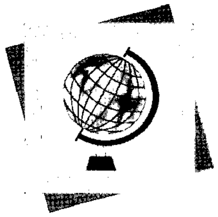
| 해외기고 02 |



A Survey on Cryptographic Techniques for RFID privacy

Junichiro SAITO · Kouichi SAKURAI
(Kyushu University)

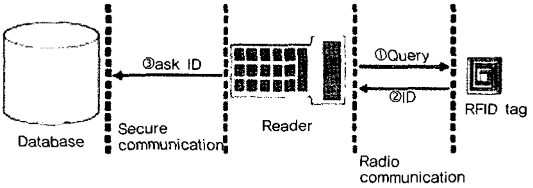
목 차

1. Introduction

A Radio-Frequency-Identification (RFID) tag is a small and cheap device which combined IC tip and an antenna for radio communications, and it emits an ID in response to query from a reader which is a radio communication device. For this reason, a RFID tag is used for management of goods and its circulation, and it is used as a substitute for a bar code in the future.

However, since a RFID system has a strong tracing ability, it may infringe on a consumer's privacy. Since RFID tags have fixed ID, you can find an object attached an RFID tag. Moreover, a location of a tag's owner can be leaked by using strong traceability of RFID tags. The privacy of owner's location is called as location privacy.

To these problems, there are some privacy protection schemes which can change ID information by using encryption circuit. In [1], RFID tags can change ID information by using one-time pad. [2] uses hash function to change ID information. [3] is a re-encryption scheme called as Universal Re-encryption. Moreover, we introduce Blocker tag which doesn't use encryption circuit [5].



(Figure 1) Basic RFID system

2. Privacy problem on RFID tags

The communication between a reader and an

RFID tag is performed by radio. Thus, it is simply tapped by an attacker. The reader can simply derive information from the RFID tag and it can be used to infringement of the privacy. There are two privacy problems on the RFID tag. First is the leakage of ID information. Since the RFID tag has unique ID, if the attacker obtains the ID, he can get information about objects that the tag was attached. For example, the size and the price of clothes, the contents of a wallet, the inventory information about the goods of a store etc. can be leaked. As a result, it infringes on the owner's privacy. We can protect this problem by using anonymity of ID information by using encryption scheme. Therefore, the attacker can not know what encrypted ID means.

schemes which can change ID information periodically to protect location privacy. Since ID information is not fixed by using these schemes, the attacker cannot trace specific tags.

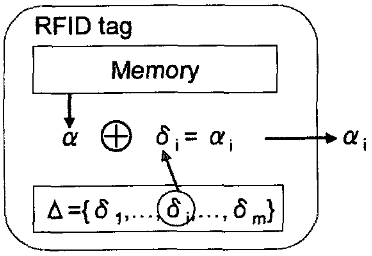
3. Implementation of cryptography

Because an RFID tag is inexpensive and small device, it cannot perform complex calculations. For example, public key cryptosystem is too costly to perform by RFID tags. The cost of RFID tags is 0.05/unit, and tags are small as 0.4mm*0.4mm and thin enough to be embedded in paper. For this reason, the processing capacity of RFID tags is limited. There are some researches on implementation of cryptography on RFID tags or some tiny devices.

In [6], the Advanced Encryption Standard (AES) [7] is integrated into the existing ISO/IEC 18000 standard [8]. The implementation has a chip area of 3,595 gates and has a current consumption of 8.15μA at a frequency of 100 kHz.

In [9], mCrypton designed for ubiquitous computing device is implemented. Crypton is a candidate of AES [10]. Crypton is designed for various implementation environments. The mCryption is a small version of Crypton to implement in tiny devices. It uses the parameters of 64-bit block length and 128-bit key length. The prototype implementation requires 4,262 gates for mCrypton encryption and decryption processor. Moreover, it requires 3,082 gates for encryption only.

On the other hand, sensor nodes in sensor networks have more powerful capability. In [11], public key cryptosystem is implemented in sensor networks. In [11], sensor nodes can perform public key encryption architectures with power consumption of less than 20μW using optimization.


