The Improvement of RFID Authentication Protocols Based on R-RAPSE

QingLing Cai and YiJu Zhan
School of Engineering, Sun Yat-sen University, Guangzhou, China
Email: caiqingl@mail.sysu.edu.cn, zhanyiju@mail.sysu.edu.cn

Jian Yang
Faculty of Automation, Guangdong University of Technology, Guangzhou, China
Email: jimmyyoungyj@163.com

Abstract—Numerous RFID authentication protocols proposed often have some security vulnerabilities, since they lack systematic theory support. We had proposed a series of rules called R-RAPSE (RFID Authentication Protocol Security Enhanced Rules) for RFID authentication protocol design and verification. By three examples, this paper justifies why the protocols do not offer sufficient security and privacy protection, and thereafter, proposes relevant solutions to fix the security holes based on R-RAPSE, which demonstrates how R-RAPSE can be used to verify and improve RFID authentication protocols.

Index Terms—RFID; Authentication; Indistinguishability; Vulnerability; Privacy

I. INTRODUCTION

Radio frequency identification (RFID), as one of the most important component of the Internet of Things (IoT) sensing layer, has been widely deployed in various applications [1]. In particular, RFID systems with the low-cost RFID tag are more extensively applied. However, the embedded nature of the technology and a lack of awareness of its potential social and personal consequences make a special issue dedicated to security and privacy [2-12]. Although there are recent investigations of low cost RFID security protocols [2-10], it is challenging to deploy higher quality safety technology to low-cost RFID tags.

To resolve the security and privacy issues of RFID systems [2-5], numerous security protocols for RFID systems have been proposed [2-12]. However these protocols often have more or less vulnerabilities since they lack systematic theory support. Even RFID protocols are proposed by some senior experts, these RFID protocols may often have some weaknesses without regard to the comprehensive security issues and special requirements for RFID systems. As a result, a protocol was proposed; its security vulnerabilities were soon discovered. Such as HY Chien proposed an ultralightweight RFID authentication protocol [6], however, there are two de-synchronization attacks to break the protocol, as pointed out by Sun et al. [7]. H. Y. Chien et al. proposed another lightweight RFID protocol in the literature [8]; the algorithm did not take the distinguishability issues into consideration, and would allow the location of the tag owner to be exposed. That is, the tag can be tracked. Meanwhile, the weaknesses of the schemes [9-12] have been reported. Thus, it is imperative to propose principled rules to provide systematic theory support for RFID authentication protocols. Therefore, we had proposed a series of RFID authentication protocol security enhanced rules (R-RAPSE, Rules — RFID Authentication Protocol Security Enhanced) (see Appendix R-RAPSE), designed to provide systematic theory support for RFID authentication protocol design and verification.

In this paper, our aims are to present R-RAPSE applications. Based on R-RAPSE, we verify and improve E. J. Yoon’s scheme [2], J. S. Cho’s scheme [3] and S. I. Ahamed’s scheme [5]. In a more general sense, we show that these algorithms are insecure, which demonstrates how R-RAPSE is used to verify and improve RFID authentication protocols.

The contribution of the paper lies in the following aspects: This paper provides R-RAPSE applications. We demonstrate how R-RAPSE is used to verify and improve RFID authentication protocols. By three examples, we verified and improved RFID authentication protocols based on R-RAPSE. We justified why the protocols do not offer sufficient security and privacy protection, and thereafter, proposed relevant stronger protocols to fix the security holes.

The remaining parts of this paper are organized as follows: Section 2, by three examples, we identified some vulnerabilities of previous protocols and demonstrate how R-RAPSE is disobeyed by these protocols, and then propose relevant solutions to mend those security holes. Our conclusions are presented in section 3. We discuss the security issues, security requirements, R-RAPSE in Appendix R-RAPSE.

II. SECURITY ANALYSIS AND IMPROVEMENT OF RFID AUTHENTICATION PROTOCOLS BASED ON R-RAPSE

In this section, this paper verifies and improves three RFID authentication protocols [2, 3, 5] based on R-RAPSE. Firstly, we illustrate that the protocols have some vulnerabilities and analyze why the protocols do...
not offer sufficient security and privacy protection, and then suggest relevant solutions to mend the security vulnerabilities.

