_d, |IDS_{T_j}|_d \) for the primary authentication. The dynamic partial pseudonyms are applied for quick search instead of exhaustive search, and the pseudonyms are transmitted instead of the real identifiers \( IDS_{R_i}, IDS_{T_j} \) in the whole communication. The other is dividing the pre-shared secret value \( S \) into three partial fields \( S_1, S_2, S_3 \) randomly. The dynamic fields \( S_1, S_2, S_3 \) used for the further authentication are self-refreshed in each session, which can reduce additional update modules and workloads. The usage of above random partition mechanisms is an optimization done to improve efficiency, enhance security, and conserve memory.

2. **Dynamic update mechanism**: the random numbers \( (r_1, r_2) \) are generated as one-time pad encryption, which are used in the bitwise logical operations for dynamic update in each session. The mechanism avoids out of synchronization among tags, readers and the database. Meanwhile, pseudorandom identifiers also keep the dynamic update. Note the costly random number generation is carried out by readers that take a seed as an independent variable and output random numbers, along with simple bitwise operations are performed by tags.

3. **Mutual authentication mechanism**: the double-entity-round mutual authentication mode is applied to ensure access control. In the whole protocol round, both entities (readers and tags) perform the primary and further authentications to provide stratified mode. If and only if both authentications succeed, the tag will transmit its TID to the reader.

**IV. Attack Model Analysis**

In the section, the attack model is analyzed according to the primary and further authentications phases. For the primary authentication, we analyze the potential typical attacks (e.g. spoofing, replay, and tracking). For the further authentication, we use the attack model proposed by Cao in Ref.[12] to analyze the tampering attack. Cao’s model focuses on probable attack scenarios and analyzes the attack success probability, which is suitable for our authentication protocol. Based on the above two-phase authentications, the attack model is as follows: (1) suppose the attacker’s identity; (2) simulate how the attack is performed by an attacker by steps; (3) create compromised conditions and deduce the security.

We support that the links between readers and the database with higher performance are regarded as secure communications. And there is a completion message exchanged between the reader and the tag to indicate completion of the protocol.

**1. Attacks in primary authentication**

**1.1 Spoofing attack**

An attacker \( \alpha \) disguises as a legal reader or tag to obtain valid responses to cheat the legal entities. Under the spoiling attack, an attacker \( \alpha \) performs the following actions:

**Case 1**

\( R_i \rightarrow \alpha(T_a) : d \)
\( \alpha(T_a) \rightarrow R_i : |IDS_{T_a}|_d \)
\( R_i : |IDS_{R_i}|_d \neq |IDS_{T_a}|_d \).

**Case 2**

\( \alpha(R_a) \rightarrow T_j : d \)
\( T_j \rightarrow \alpha(R_a) : |IDS_{T_j}|_d \)
\( \alpha(R_a) \rightarrow T_j : |IDS_{R_a}|_d PA PB PC \)
\( T_j : |IDS_{R_a}|_d \neq |IDS_{R_a}|_d \).

**Case 1** In one session, \( \alpha \) disguises as a tag \( T_a \), and receives \( d \) from \( R_i \). When \( T_a \) receives the query, it will skip the primary authentication and extract the first \( d \) bits of \( IDS_{R_i} \) to obtain \( |IDS_{R_i}|_d \), then responds it to \( R_a \). Upon receiving the response, \( R_i \) will execute the primary authentication, and check whether there is a corresponding pseudorandom identifier \( IDS_{T_a} \) matching \( IDS_{R_i} \). The result will be that the computed \( |IDS_{T_a}|_d \) does not equal the received \( |IDS_{T_a}|_d \).

