A DYNAMIC JOIN-LEAVE MULTI-PURPOSE SCHEME FOR RFID INFRASTRUCTURE

Iuon-Chang Lin
Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan

Rui-Kun Luo, Shih-Chang Tsaur
Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

Keywords: RFID, Security, Dynamic join-leave, Multi-purpose.

Abstract: Previously, a multi-purpose RFID tag needs one purpose stored one key in the tag and server, otherwise there will be security issues. In this paper, we propose a scheme for multi-purpose RFID tags. In this scheme, user can dynamic join-and-leave the purposes in multiple applications. Moreover, just authorized tags can be accessed. Our scheme is capable of applying multi-purpose tags, and we just need to store one key in a tag. Our approach will not cause the large computations of the database.

1 INTRODUCTION

In our daily life identification is a very common thing. For instance, the sale of things we will use the barcodes record number of goods. We pick up the money use the smart cards record of personal data. Compared with barcodes and smart cards, Radio frequency identification (RFID) can read a large number of tags simultaneously. And RFID can read the far distance object, does not to be in the line of vision.

RFID system contains three parts: tag, reader and back-end server. Reader sends RF signals to the tag, tag returns information for the reader sent to the host computer to do the follow-up treatment. Tag radio frequency identification systems are the information carriers, tag consist of antenna and chip. Pursuant to tag the different power supply, tag can be for active tag, passive tag, semi-passive tag. Active tag and semi-passive tag containing a battery, passive tag is not inside the battery. Tag based on the frequency can be divided into low frequency (LF), high frequency (HF), Ultra high frequency (UHF), microwave. RFID reader and RFID tag through the antenna for wireless communications, to tag identification code and data memory read or write.

General RFID usually only used in a functional. It is only certified reader can interrogated tags. For example, campus system only campus system's reader can be interrogated tags, transport system only transport system's reader can be interrogated tags, bank system only bank system's reader can be interrogated tags. Campus System tag in the transport system cannot be certified, since the tag system on campus only campus system of authentication key (Weis et al., 2004; Ohkubo et al., 2003; Molnar and Wagner, 2004; Avoine et al., 2006; Feldhofer et al., 2004; Menezes et al., 1997; Dimitriou, 2005). To store other system's key will have to increase memory capacity. Kaya et al. presented by the use of public key approach (Kaya et al., 2009), can solve the key problem of key too many, use this approach tag only store one key. However, we found that the use of public key approach will be the key issue of sharing. For example, the tag will be use on campus system and transport system. Campus systems and transport systems to share their own private key to agreement a public key. This would cause to reveal secrets. In our proposed, we suggest a scheme that tag only storage the authorized applications secret. Our approach will not cause the large compute of the database. And we promise the safety and the protection of.

2 KAYA ET AL.'S MULTI-PURPOSE SCHEME

This section we will propose Kaya et al.'s multi-purpose scheme and its weaknesses.

Nations Definition.

- *Ticket*: $E_S(RID + ReaderLocation +$

Lin I., Luo R. and Tsaur S..
A DYNAMIC JOIN-LEAVE MULTI-PURPOSE SCHEME FOR RFID INFRASTRUCTURE.
DOI: 10.5220/0003404205370540
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems* (LOCSUE-2011), pages 537-540
ISBN: 978-989-8425-48-5

$TicketIssuanceTime$).

- *Nonce*: A random number serves as a challenge
- *Response*: $E_R(Nonce)$.
- *Credential*: $E_{pub}(Ticket + TID + Nonce + E_R(Nonce))$.
- *RID*: Reader ID.
- *TID*: Tag ID.
- *S*: A secret key used by back-end server to generate (encrypt) and decrypt *Ticket*.
- *R*: Key shared between the reader and the back-end server.
- *pub*: Public key of back-end server.
- *priv*: Private key of back-end server.

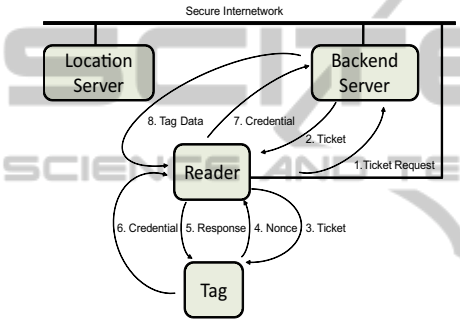

Figure 1: Kaya et al.'s scheme for multi-purpose architecture.

Communications Steps.

Step 1 : Request *Ticket* (Reader → Back-end server) : First reader sends a *Ticket* requests for back-end server.

Step 2 : *Ticket* (Back-end server → Reader) : Back-end server sends a message *Ticket* to reader.

Step 3 : *Ticket* (Reader → Tag) : Reader sends *Ticket* to tag.

Step 4 : *Nonce* (Tag → Reader) : Tag receives the *Ticket* from reader and sends a *Nonce* to reader.

Step 5 : *Response* (Reader → Tag) : Reader receives the *Nonce* and then sends a response to tag.

Step 6 : *Credential* (Tag → Reader) : Tag receives *Response* and encrypts its *Ticket* , *TID*. Then tag sends *Credential* to the reader.

