Journal of Computers Vol. 27, No. 3, 2016, pp. 41-51
doi:10.3966/199115592016102703005

An Efficient Ownership Transfer Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards

Fan-Shuo Liu[1] Hai-Bing Mu[1]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China
13120095@bjtu.edu.cn, hbmu@bjtu.edu.cn

Received 1 July 2015; Revised 27 July 2015; Accepted 10 August 2015

Abstract. Radio Frequency Identification (RFID) is a new recognition technology used to identify each item in commodity. During trading, tags which are attached to goods will be read by the readers and their ownership need transferring to complement the transaction process and service. Tags and readers are apt to be attacked during the communication between them due to the wireless channel and the weak resource of the RFID tag. Tags in trading also need forward and backward security besides the ordinary security requirements in other circumstances. There are many previously proposed protocols while most may suffer from spending problem because they use hash or keyed encryption function. In this paper we propose a secure ownership transfer protocol that conforms to the EPCglobal Class 1 Generation 2 (C1G2) standards. Our work aims to propose a low-cost protocol, based on simple XOR, Cyclic Redundancy Check (CRC) and 16-bit pseudo-random number generators (PRNG). We also put a trust model in which we make some assumption which is useful to introduce our protocol. Furthermore, we add a sign bit to mark the tag's state to introduce a "release-transfer-resumption" mode. The analysis shows that the proposed scheme can resist common attacks and ensure a security communication.

Keywords: EPC, Ownership transfer, RFID, Security

1 Introduction

RFID technology has become a general technology. It makes the identification of object without touching come true. A typical RFID system consists of tags, reader and database. Tags that carry the pivotal information are attached to object. This technology has been widely used in supply chain management, logistics, healthcare, defense and many other fields. However, RFID tags and readers just provide limited calculation resource, which makes it difficult to deal with the security risks in wireless communication.

The secure ownership transfer of objects from one owner to another has become an important feature of RFID system. For example, the manufacturer transports products to the distribution center, and then to warehouse, to retailer, the ownership of the products always changes frequently in the supply chain [1]. In our knowledge, the process of ownership transfer has two security needs: (1) only the new owner could identify and communicate with the tag, the old owner would not identify and trace the tag (2) the new owner has no way to know the communication process between the old owner and the tag. And we also use the two conditions to judge ownership having been transferred from the old owner to the new owner. Formal definitions are put for secure ownership and ownership transfer provided by van Deursen, Mauw, Radomirovic and Vullers [2].

Now, we define a classification method of ownership transfer protocol and put a trust model. The secure ownership transfer protocols can be broadly classified into double databases and single database. In the former, both the old owner and the new owner use private database, during the process of ownership transfer, the data should be transmit by secure channel (or rely on a trusted intermediary). In the trust model, that tag meets the needs of low-cost means it just save ID and only use low-cost calculation, which makes the tag always readable but only the current owner could identify and get useful information.

The rest of the paper is organized as follows. Section 2 covers the related work in this area. The trust

model is introduced in Section 3 in detail. Our proposed protocol is described in Section 4, followed by the detailed security analysis in Section 5. Section 6 concludes the paper.

2 Related work

One of the earliest schemes for ownership transfer was proposed by Osaka, Takagi, Yamazaki and Takahashi [3] which is based on hash and keyed encryption functions. Protocol contains writing stage and ownership transfer stage. In the writing stage, symmetric secret key between tag and server should be updated to k' for ownership transfer phase. In the transfer of ownership stage, k' and other pivotal information would be transferred, and then the k' is updated to k" to complete the transfer of ownership. The cost does not apply to EPC C1G2.

Japinnen and Hamalainen [4] proposed a protocol to improve the original scheme [3]. However, the improvement brought desynchronization problems that had been verified [5]. For the noise injection problem, Chen, Lee, Zhao and Chen [6] used a hash function to protect the key being transferred. This scheme still suffers from desynchronization issues.