For each protocol considered in this section, due to space considerations, the paper only provides a rough description of the complicated steps necessary. The interested readers are referred to the original publications for the detailed descriptions of these protocols. Let us quickly revisit the protocol.

The universal notations used in this paper are given as follows:

- \( H(\cdot) \) a non-invertible one way hash function.
- \( P(\cdot) \) a pseudo-random number generator.
- \( \| \) concatenation operation.
- \( \oplus \) determine whether both sides are equal.
- \( DATA \) the corresponding information of the tag kept in the server.
- \( \oplus \) XOR operation.

\[ A. \text{ The J. S. Cho et al. Protocol} \]

J. S. Cho’s paper [2] presents a hash-based mutual authentication protocol. The J. S. Cho et al. protocol is designed to send a random number generated by a tag to the server without leakage. Moreover, it uses secret values instead of random numbers as response information, which is able to produce distinct response information in every session without interference from intentional or accidental requests generated by an adversary, and the secret value is not directly transmitted.

However, according to R-RAPSE, J. S. Cho’s protocol, as opposed to their claims, does not follow Data Integrity Rule, and the protocol may give rise to de-synchronization issues between the server and the tag. This can be accomplished as follows:

1. Prior conditions and assumptions

The communication between the reader and the server is secure, while the communication between tags and readers is insecure.

The notations used in J. S. Cho’s paper are given as follows:

- \( R_i \) random number generated by reader (96-bit).
- \( R_t \) random number generated by tag (96-bit).
- \( RID \) group ID of random number (96-bit).
- \( ID_s \) the ID of the Tag k.
- \( S_j \) secret value (96-bit) mutually shared by the server and the tag, and used in the jth session.
- \( C_1 \) information generated by tag for authentication.
- \( C_2 \) blind factor.
- \( S_j \) secret value (96-bit) mutually shared by the server and the tag, and used in the jth session.
- \( C_1 \) information generated by tag for authentication.

In initialization phase, the information shared by the server and each tag is kept within respective devices:

Tag: \((ID_s, S_j)\)
Server: \((ID_s, S_j, S_{j+1}, DATA)\)

2. Protocol description

J. S. Cho’s protocol authenticates an RFID tag with a secret value. This is represented by Fig. 1. Step 2, 5 and 8 of protocol are described as follows:

Step 2: Generating response information

The tag generates some values by the following steps:

- Generates: random number \( R_i \).
- Generates: \( RID := (R_t - R_t \mod S_j + 1)(0:47) \| (R_i + S_j - R_t \mod S_j)(48:95) \), where \( RID \) of group involve \( R_t \).
- Computes: \( C_1 := H(ID_s \oplus R_i \oplus R_t \oplus RID) \).
- Computes: \( C_2 := S_j(0:47) \| ID_s(48:95) \).
- Executes: \( R_t \oplus C_2 \).

Step 5: Tag authentication

The server authenticates the tag, and updates the secret value.

The server performs the following steps, based on the saved information of each tag.

- Retrieves \( ID_s \) and \( S_j \) of a tag in the table of the server.
- Generates \( C_1 \) with the retrieved \( ID_s \) and \( S_j \).
- Retrieves \( R_t \) from \( R_i \oplus C_2 \) with the generated \( C_2 \).
- Compute group \( RID' \) with the retrieved \( S_j \) and \( R_t \).
- Generates \( C_1' \) with \( R_t \) obtained from reader, and computed \( RID' \), retrieved \( R_t \) and \( ID_s \).
- Repeats steps (a) ~ (e) until \( C_1' \) is the same as \( C_1 \).

If the same \( C_1' \) is found, the server will send the relevant tag data to the reader and update the secret value of the tag. It performs the following steps.

- Updates the secret value \( S_j \) of the tag with \( S_{j+1} \) \((S_j \leftarrow S_{j+1})\), where \( S_{j+1} \) is generated at the discretion of the server.
- Modifies the table of the relevant tag. \( S_{j+1}, S_j \) are saved into row \( S_j, S_{j+1} \) respectively.
- Generates: \( DATA [H(C_2 \oplus RID)] \| R_t \oplus S_{j+1} \).
- Data to be transmitted the reader.
- \( H(C_2 \oplus RID) \) to authenticate the server for the tag.
- \( R_t \oplus S_{j+1} \) to update the secret value of the tag.
- If the same \( C_1' \) is not found, retrieves \( S_j \) from row \( S_{j+1} \) instead of \( S_j \), and repeats phase Step 5(1).
- It is the case that a synchronization issue has occurred in the previous session.
- If the process also fails, the protocol will be stopped at that time, and the transmitted information cannot be trusted.