**Case 2** In one session, \( \alpha \) disguises as a reader \( R_a \), and sends \( d \) to \( T_j \). When \( T_j \) receives the query, it extracts the first \( d \) bits of \( IDS_{T_j} \) to gain \( |IDS_{T_j}|_d \), then responds it to \( R_a \). Upon receiving the response, \( R_a \) will skip the primary authentication on \( T_j \), and transmit the message \( |IDS_{R_a}|_d PA PB PC \) to \( T_j \) directly. Thereafter, \( T_j \) will execute the primary authentication, and there is no corresponding pseudorandom identifier \( IDS_{R_a} \) matching \( IDS_{R_a} \). The result will be that the computed \( |IDS_{R_a}|_d \) does not equal the received \( |IDS_{R_a}|_d \).
In RPAP, the attacker can not obtain the accurate the pseudorandom identifier, and there is no matching entry in the memory. Hence, the protocol can resist the spoofing attack.

(2) Replay attack

An attacker \( \alpha \) disguises as a legal reader or tag to involve into the communication to modify or skim the tag identifier. Under the replay attack, an attacker performs the following actions:

In one session, \( \alpha \) has learnt all the messages \( d \), \([IDS_{T_j}]_d\), and \([IDS_{R_a}]_d\)APBPC.

- **Case 1**

  \( R_i \rightarrow \alpha(T_a) \neq T_j : d' \)  
  \( \alpha(T_a) \rightarrow R_i : [IDS_{T_j}]_d \)  
  \( R_i : [IDS_{T_j}]_d' \neq [IDS_{T_j}]_d'. \) The primary authentication will fail.

- **Case 2**

  \( R_i \rightarrow T_j : d' \)  
  \( T_j \rightarrow \alpha(R_a) \neq R_i : [IDS_{T_j}]_d \)  
  \( \alpha(R_a) \rightarrow T_j : [IDS_{R_a}]_d\)APBPC  
  \( T_j : [IDS_{R_a}]_d' \neq [IDS_{R_a}]_d. \) The primary authentication will fail.

**Case 1** In the next session, \( \alpha \) disguises as a tag \( T_a \) to intercept the refreshed query \( d' \) from \( R_i \). When \( T_a \) receives the query, it will respond the former learnt message \([IDS_{T_j}]_d\) to \( R_i \). Upon receiving the response, \( R_i \) will execute the primary authentication, and the result will be that the computed \([IDS_{T_j}]_d'\) does not equal the received \([IDS_{T_j}]_d\).

**Case 2** In the next session, \( \alpha \) disguises as a reader \( R_a \) to intercept the response \([IDS_{T_j}]_d'\) from \( T_j \). When \( T_j \) receives the query, it will skip the primary authentication, and reply the former learnt message \([IDS_{T_j}]_d\)APBPC to \( T_j \). Upon receiving the response, \( T_j \) will execute the primary authentication, and the result will be that the computed \([IDS_{R_a}]_d'\) does not equal the received \([IDS_{R_a}]_d\).

In RPAP, the attacker may not pass the primary authentication with the dynamic update mechanism. Hence, the protocol can resist the replay attack.

(3) Tracking attack

An attacker \( \alpha \) disguises as multiple malicious readers \( R_{a_1}, R_{a_2}, \ldots, R_{a_x} \) in fixed locations to transmit the same query to \( T_j \). If the \( T_j \) responds the invariant messages in all transmissions, \( \alpha \) may launch the tracking attack. Under the tracking attack, an attacker \( \alpha \) performs the following actions:

- **Case 1**

  \( R_i \rightarrow \alpha(T_a) \neq T_j : d, d, \ldots, d \)  
  \( \alpha(T_a) \rightarrow R_i : [IDS_{T_j}]_d, [IDS_{T_j}^2]_d, \ldots, [IDS_{T_j}^x]_d \)  
  \( \alpha(R_a, R_{a_2}, \ldots, R_{a_x}) \rightarrow T_j : d, d, \ldots, d \) are pseudorandom. The primary authentication will fail.

  **(1)** \( \alpha \) disguises as different readers \( R_{a_1}, R_{a_2}, \ldots, R_{a_x} \) to capture messages from \( T_j \), then \( \alpha \) continuously queries \( T_j \) with the same query \( d \) which may yield consistent responses.