Molnar, Soppera and Wagner [7] proposed a protocol based on key-tree, and referred to the problem of ownership transfer at the end of the article. There are two ways to achieve ownership transfer of the tag: (1) “soft failure”, the new owner B communicates with a trusted center TC (Trusted Center) and learns the number k which was authorized to the original owner A. B reading the tag k + 1 times that makes A can’t read the tag. (2) “increased tag count”, c is the tag’s count, B sends a new count value c' and c' > c after B and tags established a secure channel, which makes the tag pass all pseudonym authorized for A and A can’t read the tag. The protocol is not completely transferred ownership, but “pseudo shift.”

Chen, Huang and Jiang [8] proposed an ownership transfer protocol applies for retail and follows the EPC standards. In the purchase phase, the whole authentication consists of six steps. If A wants to buy the product, A enters his personal password PW on the reader in the seventh step, the reader uses PW to make a hash function and sends the value of H(PW) to the server, then the server writes H (PW) in the matched list. In ownership transfer phase, A sales the object to B, both A and B need to enter the personal password (PW and PW_{new}), the reader sends H (PW) and H (PW_{new}) to the server, the server authenticates H(PW) and makes the update operation. After that, the server sends the acknowledgment Y and C_2 to the reader, the reader uses SN to update the tag after affirms the right of Y and C_2. This protocol just uses CRC to protect the communication which cannot prevent the tag being forged.

The mobile phone is used as a reader in [9]. First, the mobile reader needs to register on the server. During the ownership transfer process, a sends pivotainformation to B, B achieves the ownership transfer by a third party AA (Authorized Agent). Li, Hu, He and Pang [10] use a sign F_d to transfer ownership after two server authenticate each other.

3 The trust model and assumptions

3.1 Trust mode

In the supply chain, tag is attached to goods for identification. For the manufacturer and retailer, they are cooperation partners in the economy. They need to share much information for common benefit, though there are some key information must to be protected. Based on this relation, we make an assumption that the old owner shouldn’t do any attack action during the ownership transfer process. So the attack’s source may be the old or new owner after the ownership transfer process or an individual attacker during the process.

In the trust mode, we use a flag f which is set to 1 for no-transferring and 0 for transferring to present tag’s state and divide the ownership transfer into three steps: (1) ownership release; (2) information transfer; (3) ownership resumption. In the first step, the old owner change the f’s value to turn the tag’s state into transferring; in the second step, the old owner gives some import information to the new owner; in the last step, the new owner changes the value of f to turn the tag’s state into no-transferring and gets the ownership.

3.2 Notation

Table 1. Notations

SIGN	Explanation
M_{req}	Request information.
N_i	Random number.
f	The sign of tag' status.
PID_i	Pseudo identification code.
key_s	The key of common.
CRC	Cyclic Redundancy Check.
PRGN	Pseudo-random number generators.
EPC_i	EPC identification code.
DATA	Product information.
key_c	The key of ownership transfer.

4 An RFID Authentication Protocols

4.1 Common communication process

The f's value of tag is 1, which means the tag hold no-transferring states, when the tag receiving the request information from the reader it would use keys in calculation; The f's value in database is 1, which means the tag should not be transferred, when the server has recognized the tag it uses keys in calculation. This protocol is shown in Fig.1.