Step 8: Server authentication

The tag authenticates the server, and updates the secret value.

The tag authenticates the server by \( H(C_2 \oplus RID) \).
- The tag retrieves the new secret value \( S_{j+1} \) from \( R_t \oplus S_{j+1} \) with \( R_t \).
- Updates the secret value \( S_j \) with \( S_{j+1} \), after the server passes through authentication.

3. Vulnerability analysis and improvement

Analysis:

The protocol does not follow Data Integrity Rule (see R-RAPSE). The adversary can desynchronize information in the server and the tag. This can be accomplished as follows:
Step 7 The adversary may send $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$ instead of $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$.

To do the following:

The adversary can acquire $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$ via eavesdropping.

At the same time, blocks $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$ to the tag.

Then sends $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$ to the tag, where $R_i \oplus S_{s+1}$ to the tag, where $R_i \oplus S_{s+1} = (R_i \oplus S_{s+1}) \oplus \Lambda$ (which may be any number 96-bit).

Step 8
Now the tag receives $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$ but not

$H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$.

The tag retrieves the new secret value $S_{s+1}$ from $R_i \oplus S_{s+1}$ with $R_i$.

Updates the secret value $S_i$ with $S_{s+1}$ (but not $S_{s+1}$).

At the moment, the de-synchronization attack will be successful.

Improvement:
According to Data Integrity Rule (see R-RAPSE), the hash function can be used to protect data integrity, by doing the following:

Step 7 Replaces $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$ with $H(C_2 \oplus RID_i)||R_i \oplus S_{s+1}$. $R_i \oplus H(S_{s+1})$, and transmits to the tag, where the hash function $H(.)$ can ensure $S_{s+1}$ data integrity.

$H(C_2 \oplus RID_i)$ to authenticate the server for the tag.

$R_i \oplus S_{s+1}$ to update the secret value of the tag.

$R_i \oplus H(S_{s+1})$ to verify $S_{s+1}$ data integrity for the tag.

B. The E. J. Yoon Protocol

Firstly, E. J. Yoon’s paper [3] demonstrates that Yeh et al.’s protocol [4] has two serious security issues such as data integrity problems and forward secrecy problems. Then E. J. Yoon’s paper proposes an improved protocol and claims that it can prevent these security issues and adapt to the low-cost RFID environments which require a high level of security.

However, according to R-RAPSE, E. J. Yoon’s protocol may give rise to distinguishability issues, exposing the location of the tag owner. This might result in the privacy of the tag owner being violated and the traffic of the tag owner to be analyzed. The analysis is described as follows:

1. Prior conditions and assumptions
The communication of between the server and the readers is insecure, and the communication between readers and tags is also insecure.

The notations used in E. J. Yoon’s paper are defined as follows:

$EPC_i$, the 96-bit of EPC code are divided into six 16-bit blocks, and then the six blocks perform XOR into $EPC_i$.

$K_i$ the access key kept in the tag to authenticate the server by the tag at the $(i+1)$th authentication session.

$P_i$ the access key kept in the tag to authenticate the server by the tag at the $(i+1)$th authentication session.

$K_{ni}$ the new authentication key kept in the server.

$K_{ni}$ the new authentication key kept in the server.

The communication of between the server and the readers is insecure, and the communication between readers and tags is also insecure.

$ID_i$ the reader identification number.

In initialization, the manufacturer generates some random values for the server, reader and tag, the information is stored in respective devices:

Tag: $(K_i, P_i, \alpha_i, EPC_i)$

Reader: $ID_i$

Server: $(K_{ni}, \alpha_{ni}, K_{ni}, P_i, \alpha_i, ID_i, EPC_i, DATA)$

2. Protocol description
The E. J. Yoon protocol proposes an improvement of Yeh et al.’s protocol. The information kept within respective devices is same as Yeh et al.’s protocol. Moreover, the initialization phase is same as in Yeh et al.’s protocol. The improved (i+1)th authentication session is depicted in Fig. 2. Step 7 and 11 of the protocol are described as follows:

**Step 7: Reader and tag authentication**

The server authenticates the reader and the tag, and updates the secret values.