RFID tag
Memory
$\alpha \oplus \delta_i = \alpha_i$
α_i
$\Delta=\{\delta_1,\dots,\delta_i,\dots,\delta_m\}$

(Figure 2) XORing using Onetime pad

4. XORing using one-time pad

Juels proposed a scheme which uses one-time pad to change ID information on RFID tags [1]. One-time pad is a table of *n* elements. The element of one-time pad and ID information are used for XORing and generate a different output. ID on RFID tags is encrypted to prevent from leaking.

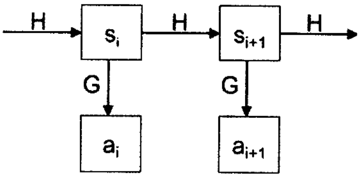
In this scheme, RFID tags have one-time pad

$\Delta = \{\delta_1, \delta_2, \ldots, \delta_m\}$ which is generated randomly and encrypted ID information α. Therefore, *i*th session is done as follows.

- An RFID tag receives a query from a reader.
- The RFID tag selects ith element δ_i in a one-time pad Δ, and XOR it and α.
- The RFID tag submits an output α_i of the XORing.

We show the process of RFID tags in (Fig. 2). When you retrieve ID information in a database, the database retrieves by using ciphertexts α and corresponding one-time pad.

Since one-time pad has *m* elements, RFID tags reuse an elements which is used before after *m* session is finished. Therefore, we should update the one-time pad. Then, we should prevent an attacker from altering the one-time pad. So an RFID tag and the database must authenticate each other. In this scheme, we use ciphertexts α, β, γ which is encrypted by using different keys. Since these ciphertexts are shared with the RFID tag and the database, we can update the one-time pad securely.



(Figure 3) Low-cost hash chain

In this scheme, RFID tags calculate only XORing. Therefore, we can implement this scheme on RFID tags low cost. Moreover, the retrieval cost on database is also low. However, since the one-time pad has limited memory capacity, we should update the one-time pad frequently.

5. Low-cost hash chain

Privacy protection scheme [2] uses two hash functions to change ID information on RFID tags. In this scheme, since RFID tags calculate hash function by using internal variable, you cannot find corresponding outputs from previous outputs. Moreover, RFID tags use two hash functions, one is used for changing internal variable, and the other is used for changing output value. So you cannot predict next output from previous outputs even if you know the hash function for outputs. We show that ID information is changed by using two hash functions in (Fig. 3).

In this scheme, an RFID tag processes following calculation in *i*the session with a reader. Here, *H* and *G* are hash functions, *s* is an internal variable in the RFID tag, and *a* is ID information which is submitted from the RFID tag to the reader.

- The RFID receives a query from the reader.
- The RFID tag submits $a_i = G(s_i)$ to the reader
- The RFID tag calculates $s_{i+1} = H(s_I)$ and keeps s_{i+1}.

An attacker cannot get s_i by using a_i because the hash function *G* is a one-way function. Since she cannot get s_{i+1}, she cannot get a_{i+1}. Therefore, we can protect location privacy. Moreover, even if the RFID tag is tampered and the attacker get an internal variable s_i, we can protect the history of outputs $\{s_1,\dots,s_{i-1}\}$ because of the one-wayness of the hash function *H*.

Since RFID tags calculate only hash functions, we can easily implement the scheme on RFID tags. However, a database should retrieve ID information until a corresponding ID information is found. For this reason, the database should calculate hash functions by using all ID information the database has. If the database has a large number of ID information, the calculation cost on the database is very high.

6. Re-encryption scheme

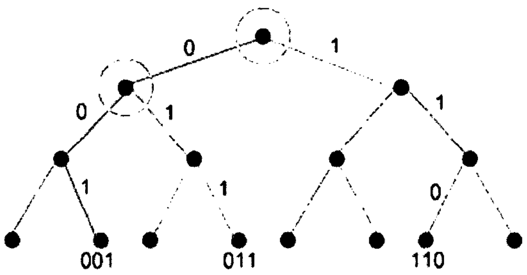
To change ID information on RFID tags, there are some schemes which use re-encryption of public key cryptosystem. Unlike the usual re-encryption scheme, universal re-encryption does not need knowledge of a public key in the case of re-encryption, but re-encryption is performed by determining a random number [3]. Moreover, re-encryption does not need the information about plaintext unlike encryption. The ciphertext which was processed re-encryption repeatedly can be once decrypted to the original plaintext using the private key.

