Privacy and Personal Information Protection in RFID Systems

Nohara, Yasunobu
Kyushu University

Baba, Kensuke
Kyushu University

Inoue, Sozo
Kyushu University

Yasuura, Hiroto
Kyushu University

https://hdl.handle.net/2324/10009

出版情報：RFID Handbook: Applications, Technology, Security, and Privacy, pp.485-497, 2008-03-17. CRC Press
バージョン：
権利関係：

Privacy and Personal Information Protection in RFID Systems

Yasunobu NOHARA, Kensuke BABA, Sozo INOUE, Hiroto YASUURA
Kyushu University

1 Introduction

In recent years, the more popular automatic identification systems using RFID have become, the more important privacy problems have become, with the phrase such as, "It is necessary to regard privacy for RFID's success".

At the same time, not many engineers, users, and RFID tag vendors, understand RFID privacy problem comprehensively, unfortunately. RFID is considered to penetrate wide application areas, some of which might be highly sensitive to users' privacy. Although applications in such privacy-sensitive areas have been avoided such as RFID tags are "killed" before handed to consumers with products, there are also predictions that RFID's power exists in the areas which involve consumers, that is, users.

We, being involved in RFID, must advance our knowledge and understanding of RFID privacy to manage troubles regarding users' privacy properly and quickly.

In this chapter, we try to clarify, as technical as possible, problems with respect to privacy and personal information protection around RFID systems.

2 Privacy and Personal Information Protection

At first, we would like to distinguish 'privacy protection' from 'personal information protection'. In the literature, information systems, including RFID systems, do not directly manage privacy protection. It is personal information protection what they manage. Although these are now confused worldwide, personal information protection can be considered as a matter of technology and operation for information systems with respect to personal data management, while privacy can be the 'right' not to let third parties invade his/her private area.

Here, an important issue is that personal information protection is not a sufficient condition, but just a necessary condition. That is, privacy invasion is accomplished after specific conditions are satisfied in addition to personal information leakage, such as the way of abusing the information, the customers' manner, and social factors such as legal environments. To wrongly skip the logic as 'using RFID is always invasion of privacy' is dangerous, since, in that case, RFID systems must manage human rights, which need sometimes unlimited cost.

Hereafter, we do not use a term of privacy, but use that of personal information protection, to clarify it is the RFID's risk.

3 Anonymity and Unlinkability

When we discuss personal information protection in RFID systems, the following two issues are considered as properties of an RFID system [1–3].

- **Anonymity:** the property to guarantee that any person using an RFID system is not identifiable from the data in the RFID system. For example, if an adversary can know the correspondence between the ID of a commodity and personal information of a user who brought the commodity, the user and the commodity do not have anonymity against the adversary.
- **Unlinkability:** the property to guarantee that any past record of behavior of a person using an RFID system is not traced from the data in the RFID system. For example, if an RFID tag of a commodity outputs the fixed value, the behavior of the person bringing the commodity can be traced by an adversary. Therefore,

the user and the commodity do not have unlinkability against the adversary.

If anonymity against an adversary is broken, the adversary can link past records of a person and the present record of the person by comparing the user's identity. Therefore, anonymity against the adversary is protected if unlinkability is protected against the adversary. On the other hand, anonymity is not always broken if unlinkability is broken. For example, consider the case that the ID of a user is stored in an RFID tag with a suitable encryption. Although the encrypted ID is fixed and linkable against the adversary, anonymity is protected since the adversary cannot know the original ID. Therefore, the achievement of unlinkability is more difficult than that of anonymity.

In a ubiquitous computing environment, a user can be identified by methods like "the person who is in front of me" even if we don't know the user's name. Therefore, it is also important to satisfy unlinkability for protecting anonymity from this viewpoint.

4 Personal Information Protection in RFID Systems

In this section, we introduce personal information protection schemes in RFID systems.

Ideas, operations and technologies for personal information protection of RFID system are basically the same as those of normal information systems. However, there are two unique features in RFID systems. The first feature is that an adversary can access an RFID tag easily without notice since RFID uses radio frequency. The second feature is that the restriction to the cost of the tag is very severe in RFID systems. Therefore, it is necessary to achieve personal information protection with lightweight operation in RFID systems. In this paper, we focus on personal information protection schemes that could solve this unique problem.

We classify the protection schemes into 1) Physical blocking approach, 2) Rewritable tag approach, and 3) Smart tag approach. Hereafter, we explain each approach.