After receiving \((M_1, \beta, \alpha, \gamma, R_\alpha, V)\), the server performs the following operations:

1. Extracts each stored ID, sequentially to compute \(H(ID_\alpha \oplus R_\alpha)\) with \(R_\alpha\), and matches the received \(V\) to identify the correct tuple and authenticate the reader.

2. Checks the value of \(\alpha\) in the tag to decide which of the two following procedures is preceded.

When \(\alpha = 0\), represents the first access. The server picks up a tuple \((K_\alpha, P_\alpha, \alpha, K_\beta, P_\alpha, \alpha, ID_\alpha, EPC_\alpha, DATA)\) stored in itself. Computes the values: \(I_1 = M_1 \oplus K_\alpha\) and \(I_2 = M_1 \oplus K_\beta\), and examines whether \(I_1\) or \(I_2\) matches \(P(EPC_\alpha, R_\alpha, \beta \oplus K_\alpha)\) or \(P(EPC_\alpha, R_\alpha, \beta \oplus K_\beta)\) computed by the server itself, where \(\beta \oplus K_\alpha\) or \(\beta \oplus K_\beta\) needs to obtain \(R_\alpha\), which is repeated for each tuple until a match is found.

Once the matching tuple is found, set \(x=n\) or \(x=o\) according to which authentication key \((K_\alpha \text{ or } K_\beta)\) in the tuple found matched with the one in the tag.

When \(\alpha \neq 0\), using \(\alpha\) as an index to find the corresponding tuple in the server. If the tuple is found by matching up by its field \(\alpha\), set \(x=\alpha\); otherwise set \(x=n\) if \(\alpha\) matches up. Then verify \(M_1\), received from the reader, if it is equal to \(P(EPC_\alpha, R_\alpha, \beta \oplus K_\alpha) \oplus K_\beta\) computed by the server itself.

Retrieves \(K_\alpha\) from the matched tuple and verifies whether the received \(\gamma\) matches \(R_\gamma \oplus (\alpha \oplus K_\alpha)\) computed by the server itself. If the two values do not match, then the protocol stops.

Computes \((M_2, Info, MAC)\) as follows:

\[
M_2=P(EPC_\alpha, R_\alpha) \oplus P_\alpha
\]

\[
Info=D_\alpha \oplus ID_\alpha
\]

\[
MAC=H(DATA \oplus R_\alpha)
\]

Finally forwards them to the reader.

If \(x=n\), then updates the tuple by replacing \(K_\alpha\) with \(K_\alpha\), \(P_\alpha\), with \(P_\alpha\) and \(\alpha\) with \(\alpha\) \oplus \(R_\alpha\), respectively. If \(x=\alpha\), then resets \(\alpha\) \oplus \(R_\alpha\) \oplus \(R_\alpha\).

**Step 11: Server authentication**

The server authenticates the server, and updates the secret value.

The tag extracts \(P_\gamma\) kept in to compute XOR with the received \(M_2\). If the product matches \(P(EPC_\alpha, R_\alpha)\) computed by the tag itself, the authentication to the server is completed.

The content kept in tag is update as \(K_{\alpha+1}=P(K_\alpha)\), \(P_{\gamma+1}=P(P_\gamma)\) and \(\alpha_{\gamma+1}=P(R_\gamma \oplus R_\alpha)\) for next session.

3. **Vulnerability analysis and improvement**

Analysis:

The protocol does not follow Indistinguishability Rule I, III (see R-RAPSE). In Step 7, Step 11 of this protocol, the server and the tag update \(K_\alpha, P_\alpha, \alpha\) respectively. However in Step 10, if an adversary blocks \(M_2\) and transmits a meaningless value \(M_2^*\) to the tag, the authentication will fail, the value \(K_\alpha, P_\alpha, \alpha\) in the tag will not be updated. In new session Step 4, the tag response information \((M_1, \beta, \alpha, \gamma)\), while \(\alpha\) remains constant. This way, the adversary may continuously send requests to the tag, which may give rise to a distinguishability issue (\(\alpha\) remains constant in each new session). The authentication of the tag owner can be exposed, and the privacy of the tag owner can be violated and the traffic of the tag owner can be analyzed.