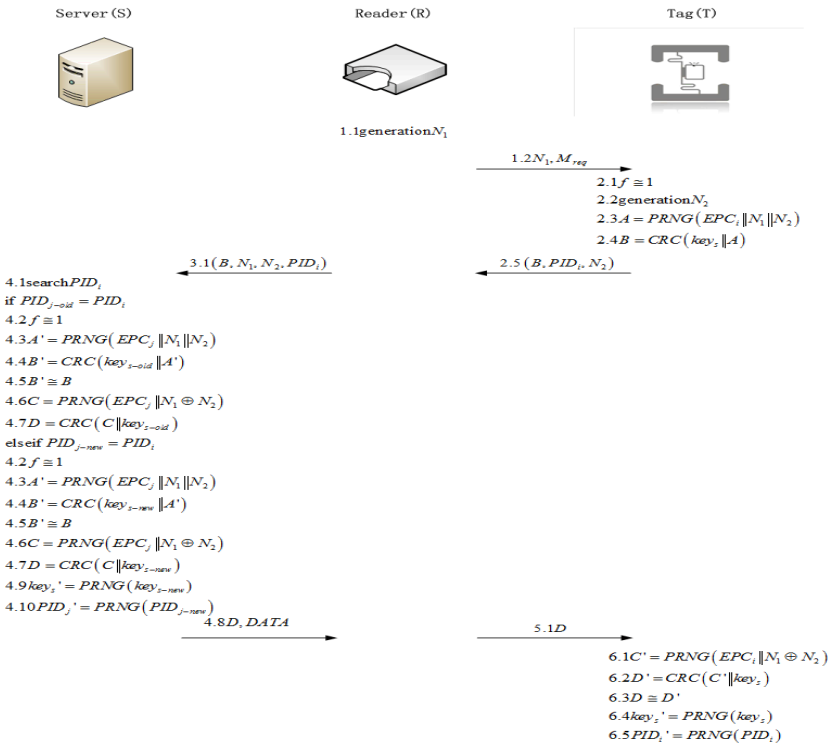

Fig. 1. Common communication process

Step1: The reader generates a nonce N_1, and then sends M_{req} and N_1 to the tag.

Step2: Upon receiving the request message, the tag checks f's value to choose the key; after that, it generates a fresh nonce N_2 and computes A, B:

$$A = PRNG\left(EPC_i \left\| N_1 \right\| N_2\right) \quad \textbf{(1)}$$

$$B = CRC\left(key_s \| A\right) \tag{2}$$

Then, it responds (B, N_2, PID_i) to the reader.

Step3: Reader receives the message from tag and forwards (B, N_2, N_1, PID_i) to the server.

Step4: After receiving the authentication request from the reader, the server checks its database. The tag is illegal if there is no matched PID_j, and the communication process would be terminated. If there is a $PID_{j\text{-}old}$ equals PID_i, which means the tag suffering denial of service attacks during the previous authentication, the server checks the k's value and computes A' and B':

$$A' = PRNG\left(EPC_j \| N_1 \| N_2\right) \tag{3}$$

$$B' = CRC\left(key_{s-old} \| A'\right) \tag{4}$$

If B' equals B, the authentication of the tag is finished. Then, the server computes C and D:

$$C = PRNG\left(EPC_j \| N_1 \oplus N_2\right) \tag{5}$$

$$D = CRC\left(C \| key_{s-old}\right) \tag{6}$$

The server sends (D, DATA) to the reader without updating operation.

If there is a $PID_{j\text{-}new}$ equalling PID_i, which means the previous authentication is prefect, the server checks the k's value and computes A', B':

$$A' = PRNG\left(EPC_j \| N_1 \| N_2\right) \tag{7}$$

$$B' = CRC\left(key_{s-new} \| A\right) \tag{8}$$

If B' equals B, the authentication of tag is finished. Then, the server computes C and D:

$$C = PRNG\left(EPC_j \| N_1 \oplus N_2\right) \tag{9}$$

$$D = CRC\left(C \| key_{s-new}\right) \tag{10}$$

The server computes key_s' and PID_j' as follows:

$$key_s' = PRNG\left(key_{s-new}\right) \tag{11}$$

$$PID_j' = PRNG\left(PID_{j-new}\right) \tag{12}$$

The server sends (D, DATA) to the reader and does update operation that replaces $key_{s\text{-}old}$ with $key_{s\text{-}new}$, replaces $key_{s\text{-}new}$ with key_s', replaces $PID_{j\text{-}old}$ with $PID_{j\text{-}new}$, replaces $PID_{j\text{-}new}$ with PID_j'.

Step5: Reader receives the message from the server and sends D to the tag.

Step6: After receiving the message from the reader, tag computes C' and D':

$$C' = PRNG\left(EPC_i \| N_1 \oplus N_2\right) \tag{13}$$

$$D' = CRC\left(C' \| key_s\right) \tag{14}$$

If D' equals D, the authentication of reader and server is finished. Then, the tag computes PID_i' and key_s':

$$key_s' = PRNG\left(key_s\right) \tag{15}$$

$$PID_i' = PRNG\left(PID_i\right) \tag{16}$$

The tag does the update operation that replaces key_s with key_s' and replaces PID_i with PID_i'.