  **(2)** When \( T_j \) receives the queries sequentially, \( T_j \) will respond readers \( R_{a_1}, R_{a_2}, \ldots, R_{a_x} \) with a series of partial pseudorandom identifiers \([IDS_{T_j}]_d, [IDS_{T_j}^2]_d, \ldots, [IDS_{T_j}^x]_d\). For instance, \( T_j \) responds with \([IDS_{T_j}]_d\) in one site, and responds with \([IDS_{T_j}^2]_d\) in another site, and so forth. Any two responses are independent since the pseudorandom identifiers \([IDS_{T_j}]_d, [IDS_{T_j}^2]_d, \ldots, [IDS_{T_j}^x]_d\) are randomly generated. Thus, the protocol can prevent tracking special tag by the random mechanism.

In RPAP, the attacker can not recognize which tag responds the messages since the random integer \( d \) is used for update in each session. Therefore, the attacker is impossible to launch the tracking attack so that the location privacy is guarded.

2. Attacks in further authentication

(1) Tampering attack: Changing A, C, and D

An attacker \( \alpha \) can eavesdrop and record on the ongoing protocol, then it changes \([IDS_{R_a}]_d\)APBPC and \( D \) to \([IDS_{R_a}]_d\)AP\( ^{\prime} \)BPC and \( D' \), in which \( \hat{A} = A \oplus [I]_x, \hat{C} = C \oplus [I]_x, \hat{D} = D \oplus [I]_x, \) and \([I]_x = [00\ldots 00] \) (set the \( x \)-th bit of \( I \) as 1, the other bits as 0). The tampering attack progress is given as follows:

\[ R_i \rightarrow \alpha(T_a) : [IDS_{R_a}]_d\)APBPC \]
\[ \alpha(R_a) \rightarrow T_j : [IDS_{T_j}]_d\)AP\( ^{\prime} \)BPC \]
\[ T_j : C' \neq C. \] The further authentication will fail.

\[ T_j \rightarrow \alpha(R_a) : D \]
\[ \alpha(T_a) \rightarrow R_i : D \]
\( R_i : D' \neq D. \) The further authentication will fail.

In one session, the attacker \( \alpha \) disguises as an imitated tag \( T_a \) to intercept the message \([IDS_{R_a}]_d\)APBPC from \( R_i \) to \( T_j \), then \( \alpha \) disguises as an imitated reader \( R_a \) to send \([IDS_{R_a}]_d\)APBPC to \( T_j \). The validity of the message \([IDS_{R_a}]_d\)APBPC can be analyzed as follows.

\[ \hat{r}_1 = \hat{A} \oplus ([IDS_{T_j}]_dS_1) \]
\[ = (A \oplus [I]_x) \oplus ([IDS_{T_j}]_dS_1) \]
\[ = ([IDS_{T_j}]_dS_1 \oplus r_1 \oplus [I]_x) \oplus ([IDS_{T_j}]_dS_1) \]
\[ = r_1 \oplus [I]_x \] (1)

\[ \hat{r}_2 = \hat{B} \oplus [IDS_{T_j} - S_2 \oplus ([IDS_{T_j}]_d - S_2 \oplus r_2)] \]
\[ = ([IDS_{T_j}]_dS_2 \oplus r_2) \oplus ([IDS_{T_j}]_d - S_2 \oplus r_2) \]
\[ = r_2 \] (2)

\[ \hat{C} = \hat{C} \oplus [I]_x \]
\[ = ([S_3 \oplus r_1] \oplus ([S_3 \oplus r_2]) \oplus [I]_x \]
\[ = ([S_3 \oplus r_1] \oplus ([S_3 \oplus r_2] \oplus [I]_x) \]
\[ = S_3 \oplus r_1 \oplus [I]_x \] (3)

\[ C' = (S_3 \oplus r_1) \oplus ([S_3 \oplus r_2] \oplus r_2) \]
\[ = S_3 \oplus r_1 \oplus [I]_x \] (4)

Here, the values \( S_1, S_2, S_3 \) are not affected, and the computed \( C' \) does not equal \( \hat{C} \). \( T_j \) does not accept the imitated message \([IDS_{R_a}]_d\)APBPC.