**Improvement:**

According to Indistinguishability Rule III (see R-RAPSE), the transmitted information should be ensured randomization after the authentication failed to pass, and the tag key cannot be updated. In the new session Step 4, the tag response information is \((M_1, \beta, \alpha, \gamma)\), while \(\alpha\) remains constant. Thus, \(\alpha\) should be changed into a “random” number.

According to Indistinguishability Rule I (see R-RAPSE), we can introduce random numbers into the static information (constant) by some operations (such as XOR) to ensure indistinguishability of the tag. Modifying as follows: In Step 4, \(\alpha\) may be replaced with \(\alpha \oplus R_\alpha\).

![Figure 3. The protocol of S. I. Ahamed et al.](image-url)

**C. The S. I. Ahamed et al. Protocol**

S. I. Ahamed et al.’s paper presents an authentication protocol [5] that described as serverless, lightweight, forward secured and untraceable authentication protocol for RFID tags, and claims that the protocol ensures security against almost all major attacks without the intervention of server, and is very critical to guarantee intractability and scalability simultaneously.

However, according to R-RAPSE, S. I. Ahamed’s protocol may give rise to distinguishability issues. There is a possibility of exposing the location of the tag owner, violating the privacy of the tag owner can and analyzing the traffic of the tag owner. The analysis is described as follows:

1. Prior conditions and assumptions

   The communication between the reader and the tag is insecure, without the intervention of the server.

   The notations used in S. I. Ahamed’s paper are given as follows:
The protocol does not follow Key Synchronization Rule I, II (see R-RAPSE). In Step 4, Step 6 of this protocol, the reader and the tag update seed_{ij}, seed_{ij} respectively. However in Step 5, if an adversary blocks n_i and transmits a meaningless value n_i' to the tag, the authentication will fail and the tag seed_{ij} will not be updated. This way, the key of the reader and tag will become desynchronized.

Improvement:
According to Key Synchronization Rule II (see R-RAPSE), Step 4 and Step 6 adopt the Double Key Update method. Like this, the desynchronized issue can be prevented to a large extent.

Distinguishability Issue
Analysis:
The protocol does not follow Mutual Authentication Rule II and Indistinguishability Rule I, II, III (see R-RAPSE). As stated above (see II. C. 3 (1)), in Step 6, after the reader authentication fails, and the shared key between the reader and the tag become desynchronized, resulting in the fact that the tag seed_{ij} is not updated. The value seed_{ij} thus becomes a constant. At this time, the adversary may launch a new session with a request: R_i (where R_i=0). In Step 3 the adversary receives (n_j, R_i), where n_j=P(seed_{ij}⊕(0||R_i)). As seed_{ij} is a constant, the MSB (seed_{ij}⊕(0||R_i)) is a constant and the LSB (seed_{ij}⊕(0||R_i)) only is of 8-bit (PRNG is 16-bit), and R_i can be intercepted by the adversary. Based on attributes PRNG i, ii, iii, (see Appendix R-RAPSE. B), it is easy to obtain seed_{ij}, which may give rise to distinguishability issues.

The location of the tag owner can be exposed, the privacy of the tag owner can be violated and the traffic of the tag owner can be analyzed.

Improvement:
According to Indistinguishability Rule II (see R-RAPSE), we can choose the random number R_i, R_i with high entropy and without 0/1.

According to Indistinguishability Rule III (see R-RAPSE), the transmitted information should be ensured randomization after the authentication failed to pass, and the tag key cannot be updated. In the new session Step 3, the tag response information is n_j, R_j, while n_j remains constant. Thus, n_j should be changed into a “random” number.

According to Indistinguishability Rule I (see R-RAPSE), we can introduce random numbers into the static information (constant) by some operations (such as XOR) to ensure indistinguishability of the tag. Modifying as follows: In Step 2, seed_{ij} may be replaced with seed_{ij}⊕(R_i||R_j)’, where R_i, R_j are two random numbers generated by the tag.

According to Mutual Authentication Rule II (see R-RAPSE), a single RN16 by itself cannot effectively protect the transmitted information to secure the information based on attributes PRNG i, ii, iii. Thus, a single RN16 by itself is not suitable for the encryption and authentication tools of RFID systems.