4.2 Ownership transfer

The process of ownership transfer can be divided into three phases: ownership release, information transfer and ownership resumption.

4.2.1 Ownership release

In this phase, the old owner communicates with the tag and uses key_c to change tag's state into transferring. This protocol is shown in Fig.2.

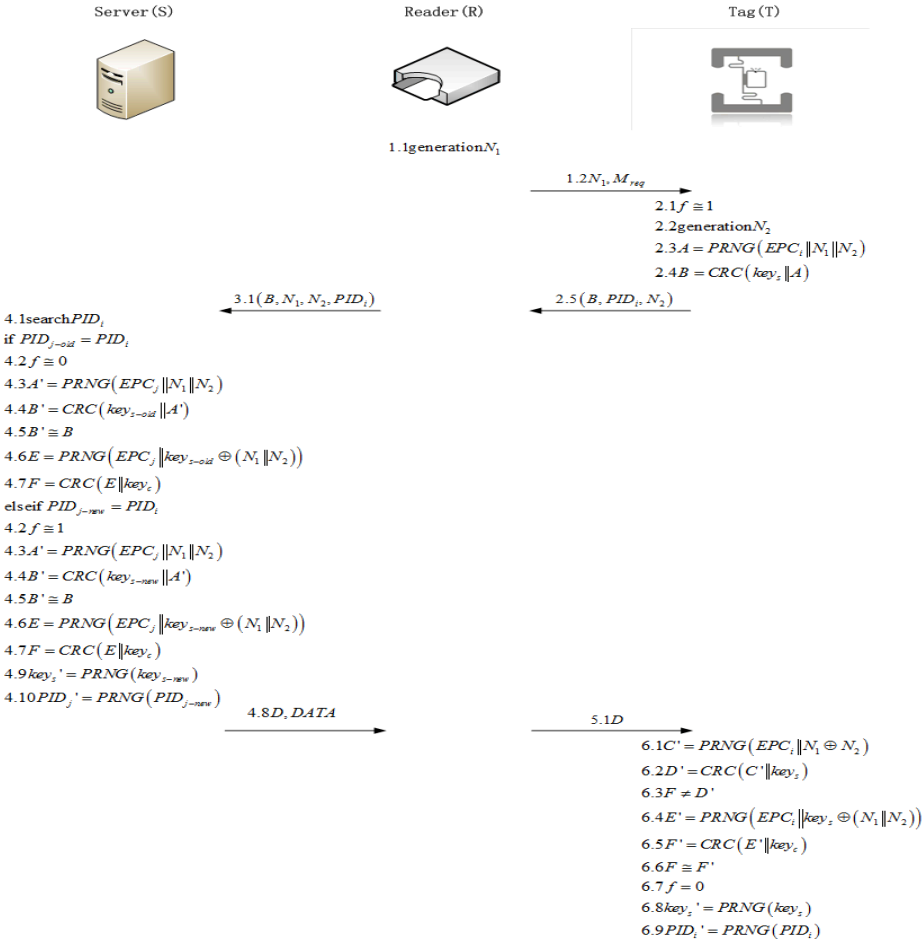

Fig. 2. Ownership release

Step1: The reader generates a nonce N_1, and sends M_{req} and N_1 to the tag.

Step2: Upon receiving the request message, the tag chooses the key according to the value of f; after that, it generates a fresh nonce N_2 and computes A, B:

$$A = PRNG\left(EPC_i \left\| N_1 \right\| N_2\right) \tag{17}$$

$$B = CRC\left(key_s \left\| A\right.\right) \tag{18}$$

Then, it responds (B, N_2, PID_i) to the reader.

Step3: Reader receives the message from tag and forwards (B, N_2, N_1, PID_i) to server.