If universal re-encryption is used for RFID tags, we can request a reader to update ID information. Moreover, since universal re-encryption doesn't need public key and plaintext, we can request third party's reader. Therefore, universal re-encryption is more secure than other re-encryption scheme because we can update ID information more frequently.

The protocol of universal re-encryption based on ElGamal is shown below.

- Key generation: Output secret key x and public key ($y=g^x$).
- Encryption: Ciphertext $C=[(\alpha_0,\beta_0);(\alpha_1,\beta_1)]$ is generated from the following formulas using message m, public key y, and random number $r=(k_0,k_1)$. $C=[(\alpha_0,\beta_0);(\alpha_1,\beta_1)]$, $\alpha_0=my^{k_0},\beta_0=g^{k_0},\alpha_1=y^{k_1},\beta=g^{k_1}$·Ciphertext C is written in a RFID tag.
- Decryption: A reader receives ciphertext C from the RFID tag, and sends to a database. A database calculates decryption algorithm described as follow. Compute $m_0=\alpha_0/\beta_0$ and $m_1=\alpha_1/\beta_1$ using ciphertext $C=[(\alpha_0,\beta_0);(\alpha_1,\beta_1)]$ under a public key y from the RFID tag and a secret key x. If $m_1=1$, then output message $m=m_0$. Otherwise the decryption fails, and a special symbol ⊥ is output. A given key can be decrypted only under one given key. It will get the message m_0 as ID of the RFID tag. Even if the ciphertext C is re-encrypted many times, it can return to plaintext by decryption at once. And the database searches the information on the RFID tag using its ID, and transmits it to the reader for reading ID information.
- Re-en
- cryption: The reader derives ciphertext $C=[(\alpha_0,\beta_0);(\alpha_1,\beta_1)]$ from the RFID tag, and sends it to the database. The database selects random number $r'=(k'_0,k'_1)$. And the database generates new ciphertext C' by calculating the formula described as follow. $C=[(\alpha'_0,\beta'_0);(\alpha'_1,\beta'_1)]=[(\alpha_0\alpha_1^{k'_0},\beta_0\beta_1^{k'_0});(\alpha_1^{k'_1},\beta_1^{k'_1})]$. Re-encrypted ciphertext C' is written in the RFID tag by the reader.

In this scheme, we can change ID information on RFID tags by re-encrypting. However, if an attacker can alter ID information on a RFID tag, she can trace the RFID tag. Even if re-encryption is performed, ID information is not changed if $C=[(\alpha_0,\beta_0);(\alpha_1,\beta_1)]=[(1,1);(1,1)]$. Therefore, we need a check scheme which prevents the attacker from altering. We proposed a re-encryption scheme with a check [4]. In this scheme, RFID tags check ciphertext to prevent from altering when a reader writes it.



(Figure 4) Tree-Walking

7. Blocker tag

Blocker tag exploit the singulation procedure called as the binary tree-walking. The binary tree-walking is an anti-collision scheme. When two or more RFID tags submit different signals at once, collision occurs and a reader cannot distinguish the signal. Binary tree walking is composed of reader's next bit query and tag's response. We assume there are three ID information (*001, 011, 110*). In the binary tree-walking, IDs are located at binary tree's leaves like (Fig. 4). First, then the reader queries the first bit to RFID tags, the reader recognizes that collision occurs. Therefore, the reader queries the next bit to RFID tags whose first bit is *0*. By repeating this process, the reader can recognize all ID information.

Blocker tag always submit *1* and *0* signal to cause collision. By causing collision on purpose, the reader can not know ID information in the communication range of blocker tag. Therefore, if you have blocker tag, you are not troubled with privacy problem on RFID tags any more.

〈Table 1〉 Comparison of privacy protection schemes

	Update frequency	Calculation cost on RFID tags	Retrieval costt on a database	ID leakage	Location privacy
XOR [1]	Each output (Depend on the size of one-time pad)	XOR	XOR (Using all ID information)	Encryption	One-time pad
Hash chain [2]	Each output	Hash	Hash (Using all ID information)	Hash function	One-time pad
Re-encryption [3]	Depend on a reader	Noting (Require to protect tampering)	Decryption of public key cryptosystem	Encryption	Re-encryption

8. Comparisons

We compare schemes which use cryptosystems about these update frequency, calculation cost, retrieval cost, and security. About update frequency, XOR scheme and lo-cost hash chain change ID information every at each outputs. On the other hand, re-encryption scheme needs re-encryption process by a reader to change ID information. Therefore, update frequency of re-encryption scheme depends on the reader. However, we can change ID information more frequently by requesting a third party.