III. CONCLUSIONS

Design and verification of RFID authentication protocol should be supported by theoretical guarantees, rather than depending on intuition and experience. We had summed up a series of rules called R-RAPSE (see Appendix R-RAPSE). This paper’s aims are to present R-RAPSE applications. Based on R-RAPSE, we verify and improve E. J. Yoon’s scheme [2], J. S. Cho’s scheme [3] and S. I. Ahamed’s scheme [5]. In a general sense, we show that these algorithms are insecure, and demonstrate how R-RAPSE is used to verify and improve RFID authentication protocols. The results of the study suggest that R-RAPSE will be significant for design and validation of RFID authentication protocols.

IV. APPENDIX R-RAPSE

Security Issues and Requirements: The security issues of RFID systems mainly includes eavesdropping, unauthorized access, tampering, key leakage, key update
issue, privacy leakage, traffic analysis, location tracks replay attacks, impersonation attacks, DoS attacks, de-synchronization attacks, and cloning attacks. Thereinto, privacy leakage, traffic analysis, location tracks and cloning attacks are the special security issues for RFID systems. Thereinto, privacy leakage, traffic analysis, location tracks and cloning attacks are the special security issues for RFID systems.

We had summarized the following security requirements based on the essence of the security issues within RFID systems. The security requirements mainly include: indistinguishability, confidentiality, data integrity, forward security, impersonation resistance and de-synchronization resistance. Thereinto, indistinguishability is the special security requirements for RFID systems.

RFID Authentication Protocol Security Enhanced Rules (R-RAPSE): Design and verification of RFID authentication protocol should be supported by theoretical guarantees, rather than depending on intuition and experience. Based on the specific security issues and security requirements of RFID systems, we had summed up a series of rules called R-RAPSE, designed to provide systematic theory support for RFID authentication protocol design and verification. R-RAPSE is described as follows:

A. Lightweight Rule (LR): RFID authentication protocols should be lightweight. Restrained by cost and resources, the low-cost RFID tag is only able to support HASH, PRNG, XOR, and CRC.

B. Mutual Authentication Rule (MAR): MAR is necessary and critical. One of the parties (readers and tags) involved in a transaction needs to verify the identity of the other in RFID systems. Without the mutual authentication rule, it is possible for either or both of the parties to engage in illegitimate operations.

In RFID systems, algorithms that can usually be used to establish authentication protocols primarily include HASH, PRNG and CRC; their performances are analyzed as follows:

| HASH | The hash function usually possesses three additional attributes when it is employed in cryptography [13, 14]: Authorization verification. Signature verification. File verification. Based on attributes HASH i, ii, the hash function is suitable for RFID authentication protocols that implement mutual authentication between the reader and the tag. Based on the attribute HASH iii, the hash function can be used to protect the transmitted information between the readers and the tags from tampering, thus ensuring data integrity. PRNG Being consistent with EPC Gen2, this paper defines the low-cost RFID tag as having a 16-bit pseudo-random number generator. The attributes are as follows [10, 14]: Probability of a single RN16: \( 0.82^{16} \times \text{Prob}(\text{RN16}=\text{RN}) \times 1.25/2^{16} \). Collisions: \( \text{Prob} < 0.1 \% \), for a tag population of 10,000. Predicting a RN16: \( \text{Prob} < 0.025 \% \), if the prior draws are known.

Based on attributes PRNG i, ii, iii, a single RN16 by itself cannot effectively ensure the transmitted information security. Thus, a single RN16 by itself is not suitable for the encryption and authentication tools of RFID systems.

CRC

A 16-bit CRC can be generated as follows: CRC possesses three attributes:

\[
CRC(A \oplus B) = CRC(A) \oplus CRC(B) \tag{1}
\]

\[
CRC(A \cdot x^{n-1}) = CRC(CRC(\cdots CRC(A))) \rightarrow CRC^k(A) \tag{2}
\]

If \( P, Q \) are 16-bit numbers, then

\[
CRC(P(x) + Q(x)) = CRC(P(x)) + CRC(Q(x)) + CRC(0) \tag{3}
\]

Based on CRC generation algorithms and CRC attributes, the CRC by itself cannot effectively protect against the intentional (malicious) alteration of data [10]. Therefore, CRC by itself is not suitable for the encryption and authentication tools of RFID systems.

C. Key Update Rule (KUR): To ensure forward security, keys or secret values should be updated after each successful authentication. It guarantees forward security by disconnecting the relation between the previous key and secret values.