Step4: After receiving the authentication request from the reader, the server checks its database. The tag is illegal if there is no matched PID_j, and the communication process would be terminated. The f's value in database is 0 which means the tag needs to be transferred. If there is a $PID_{j\text{-}old}$ equalling PID_i, which means the tag suffering denial of service attack during the previous authentication, the server computes A', B':

$$A' = PRNG\left(EPC_j \| N_1 \| N_2\right) \quad \textbf{(19-1)}$$

$$B' = CRC\left(key_{s-old} \| A'\right) \quad \textbf{(20-1)}$$

If B' equals B, the authentication of the tag is finished. Then, the server computes E and F:

$$E = PRNG\left(EPC_j \| \left(key_{s\_old} \oplus \left(N_1 \| N_2\right)\right)\right) \quad \textbf{(21-1)}$$

$$F = CRC\left(E \| key_c\right) \quad \textbf{(22-1)}$$

The server sends (F, DATA) to the reader without updating operation.

If there is a $PID_{j\text{-}new}$ equalling PID_i, which means the previous authentication is prefect, the server computes A', B':

$$A' = PRNG\left(EPC_j \| N_1 \| N_2\right) \quad \textbf{(19-2)}$$

$$B' = CRC\left(key_{s-new} \| A\right) \quad \textbf{(20-2)}$$

If B' equals B, the authentication of tag is finished. Then, the server computes E and F:

$$E = PRNG\left(EPC_j \| \left(key_{s\_new} \oplus \left(N_1 \| N_2\right)\right)\right) \quad \textbf{(21-2)}$$

$$F = CRC\left(E \| key_c\right) \quad \textbf{(22-2)}$$

The server computes key_s' and PID_j' as follows:

$$key_s' = PRNG\left(key_{s-new}\right) \quad \textbf{(23)}$$

$$PID_j' = PRNG\left(PID_{j-new}\right) \quad \textbf{(24)}$$

The server sends (F, DATA) to the reader and does update operation that replaces $key_{s\text{-}old}$ with $key_{s\text{-}new}$, replaces $key_{s\text{-}new}$ with key_s', replaces $PID_{j\text{-}old}$ with $PID_{j\text{-}new}$, replaces $PID_{j\text{-}new}$ with PID_j'.

Step5: Reader receives the message from the server and sends F to the tag.

Step6: After receiving the message from the reader, tag computes C' and D':

$$C' = PRNG\left(EPC_i \| N_1 \oplus N_2\right) \quad \textbf{(25)}$$

$$D' = CRC\left(C' \| key_s\right) \quad \textbf{(26)}$$

If D' is not equal to F, the tag tries to compute E' and F':

$$E' = PRNG\left(EPC_i \| \left(key_s \oplus \left(N_1 \| N_2\right)\right)\right) \quad \textbf{(27)}$$

$$F' = CRC\left(E' \| key_c\right) \quad \textbf{(28)}$$

If F' equals F, the tag changes f's value into 0 and computes PID_i' and key_s':

$$key_s' = PRNG\left(key_s\right) \quad \textbf{(29)}$$

$$PID_i' = PRNG\left(PID_i\right) \quad \textbf{(30)}$$

The tag does the update operation that replaces key_s with key_s' and re places PID_i with PID_i'.

4.2.2 Information transfer

The old owner gives the necessary data (k, EPC, $PID_{j\text{-}new}$, $key_{c\text{-}new}$, $key_{s\text{-}new}$) to the new owner by secure channel.

4.2.3 Ownership resumption

The new owner recovers the tag from transferring in to no-transferring. This protocol is shown in Fig.3.