In the worse condition, \( T_j \) accepts \([IDS_{R_a}]_d\)APBPC in a small probability. Attacker \( \alpha \) continues to disguise as an imitated reader \( R_a \) to forward \( D \) to \( R_i \),

\[ \hat{D} = \hat{D} \oplus [I]_x \]
\[ = ([S_3]U[S_2] \oplus S_3) \oplus [I]_x \]
\[ = (\text{Rot}((S_1 \oplus \hat{r}_1) \oplus d) \text{Rot}((S_2 \oplus r_2), d)) \oplus S_3 \oplus [I]_x \]
\[ = \text{Rot}((S_1 \oplus \hat{r}_1 \oplus [I]_x), d) \text{Rot}((S_2 \oplus r_2), d) \oplus S_3 \oplus [I]_x \]
\[ \hat{D}' = ([S_3]U[S_2] \oplus S_3) \]
\[ = (\text{Rot}((S_1 \oplus \hat{r}_1) \oplus d) \text{Rot}((S_2 \oplus r_2), d)) \oplus S_3 \]
\[ \neq \hat{D} \] (5)

Note that the scenario of changing \( B, C, \) and \( D \) is similar to Attack 1. Both \( T_j \) and \( R_i \) do not accept the imitated message \([IDS_{R_a}]_d\)AP\( ^{\prime} \)BPC and \( D' \), and the probability of accepting \([IDS_{R_a}]_d\)AP\( ^{\prime} \)BPC and \( \hat{D} \) is negligible.

(2) Tampering attack 2: Changing B and C

An attacker \( \alpha \) can eavesdrop and record on the ongoing protocol, then it changes \([IDS_{R_a}]_d\)APBPC to \([IDS_{R_a}]_d\)AP\( ^{\prime} \)BPC, in which \( \hat{B} = \hat{B} \oplus [I]_x, \hat{C} = \hat{C} \oplus [I]_x, \) and \([I]_x = [00\ldots 00] \) (set the \( x \)-th bit of \( I \) as 1, the other bits as 0). The tampering attack progress is given as follows:

\[ R_i \rightarrow \alpha(T_a) : [IDS_{R_a}]_d\)APBPC \]
α(Rα) → Tj : [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC
Tj : C' ≠ C. The further authentication will fail.

In one session, the attacker α disguises as an imitated tag Tα
To intercept the message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC from Rj to Tj, then α
disguises as an imitated reader Rα to send [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC to Tj.
The validity of the message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC can be analyzed as follows.

\[ r_1 = A \oplus (IDS_{gj}U_{S1}) \]
= ([IDS_{gj}U_{S1}]) \oplus r_1) \oplus (IDS_{gj}U_{S1})

\[ r_2 = \hat{B} \oplus IDS_{gj} - S_2 \]
= ([B \oplus [I_{x}]) \oplus IDS_{gj} - S_2)
= ([IDS_{gj} \oplus (S_2 + r_3)) \oplus [I_{x}] \oplus IDS_{gj} - S_2)
= r_2 \oplus IDS[I]_x

\[ \bar{C} = \bar{C} \oplus [I_{x}]_x \]
= S_3 \oplus r_1 \oplus [I_{x}] + (S_1 \oplus S_2)U_{r_1 \oplus [I_{x}] \oplus r_2} \oplus [I_{x}]_x

C' = (S_3 \oplus r_1) + (S_1 \oplus r_2)
= S_3 \oplus r_1 + (S_1 \oplus S_2)U_{r_1 \oplus [I_{x}] \oplus r_2}

\[ \# \bar{C} \]
Suppose that S_2 \oplus [I_{x}]_x equals S_2, we create such a worse condition which is an event of small probability. Here, the values (S_1, S_2, S_3) are not affected, and the computed C' still does not equal C. Tj does not accept the imitated message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC.

Note that the scenario of changing A and C is similar to Attack 2. Tj does not accept the imitated message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC, and the probability of accepting [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC is negligible.