D. Key Synchronization Rule (KSR): Key synchronization mostly depends on the key update methods. Based on the number of keys, existing key update methods can be categorized as follows:

Single Key Update: the server and tag only preserve the single latest key. Based on the mode, it is generally difficult to maintain key synchronization. If one party (such as the server) has updated its key successfully and the other party (such as the tag) cannot update its key, the key shared between the server and the tag will become desynchronized.

Double Key Update: The server and the tag preserve the double latest keys. Based on the mode, it is possible to maintain key synchronization. If the authentication

| TABLE I. THE RELATIONSHIP OF SECURITY ISSUES, SECURITY REQUIREMENTS AND SECURITY RULES. |
|-------------------------------------------------|-------------------------------------------------|
| R-RAPSE | Security Issues | Security Requirements |
| Confidentiality Rule Mutual Authentication Rule | Eavesdropping Unauthorized access | Confidentiality |
| Data integrity Rule | Tampering | Data integrity |
| Indistinguishability Rule Mutual Authentication Rule | Privacy leakage Traffic analysis Location track | Indistinguishability |
| Confidentiality Rule Indistinguishability Rule Mutual Authentication Rule | Replay attack Cloning attack | Impersonation resistance |
| Key Update Rule | Key leakage | Forward security |
| Key Synchronization Rule | Key update issue DoS attack De-synchronization attack | De-synchronization resistance |

© 2014 ACADEMY PUBLISHER
fails, and one party (such as the server) has updated its key successfully and the other party (such as the tag) cannot update its key, the updated party can still take advantage of the previous key (the backup key) instead of the present key (the updated key), thus the desynchronized issue can be prevented to a large extent.

E. Scalability Rule (SR): There is a need for efficient communication protocols that allow the scalable deployment of RFID systems with a large number of tags [16, 17].

F. Confidentiality Rule (CR): To ensure confidentiality of the tag, the reader and the tag should be complied with the following:

RFID tags should only be allowed access by authorized readers.

RFID tags should not leak any secret information to unauthorized readers.

Minimizing the information exposure as possible (even random numbers) [9].

G. Indistinguishability Rule (IR): To ensure indistinguishability of the tag, the transmitted information should be randomized. The reader and the tag should be complied with the following:

Ensuring indistinguishability of the tag by introducing random numbers into the static information (remains constant) by some operations (such as XOR).

Ensuring transmission information security when the random numbers are all 0s or all 1s.

Ensuring randomization of the transmitted information after the authentication failed to pass, and the tag key cannot be updated.

H. Data Integrity Rule (DIR): To protect the important information (such as keys or secret values) from being tampered, and to ensure validity of the information, DIR should be followed. Based on the attribute HASH iii, the hash function can be used to protect the data integrity.

V. CONCLUSIONS

The general principles of RFID authentication protocols should comply with Rules A-E.

The transmitted information of RFID authentication protocols should comply with Rules F-H.

The relationship of security issues, security requirements and security rules is given in Table I.

ACKNOWLEDGEMENTS

This work is sponsored by Project 61071038 supported by NSFC (National Natural Science Foundation of China), Project 61102034 supported by NSFC, Project 61172156 supported by NSFC, and the Combination Project of Industry and Research by the Ministry of Education of Guangdong Province 2011A090200128.

REFERENCES


professor in School of Engineering, Sun Yat-sen University. Her research interests include information security, communication security and wireless networks.

YiJu Zhan has received his B. S. degree in Electrical Engineering, Hefei University of Technology, P. R. China, 1982, M. S. degree in Electrical Engineering, Hefei University of Technology, P. R. China, 1986, and Ph. D. degree in Electrical Engineering, the University of Hong Kong, in 1998. Currently, he is a professor in School of Engineering, Sun Yat-sen University. His research interests include automation control, intelligent information, automation distinguishing technology and wireless networks.

Jian Yang received the B. S. degree and M. S. degree in Biology Engineering from Northeastern University, Ph. D. degree in Communication and Information System from Sun Yat-sen University, China, in 2003, 2006 and 2009, respectively. His areas of specialized research interest include cognitive network, RFID MAC protocol and intelligent signals processing. Now he is a lecturer of faculty of automation, Guangdong University of Technology.