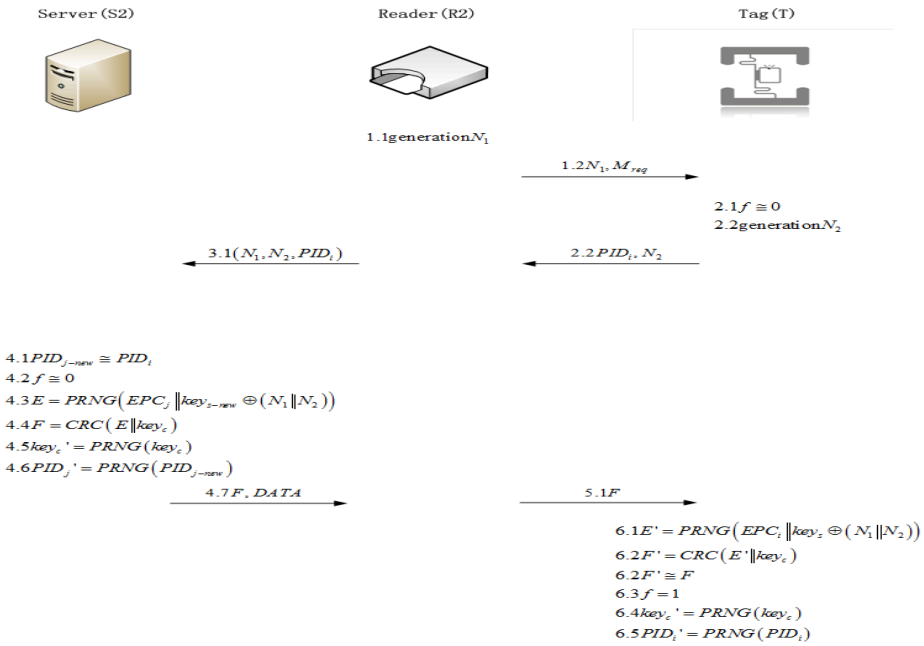

Fig. 3. Ownership resumption

Step1: The reader generates a nonce N_1, and then sends M_{req} and N_1 to the tag.

Step2: Upon receiving the request message, the tag checks f's value; after that, it generates a fresh nonce N_2 and responds (N_2, PID_i) to the reader.

Step3: Reader receives the message from tag and forwards (N_2, N_1, PID_i) to theserver.

Step4: After receiving the authentication request from the reader, the server checks its database. The tag is illegal if there is no matched PID_j, and the communication process would be terminated. The f's value in database is 0 which means the tag in transfer state. If there is a $PID_{j\text{-}new}$ equaling PID_i, the server computes E and F:

$$E = PRNG\left(EPC_j \left\| \left(key_{s\_new} \oplus \left(N_1 \| N_2\right)\right)\right.\right) \tag{31}$$

$$F = CRC\left(E \| key_c\right) \tag{32}$$

The server computes key_s' and PID_j' as follows:

$$key_s' = PRNG\left(key_{s-new}\right) \tag{33}$$

$$PID_j' = PRNG\left(PID_{j-new}\right) \tag{34}$$

The server sends (F, DATA) to the reader and does update operation that replaces $key_{s\text{-}old}$ with $key_{s\text{-}new}$, replaces $key_{s\text{-}new}$ with key_s', replaces $PID_{j\text{-}old}$ with $PID_{j\text{-}new}$, replaces $PID_{j\text{-}new}$ with PID_j'.

Step5: Reader receives the message from the server and sends F to the tag.

Step6: After receiving the message from the reader, tag computes E' and F':

$$E' = PRNG\left(EPC_i \left\| \left(key_s \oplus \left(N_1 \| N_2\right)\right)\right.\right) \tag{35}$$

$$F' = CRC\left(E' \| key_c\right) \tag{36}$$

If F' equals F, the tag changes f's value into 1 and computes PID_i', key_s':

$$key_s' = PRNG\left(key_s\right) \tag{37}$$

$$PID_i' = PRNG\left(PID_i\right) \tag{38}$$

The tag does the update operation that replaces key_s with key_s' and re places PID_i with PID_i'.

5 Security analysis and discussion

In this part, we will discuss the protocol's security. First, we need talk something about the attacker M. Our protocol conforms to the EPCglobal C1G2 standards, so M could monitor the whole communication. Besides, M could be a manufacturer who once owned the tag, he could know more detail about our protocol. To make an attack, M must monitor an intact communication. Some important information learned by the legitimate communication is used to guess the authentication information.