(3) Tampering attack 3: Changing A and D

An attacker α can eavesdrop and record on the ongoing protocol, then it changes [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC and D to [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC, in which A = A \oplus [I_{x}]_x and \hat{B} = B \oplus [I_{x}]_x, and [I_{x}]_x = [000...00] (set the x-th bit of I as 1, the other bits as 0). The tampering attack progress is given as follows:

\[ R_1 \rightarrow \alpha(Tα) : [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC \]
\[ \alpha(Rα) \rightarrow Tj : [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC \]
\[ Tj : C' ≠ C. The further authentication will fail. \]
\[ \alpha(Tα) \rightarrow \alpha(Rα) : D \]
\[ \alpha(Tα) \rightarrow R_1 : \hat{D} \]
\[ R_1 : D' ≠ \hat{D}. The further authentication will fail. \]

In one session, the attacker α disguises as an imitated tag Tα
To intercept the message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC from Rj to Tj, then α
disguises as an imitated reader Rα to send [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC to Tj.
The validity of the message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC can be analyzed as follows.

\[ r_1 = r_1 \oplus [I_{x}]_x \]
\[ r_2 = r_2 \oplus [I_{x}]_x \]
\[ C = (S_3 \oplus r_1) \oplus (S_1 \oplus r_2) \]
= S_3 \oplus r_1 + (S_1 \oplus S_2)U_{r_1 \oplus [I_{x}] \oplus r_2}
\[ C' = (S_3 \oplus r_1) + (S_1 \oplus r_2) \]
= S_3 \oplus r_1 + (S_1 \oplus S_2)U_{r_1 \oplus [I_{x}] \oplus r_2}
\[ \# \bar{C} \]

\[ \# \bar{C} \]
Suppose that S_2 \oplus [I_{x}]_x equals S_2, we create such a worse condition which is an event of small probability. Here, the values (S_1, S_2, S_3) are not affected, and the computed C' still does not equal C. Tj does not accept the imitated message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC.

In the worse condition, Tj accepts [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC in a small probability. Attacker α continues to disguise as an imitated reader Rα to forward D to Rj.

\[ \hat{D} = D \oplus [I_{x}]_x \]
= ([S_{j}U_{S_{j}}] \oplus [S_{j}]_x) \oplus [I_{x}]_x
= ([R_{j}(S_1 \oplus r_1, d)U_{R_{j}}((S_2 \oplus r_2, d)] \oplus [I_{x}]_x
= R_{j}(S_1 \oplus r_1 + [I_{x}]_x, d)U_{R_{j}}((S_2 \oplus r_2, d) \oplus S_3 + [I_{x}]_x
\# \hat{D} \]

(4) Tampering attack 4: Changing A and B

An attacker α can eavesdrop and record on the ongoing protocol, then it changes [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC to [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC, in which A = A \oplus [I_{x}]_x and \hat{B} = B \oplus [I_{x}]_x, and [I_{x}]_x = [000...00] (set the x-th bit of I as 1, the other bits as 0). The tampering attack progress is given as follows:

\[ R_1 \rightarrow \alpha(Tα) : [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC \]
\[ \alpha(Rα) \rightarrow Tj : [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC \]
\[ Tj : C' ≠ C. The further authentication will fail. \]

In one session, the attacker α disguises as an imitated tag Tα
To intercept the message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC from Rj to Tj, then α
disguises as an imitated reader Rα to send [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC to Tj.
The validity of the message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC can be analyzed as follows.

\[ r_1 = r_1 \oplus [I_{x}]_x \]
\[ r_2 = r_2 \oplus [I_{x}]_x \]
\[ C = (S_3 \oplus r_1) + (S_1 \oplus r_2) \]
= S_3 \oplus r_1 + (S_1 \oplus S_2)U_{r_1 \oplus [I_{x}] \oplus r_2}
\[ C' = (S_3 \oplus r_1) + (S_1 \oplus r_2) \]
= S_3 \oplus r_1 + (S_1 \oplus S_2)U_{r_1 \oplus [I_{x}] \oplus r_2}
\[ \# \bar{C} \]

\[ \# \bar{C} \]
Suppose that S_2 \oplus [I_{x}]_x equals S_2, we create such a worse condition which is an event of small probability. Here, the values (S_1, S_2, S_3) are not affected, and the computed C' still does not equal C. Tj does not accept the imitated message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC.