Step 7 : *Credential* (Reader → Back-end server) : Reader sends *Credential* to back-end server.

Step 8 : Tag data (Back-end server → Reader) : Back-end server receives *Credential* and verifies *Ticket*. If *Ticket* is authorized by server, server sends the tag data to reader.

Cryptographic Operations Performed by the Back-end Server of the Protocol.

Step 1 : $D_{priv}(Credential) = D_{priv}(E_{pub}(Ticket + TID + Nonce + E_R(Nonce))) = Ticket + TID + Nonce + E_R(Nonce)$.

Step 2 : $D_S(Ticket) = D_S(E_S(RID + ReaderLocation + TicketIssuanceTime)) = RID + ReaderLocation + TicketIssuanceTime$.

Step 3 : Look for *R* from the server and check if $D_R(E_R(Nonce)) == Nonce$ obtained in Step 1.

However, the scheme proposed by Kaya et al. and previous other authors scheme have a weaknesses. Their proposed scheme cannot dynamically join and leave. In Kaya et al. scheme, they assume the back-end servers are trusted in their model, and the back-end servers share the same public key and private key. This was not scalable. If an attacker adds this groups to obtain the private key, the system will be collapse. All message can be decrypted. Otherwise, if back-end servers store different private keys. When a new purpose is added to the groups, in the public key encryption algorithms all of the public key must be renewed. Tags should not select the application the tags want to add. Since a new application join and tags use new public key encrypt messages. All of the back-end servers also decrypt messages.

3 OUR PROPOSED SCHEME

This section we will propose a dynamic join and leave scheme for multi-purpose RFID infrastructure.

Abbreviations Symbols Definition.

- LS : Location server.
- BS : Back-end server.
- CA : Certification authority.
- R : Reader.
- T : Tag.

Nations Definition.

- *Ticket*: $E_S(RID + ReaderLocation + TicketIssuanceTime)$.
- *Nonce*: A random number serves as a challenge.
- *Response*: $E_R(Nonce)$.
- *Credential*: $E_K(Ticket + TID + Nonce + E_R(Nonce))$.
- *RID*: Reader ID.

- TID: Tag ID.
- S: A secret key used by back-end server to generate (encrypt) and decrypt $Ticket$.
- K: A secret key used by tags to generate (encrypt) and decrypt $Credential$.
- R: Key shared between the reader and the server.
- X: A random number.
- r: A random number.
- B: Registers number used by tags, store in tags and CA.
- $H(*)$: Hash function.
- Key generation: CA registers their purposes tag. K and r to encrypt B. B using function generated, $B = \Pi H(K_i \parallel r)$, where B, r, are stored in tag.
- Some step at the tag: Tag selects a key K. U generates, $U = B \times X + K$. Then sends U, $Credential$, R to the back-end server to decrypt.
- Authentication: Back-end server, using the private key as long as the K declassified $D_K(Credential)$, verifies the accuracy of $Credential$ on the line.

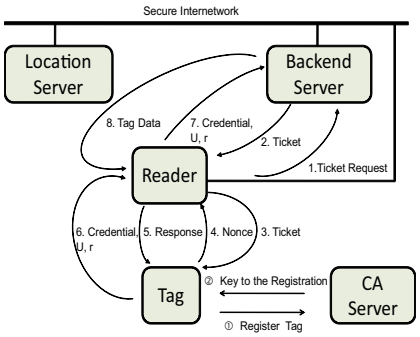

Figure 2: Our scheme.

Assumptions.

The protocol assumes that there are trusted location servers that can either track the readers or locate them when asked by the back-end server. Tag stores B, r. Reader stores R. back-end server stores S, R, K.

Before communications, tag must be registered with the CA. CA would tag purpose for registration to the corresponding purpose of the communications group owned password B.

Communications Step.

Step 1 : Request $Ticket$ (R $\rightarrow$ BS) : First reader sends a $Ticket$ request to back-end server.

Step 2 : $Ticket$ (BS $\rightarrow$ R) : Back-end server sends a message $Ticket$ to reader.

Step 3 : $Ticket$ (R $\rightarrow$ T) : Reader sends $Ticket$ to tag.

Step 4 : $Nonce$ (T $\rightarrow$ R) : Tag receives the $Ticket$ from reader and sends a $Nonce$ to reader.

Step 5 : $Response$ (R $\rightarrow$ T) : Reader receives the $Nonce$ and then responds to tag.

Step 6 : $Credential$ (T $\rightarrow$ R) : Tag receives the $Response$ and use K encrypts $Ticket$, TID, $Nonce$, $E_R(Nonce)$ as $Credential$. Then tag sends $Credential$, U, r to the reader.

Step 7 : $Credential$ (R $\rightarrow$ BS) : Reader sends $Credential$, U, r to back-end server.

Step 8 : Tag data (BS $\rightarrow$ R) : Back-end server receives $Credential$, U, r and computes K = U mod $H(K_i \parallel r)$ to verify K. If K is authorized by back-end server, then back-end server sends tag data to reader.

Cryptographic Operations Performed by the Back-end Server of the Protocol.