5.1 Resist tag impersonation attack

For the identification by the server, after receiving the request message M needs to response B_M, PID_i and N_2' to the reader. M may guess B_M by XOR calculation:

$$B \oplus B_K = CRC\left(key_s \| A\right) \oplus CRC\left(key_s \| A_K\right) \tag{39}$$

$$B_K = B \oplus CRC\left(key_s \| A\right) \oplus CRC\left(key_s \| A_K\right) \tag{40}$$

$$B_K = B \oplus CRC\left(A \oplus A_K\right) \tag{41}$$

$$B_K = B \oplus CRC\left(PRNG\left(EPC_i \| N_1 \| N_2\right) \oplus PRNG\left(EPC_i \| N_1' \| N_2'\right)\right) \tag{42}$$

In order to calculate B_M, M need A's value, but A don't participate communication, so M can't get A by listening that M can't forge tag by guessing B_M.

5.2 Resist server impersonation attack

To Initiate communication, M sends M_{req} and N_1' to the tag. When receiving the request message from tag, M responses D_M. M may guess D_M by XOR calculation:

$$D \oplus D_K = CRC\left(key_s \| C\right) \oplus CRC\left(key_s \| C_K\right) \tag{43}$$

$$D_K = D \oplus CRC\left(key_s \| C\right) \oplus CRC\left(key_s C_K\right) \tag{44}$$

$$D_K = D \oplus CRC\left(C \oplus C_K\right) \tag{45}$$

$$D_K = D \oplus CRC\left(PRNG\left(EPC_i \| N_1 \oplus N_2\right) \oplus PRNG\left(EPC_i \| N_1' \oplus N_2'\right)\right) \tag{46}$$

In order to calculate D_M, M needs C's value, but C don't participate communication, so M can't get C by listening that M can't forge server by guessing D_M.

5.3 Resist location tracking

The attacker M needs to listen to a legitimate communication process and record the traced target T. The process is as follow:

$$(1)R \to T : M_{req}, N_1 \tag{47}$$

$$(2)T \to R : B, PID_i, N_2 \tag{48}$$

For tracing the target, M listens to another communication process：

$$(1)R \to T : M_{req}, N_1' \tag{49}$$

$$(2)T \to R : B', PID_i', N_2' \tag{50}$$

Because the value of PID_i must be updated each communication, M can't judge if the two tags are the same one by PID. This protocol could resist trace attack.

5.4 Resist replay attack

If M pretends to be a reader, first, M sends M_{req},N_1 to the tag, then M receives B', PID_i', N from tag, at last M sends D to tag. M can't use B, PID_i, N_2 to decode EPC_i and keys, so K can't use the listened information to make other attack; the tag calculates C' and D' after receiving D:

$$C' = PRNG\left(EPC_i \| N_1 \oplus N_2'\right) \tag{51}$$

$$D' = CRC\left(C' \| key_s'\right) \tag{52}$$

But D' not equals D, tag can't make the authentication about M, and tag doesn't update PID_i and keys yet, so the data between the tag and the database could keep synchronization.

If M pretends to be a tag, when listening to a new communication, M sends B, PID_i, N_2 to the reader. The sever searches in the database to find a $PID_{j\text{-}new}$ equaling $PID_i$and computes A', B':

$$A' = PRNG\left(EPC_j \| N_1' \| N_2\right) \tag{53}$$

$$B' = CRC\left(key_{s-old} \| A'\right) \tag{54}$$

But B' is not equal to B, the server can't make the authentication about M, and the communication must be interrupt, the data between the tag and the database would never be changed.

5.5 Resist replay attack

During the fifth step of the legitimate communication, M changes D into D_M. The tag receives D_M and computes C', D':

$$C' = PRNG\left(EPC_i \| N_1 \oplus N_2'\right) \tag{55}$$

$$D' = CRC\left(C' \| key_s'\right) \tag{56}$$

Because D' not equals D, tag can't make the authentication about server, and tag doesn't update PIDi and keys yet, but the server has updated the data after sending information，which makes the data between the tag and the database not synchronization.