In the worse condition, Tj accepts [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC in a small probability. Attacker α continues to disguise as an imitated reader Rα to forward D to Rj.

\[ \hat{D} = D \oplus [I_{x}]_x \]
= ([S_{j}U_{S_{j}}] \oplus [S_{j}]_x) \oplus [I_{x}]_x
= ([Rot((S_1 \oplus r_1, d)U_{Rot}((S_2 \oplus r_2, d)] \oplus [I_{x}]_x
= Rot((S_1 \oplus r_1 + [I_{x}]_x, d)U_{Rot}((S_2 \oplus r_2, d) \oplus S_3 + [I_{x}]_x
\# \hat{D} \]

Note that the scenario of changing B and D is similar to Attack 3. Neither Tj nor Rj accepts the imitated message [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC and D, and the probability of accepting [IDS_{Rj}]_\alpha P\bar{A}P\bar{B}PC and D is negligible.

V. Security Analysis

In RFID systems, the wireless links between the readers and tags are confronting more serious circumstances other than the relatively safe reader–database links. In the section, the security of RPAP is analyzed through the evaluation of main safety properties.

1. Data confidentiality

Confidentiality requires that all of the messages are securely transmitted during the wireless communication. In both forward
and backward links, the identifiers ($1D_{R1}, 1D_{ST}$) are substituted by the pseudorandom identifiers ($1D_{SR1}, 1D_{ST}$). The attackers cannot get any information from the intercepted messages due to the dynamic replaceable elements. Meanwhile, the calculations of $A, B, C, D$ involve at least two secret values, such as the dynamic partial fields ($S_1, S_2, S_3$) and the random numbers ($d, r_1, r_2$). For instance, the tag’s response $D = (D = (S_1 \oplus U_{S_2}) \oplus S_3)$, where $S_1' = \text{Rot}(S_1 \oplus r_1), d$ and $S_2' = \text{Rot}(S_2 \oplus r_2), d$ include an XOR operation with two random values, then operator $\text{Rot}(\cdot)$ left rotates the XORRed values with $d$ bits.

2. Data integrity
The pseudorandom identifiers ($1D_{SR1}, 1D_{ST}$) and the secret value ($S$) are stored in the rewritable tag memory, which makes it possible that data integrity may be destroyed by the malicious tampering. In RPAP, the pseudorandom identifier and pre-shared secret value are updated periodically, and only the legal readers and tags can calculate the values in accordance with the appointed rules. If an attacker succeeds to modify the exchanged data from a reader, then a tag would not deduce the inconsistent data, and should recognize the illegal attacker.

3. Authentication
The scheme provides mutual authentication between tags and readers by checking whether the computed values equal the previous received values according to the same algorithm. Double-entity-round authentication mode is executed to block an unauthorized access, including the primary and further authentications. Four pairs of values of ($1D_{SR1}, 1D_{SR2}$), ($1D_{ST}, 1D_{ST}$), ($C', C$), ($D', D$) are compared to achieve dual authentications. The tag and the reader can authenticate each other, since only the legal entities pre-share the secret random values $S$. For instance, the reader calculates the consistent APBTC for further authentication by $T_j$, then the tag derives the random numbers ($r_1, r_2$) and continues calculate $D$ for further authentication by $R_i$.