Step 1 : Compute K = U mod $H(K_i \parallel r)$.

Step 2 : $D_K(Credential) = D_K(E_K(Ticket + TID + Nonce + E_R(Nonce))) = Ticket + TID + Nonce + E_R(Nonce)$.

Step 3 : $D_S(Ticket) = D_S(E_S(RID + ReaderLocation + TicketIssuanceTime)) = RID + ReaderLocation + TicketIssuanceTime$.

Step 4 : Look for R from the server and check if $D_R(E_R(Nonce)) == Nonce$ obtained in Step 1.

4 DISCUSSIONS

In this section, we analyze security of our protocol against impersonation, replay, tracking, and Dynamic Join-Leave.

4.1 Impersonation Attacks

A security RFID architecture should not have the fake reader or tag. A malicious reader cannot impersonate another reader and get access to information pertaining to a tag, since there is a secure and authenticated channel between the reader and the back-end server. And the impersonate tags cannot authenticate the reader and the back-end server, since no tag id is sent in clear text. Tag id is encrypted within the $Credential$ with symmetric key of the CA.

4.2 Replay Attacks

A reader can use the same *Credential* repeatedly authentication until the measurement time expires. Ways to prevent the replay attacks are measurement time of location that serves as a timestamp and random nonce values that change in every tag query. In the proposed scheme the reader id contain in *Credential*. The illegitimate reader cannot eavesdrop and use a credential intended for another reader. And the channels between reader and back-end server are secure, only legitimate reader can authenticate.

4.3 Traceability

If tag id is sent in clear text or the tag responds to the reader's queries always in the same way, an attacker can track the tags by the message. In the proposed scheme the tag id is encrypted in *Credential*. The tag id and the credential are both with random nonce. So tag id and the responds to the reader's queries change in every time. Reader cannot track the tags by the message.

4.4 Dynamic Join-leave

The schemes proposed by Kaya et al. and previous other author have a weaknesses. Their proposed scheme cannot dynamically join and leave. In Kaya et al. scheme, they assume the back-end servers are trusted in their model, and the back-end servers share the same public key and private key. This was not scalable. If an attacker is added to this groups to obtain the private key, the system will collapse. All messages can be decrypted. Otherwise, back-end servers store different private keys. When a new purpose is added to the groups, in the public key encryption algorithms all of the public key must be renewed. Tags should not select the application the tags want to add. Since a new application joins and tags use new a public key to encrypt messages. All of the back-end servers also decrypt messages.

However, in our scheme. We use mod function to solve key management of Kaya et al.'s scheme. Our scheme use CA the trust server register all tags keys U. When server will authentication a tag, and server computes $K = U \bmod H(K_i \parallel r)$ to verify K. If K is authorized by back-end server, then back-end server sends tag data to reader. So in our scheme, authentication key can dynamically join-and-leave.

The security comparisons among Kaya et al.'s scheme and our scheme are listed in Table 1.

Table 1: The security scheme comparison among Kaya et al.

Security	Kaya et al.'s scheme	Our scheme
Impersonation attacks	◯	◯
Replay attacks	◯	◯
Traceability	◯	◯
Dynamic Join-Leave	×	◯

5 CONCLUSIONS

The proposed scheme presents an application on multi-purpose scheme. In accordance with the needs of tag dynamically add and remove different applications. While the U.S. may lead to, other than Kaya et al.'s scheme more computing schemes, however, our scheme can solve the disadvantage that the user can not add and remove. Furthermore, our scheme is also able to satisfy all the security requirements of Kaya et al.'s scheme.

REFERENCES

Avoine, G., Dysli, E., and Oechslin, P., 2006. Reducing time complexity in RFID systems. *Lecture Notes in Computer Science - Selected Areas in Cryptography 2006*, 3897:291–306.

Dimitriou, T., 2005. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Proceedings of IEEE First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 59–66, Athens, Greece.

Feldhofer, M., Dominikus, S., and Wolkerstorfer, J., 2004. Strong authentication for RFID systems using the AES algorithm. *Lecture Notes in Computer Science - Cryptographic Hardware and Embedded Systems - CHES 2004*, 3156:357–370.

Kaya, S. V., Savaş, E., Levi, A., and Erçetin, O., 2009. Public key cryptography based privacy preserving multi-context RFID infrastructure. *Ad Hoc Networks*, 7:136–152.

Menezes, A., van Oorschot, P., and Vanstone, S., 1997. *Handbook of Applied Cryptography*.

Molnar, D. and Wagner, D., 2004. Privacy and security in library RFID: Issues, practices, and architectures. In *Proceedings of 11th ACM conference on Computer and communications security*, pages 210–219, Washington, DC, USA.

Ohkubo, M., Suzuki, K., and Kinoshita, S., 2003. Cryptographic approach to privacy-friendly tags. In *RFID Privacy Workshop*.

Weis, S. A., Sarma, S. E., Rivest, R. L., and Engels, D. W., 2004. Security and privacy, aspects of low-cost radio frequency identification systems. *Lecture Notes in Computer Science - Security in Pervasive Computing 2003*, 2802:50–59.