The database keeps $PID_{j\text{-}old}$, $PID_{j\text{-}new}$, $key_{s\text{-}old}$ and $key_{s\text{-}new}$, when the reader communicates with an attacked tag, the server could find $PID_{j\text{-}old}$ equals PID_i, and uses $key_{s\text{-}old}$ to compute A', B':

$$A' = PRNG\left(EPC_j \| N_1 \| N_2\right) \tag{57}$$

$$B' = CRC\left(key_{s-old} \| A'\right) \tag{58}$$

If B' equals B, the server computes C and D：

$$C = PRNG\left(EPC_j \| N_1 \oplus N_2\right) \quad \textbf{(59)}$$

$$D = CRC\left(C \| key_{s-old}\right) \quad \textbf{(60)}$$

Tag receives D from the reader and computes C' and D':

$$C' = PRNG\left(EPC_i \| N_1 \oplus N_2\right) \quad \textbf{(61)}$$

$$D' = CRC\left(C' \| key_s\right) \quad \textbf{(62)}$$

If D' equals D, tag computes PID_i', keys', and does the update operation, which makes the tag and the database synchronized in data again.

5.6 The desynchronization of PRNG

In our protocol, the PRNG is used to update the pivotal information. The information could keep synchronization if there was on attack. The attack behavior makes the times of PRNG different that leads to the desynchronization issue.

Because the database uses more room to save the new and old information, the server could detect the attack behavior. When the attack behavior has been found, the server interrupts the current communication and sends a special broadcast to let the tag reset the PRNG that solve the desynchronization issue.

6 Conclusion

In this paper, we built a trust model for Double-database System. Then, we proposed a Secure Ownership Transfer protocol based on the model. This protocol is ultra-lightweight and meets the EPC C1G2 standard, in which only CRC, XOR and 16 bit PRNG functions are employed. Security analysis shows that the protocol can resist tag impersonation attack, server impersonation attack, location tracking, replay attack and man-in-the-middle attack. In future work, our aim is to improve the protocol and propose a version for single-database.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grant 61201159.

References

[1] Saravanan Sundaresan, Robin Doss, Wanlei Zhou. (2015). Secure ownership transfer for multi-tag multi-owner passive RFID environment with individual-owner-privacy. *Computer Communications, 55*(C), 112-124.

[2] van Deursen, T., Mauw, S., Radomirovic', S., & Vullers, P. (2009, September). *Secure ownership and ownership transfer in RFID systems*. Paper presented at the 14th European Symposium on Research in Computer Security-ESORICS 2009, Saint-Malo, France.

[3] Osaka, K., Takagi, T., Yamazaki, K., & Takahashi, O. (2006, November). *An efficient and secure RFID security method with ownership transfer*. Paper presented at the 2006 International Conference on Computational Intelligence and Security, Guangzhou, China.

[4] Japinnen, P., & Hamalainen, H. (2008, December). *Enhanced RFID security method with ownership transfer*. Paper presented at International Conference on Computational Intelligence and Security (CIS'08), Suzhou, China.

[5] Kapoor, G., & Piramuthu, S. (2010). Vulnerabilities in some recently proposed RFID ownership transfer protocols. *IEEE*

Communication Letter, 14(3), 260-262.

[6] Chen, H., Lee, W., Zhao, Y., & Chen, Y. (2009, January). *Enhancement of the RFID security method with ownership transfer*. Paper presented at the Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC'09), Suwon, Korea.

[7] Molnar, D., Soppera, A., & Wagner, D. (2005, August). *A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags*. Paper presented at Proceedings oftheSAC'05:Proceedings ofthe 2005 ACM Symposium on Applied Computing, New York, NY.

[8] Chen, C.-L., Huang, Y.-C., & Jiang, J.-R. (2013). A secure ownership transfer protocol using EPCglobal Gen-2 RFID. *Telecommunication Systems, 53*(4), 387-399.

[9] Chen, C.-L., & Chien, C.-F. (2013). An ownership transfer scheme using mobile RFIDs. Wireless *Personal* Communications, *68*(3), 1093-1119.

[10] Li, H.-X., Hu, J.-H., He, L.-W., & Pang, L.-J. (2012, November). *Mutual authentication and ownership transfer scheme conforming to EPC-C1G2 standard*. Paper presented at the 8th International Conference on Computational Intelligence and Security IEEE Computer Society, Guangzhou, China.