4. Anonymity
The protocol offers anonymity using pseudorandom identifiers instead of exposing the real identifiers so that the attacker can not identify the entire TID. Additionally, exchanged messages are random wraps because of random numbers ($d, r_1, r_2$) used to realize dynamic update and random access control. Even if the attacker intercepts and decodes the messages, it may only obtain the irregular pseudorandom values instead of the desired TID. The random integer $d$ is used to replace pseudonyms ($1D_{SR1}, 1D_{ST}$) with the first $d$ bit fields ($1D_{SR1}, 1D_{SR2}$) for delivery. Random number ($r_1, r_2$) are used to hide $1D_{ST}$, and the XORRed combination ($r_1 \oplus r_2$) is used to hide $S_1 \oplus S_2$ that belongs to the subelement of $S_3$. Hence, it seems irregular for reader-tag successive communication, and the attacker can not discern the pseudonyms and derive the tag identifier. Meanwhile, the forward security is guarded by such anonymous pseudonyms. Even if an attacker compromises and intercepts the messages in the former session, it still can not deduce the data in the current session since all the calculations are self-refreshed in each session.

5. Availability
Subversive denial of access (e.g. intercepting, blocking, and jamming messages) threatens availability in RFID systems. There is less denial of authorized access to communication in which the index-pseudonym as an access list to realize random access control. The reader sends the random query $d$ to awake the tag. Even though violent attacks occur, the scheme may provide recovery function to terminate the authentication with an error code enforced.

VI. Performance Analysis
The performance of RPAP is evaluated based on its ability to minimize tag storage, computation load and communication overhead.

The main tag storage requirement includes its static identifier $1D_{SR1}$, the updatable pseudorandom identifier $1D_{ST}$, the list of index-pseudonyms $1D_{SR2}$, the pre-shared secret value $S$, and other storage for bitwise logical operations. Specifically, the bitwise operations need much less storage than other cryptographic primitives such as hash function, cyclic redundancy code, and signature-then-encryption algorithms. For the computational cost, the tag involves only simple bitwise operations (i.e. XOR, OR, addition/subtraction mod 2$^d$, and bit rotation), which can be implemented with low cost and high efficiency in ubiquitous RFID systems. There are no additional hardware requirements, which may further reduce the database’s computation load and increase flexibility. The communication overhead also depends on terms of the number of messages exchanged during the protocol running round. The total authentication progress completes via the least four rounds in the normal condition. According to the EPCglobal Gen2 tag, 96-bit length is assumed for the secret values used in data deliveries, most of the overheads are contributed by the message ($\{1D_{SR1}\}$PAPBPC) of 384 bits length. Hence, the communication overhead is lightweight with high efficiency and reliability.

In summary, all the tag storage requirement, computation load and communication overhead are lightweight even ultralightweight. Hence, RPAP is ensured to be implemented without obvious vulnerabilities and is quite suitable for the resource-limited RFID applications to achieve highly cost effective requirements.

VII. Conclusion
In the paper, we propose an ultralightweight mutual authentication protocol. By applying the random partitions mechanism of the pseudorandom identifier and pre-shared secret value, integration and balance of security and performance issues are proved valid. The random partitions provide relative robust security with dynamic update mechanism and double-entity-round mutual authentication mechanism, which can withstand the typical attacks efficiently. The proper combination of triple mechanisms improves execution efficiency by index-pseudonyms and avoids additional modules for updates in each session. Moreover, lightweight bitwise operations are required to realize eximious functions, and it can be applied to low-cost and resource-limited RFID systems such as logistics and assets management.

References


NING Huansheng was born in 1975 in Anhui Province, China. He received B.S. degree from Anhui University in 1996 and Ph.D. degree from Beihang University in 2001. Now he is an Associate Professor in School of Electronic and Information Engineering, Beihang University. His current research focuses on RFID, Internet of Things and its application in aviation security. He has presided over NSFC, 863 Project etc. He has published more than 30 papers in journals, international conferences and workshops, and 5 books on RFID and Internet of Things. (E-mail: ninghuansheng@buaa.edu.cn)

LIU Hong was born in 1985 in Shandong Province. She is a Ph.D. candidate in School of Electronic and Information Engineering, Beihang University. Her research areas are security authentication protocol and architecture in RFID and IoT system. (E-mail: liuhongler@ee.buaa.edu.cn)

YANG Chen received Ph.D. degree from Xidian University in 2007. He is working in China Electronics Standardization Institute, and his research interests include cryptology, networks and information security, and information security standardization. (E-mail: yangchenyf@163.com)