Next, we compare about calculation cost on RFID tags. XOR scheme and low-cost hash chain need XORing and hash function on RFID tags respectively. These calculation cost is very low. In re-encryption scheme, RFID tags don't perform any calculations because the reader performs re-encryption. However, RFID tags must check altering ID information by an attacker. If RFID tags checks ID information by decrypting re-encrypted ciphertext, the cost is too high to implement on RFID tags.

Finally, we compare about retrieval cost on a database. In XOR scheme and low-cost hash chain, the database must retrieve ID information until corresponding ID is found. The database processes XORing or hash function by using all IDs the database has. Even if the calculation cost of XORing and hash function, the retrieval cost on the database is very large if the database has the large number of ID information. On the other hand, we can get ID by decrypting ciphertext at once in re-encryption scheme.

9. Conclusion

In this paper, we introduce some privacy protection schemes and compare these schemes. To prevent infringement of privacy, we should hide tag's ID and change ID information. We think not only the calculation cost on RFID tags but also the retrieval cost on the database is important.

References

[1] A. Juels, "Minimalist Cryptography for RFID Tags", Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers Series: Lecture Notes in Computer Science, Vol. 3352.

[2] M. Ohkubo, K. Suzuki and S. Kinoshita, "Cryptographic Approach to a Privacy Friendly Tag", RFID Privacy Workshop, 2004.

[3] P. Golle, M. Jakobsson, A. Juels and P. Syverson, "Universal Re-Encryption for Mixnets", In Topics in Cryptology -- CT-RSA'04, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings Series: Lecture Notes in Computer Science, Vol. 2964.

[4] J. Saito and J. C. Ryou and K. Sakurai, "Enhancing Privacy of Universal Re-encryption Scheme for RFID Tags", Embedded and Ubiquitous Computing, International Conference EUC 2004, Aizu-Wakamatsu City, Japan, August 25-27, 2004, Proceedings Series: Lecture Notes in Computer Science, Vol. 3207.

[5] A. Juels, R. L. Rivest and M. Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy", ACM Press. 2003.

[6] M. Feldhofer, S. Dominikus and J. Wolkerstorfer, "Strong Authentication for RFID systems Using the AES Algorithm", Cryptographic Hardware and Embedded Systems -- CHES 2004, 6th International Workshop Cambridge, MA, USA, August 11-13, 2004, Proceedings Series: Lecture Notes in Computer Science, Vol. 3156.

[7] National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, 2001. http://www.itl.nist.gov/fipspubs/

[8] International Organization for Standardization. ISO/IEC 13000-3. Information Technology AIDC Techniques -- RFID for Item Management, 2003.

[9] C. H. Lim and T. Korkishko, "Design and Implementation of Efficient Block Cipher for Ubiquitous Computing Security", The 6th International Workshop on Information Security Applications -- WISA 2005, 2005.

[10] C. H. Lim, "A revised version of CRYPTON: CRYPTON v1.0", Fast Software Encryption, 6th International Workshop, FSE'99 Rome, Italy, March 24-26, 1999, Proceedings Series: Lecture Notes in Computer Science, Vol. 1636.

[11] G. Gaubatz, Jens-Peter Kaps and B. Sunar, "Public Key Cryptography in Sensor Networks-Revisited", Security in Ad-hoc and Sensor Networks, First European Workshop, ESAS 2004, Heidelberg, Germany, August 6, 2004, Revised Selected Papers Series: Lecture Notes in Computer Science, Vol. 3313.

Author History

Graduate School of Information Science and Electrical Engineering, Kyushu University
E-mail : saito@itslab.csce.kyushu-u.ac.jp

Faculty of Information Science and Electrical Engineering, Kyushu University
E-mail : sakurai@csce.kyushu-u.ac.jp