4.1 Physical Blocking Approach

Physical blocking approach satisfies anonymity and unlinkability by preventing an adversary from accessing RFID tags physically.

The EPCglobal standard [4] specifies "Kill command", which disables functionality of the tag. Kill command is protected by PIN to prevent wanton deactivation of tags.

"Faraday cage" is an enclosure formed by conducting material, and blocks out radio frequency. While a user encloses RFID tags with a faraday cage, the tags don't work well because the cage prevents communications between tags and readers.

Juels *et al.* [5] propose "Blocker tag", which prevents an adversary from reading the ID of the tags which are near the blocker tag. The blocker tag is a cheap passive RFID device that can simulate many ordinary RFID tags simultaneously. Since the blocker tag pretends that all possible tags exist there, an adversary cannot identify the tags that are actually present there. A blocker tag can block selectively by simulating only selected subsets of ID codes, such as those by a particular manufacturer.

Karjoth *et al.* [6] propose "Clipped tags", in which a user can physically separate the chip from its antenna. In this system, user can deactivate the tag by removing its antenna. This separation provides visual confirmation that the tag has been deactivated.

A physical blocking approach has a problem that a user cannot use the RFID services because even regular service's reader cannot access the RFID tags.

4.2 Changing Output Approach

In this approach, an adversary can access an RFID tag, and read the output of the tag freely. However, the approach satisfies anonymity and unlinkability by changing the output of the RFID tag. To satisfy unlinkability, it is necessary to change the output of the tag frequently and prevent an adversary from discerning the relations between the outputs.

The changing output approach can be classified into 1) rewritable tag approach, and 2) smart tag approach.

4.2.1 Rewritable Tag Approach

In this approach, a non-volatile RAM (NVRAM), such as a flash memory is embedded within each RFID tag. The ID of the tag is stored in the NVRAM and the server can rewrite the ID.

Juels *et al.* [7] and Kinoshita *et al.* [8] propose "External re-encryption scheme" and "Anonymous-ID scheme". These schemes use a re-encryption scheme, which allows transforming a ciphertext C into a new unlinkable ciphertext C' using the public key only, without changing the plaintext. The tag outputs the encrypted ID which is stored in the NVRAM of the tag. The encrypted ID stored in the tag must be renewed because the tag outputs constant value until renewing the encrypted ID. The renewing process is as follows:

Step 1: The reader gets the encrypted ID from the tag.
Step 2: The reader re-encrypts the encrypted ID with the public key.
Step 3: The reader rewrites the old encrypted ID with the new encrypted ID.

The reading process is as follows:

Step 1: The reader gets the encrypted ID from the tag and sends it to the server.
Step 2: The server decrypts the encrypted ID using the private key, and obtains the ID of the tag.

Inoue *et al.* [2] propose "Private ID scheme", in which each tag has a ROM and a NVRAM. Permanent ID of the tag is stored in the ROM by a producer, and user can rewrite temporary ID stored in the NVRAM. The ROM and the NVRAM are used only exclusively. A user cannot read the permanent ID while the temporary ID is stored in the NVRAM. The user can read the permanent ID only when no value is stored in the NVRAM. Permanent ID is used for public uses like supply chain or recycling. And temporary ID is used for private uses.

In the rewritable tag approach, each RFID tag stores its ID in the NVRAM, and server updates these IDs periodically. Since the tag doesn't need cryptographic function, the cost of RFID tag is low. However, running cost of the system is high because the server has to update tag's ID periodically. And since the tag outputs constant value until next update, unlinkability against an adversary is limited.

4.2.2 Smart Tag Approach

In this approach, a cryptographic function and a ROM are embedded within each RFID tag. An RFID tag changes its output every time using a cryptographic function –public key encryption, common key encryption and hash function– on itself.

Let N be the number of RFID tags in an RFID system where the ID id_i of an RFID tag T_i is a string of length L over a finite alphabet Σ for $1 \leq i \leq N$. We assume that if $i \neq j$, then $id_i \neq id_j$ for $1 \leq i, j \leq N$, and $2^L \gg N$. For $s, t \in \Sigma^*$, we denote by $s \| t$ the concatenation of s and t.

■Public Key Encryption Kinoshita *et al.* [9] propose "Internal re-encryption scheme", which uses a public key encryption. In this scheme, a public key encryption function and a NVRAM are embedded within each RFID tag. The encrypted ID stored in the NVRAM is re-encrypted by the public key encryption function on the RFID tag. Since the tag changes its output every time, this scheme provides good personal information protection. However, there is a problem that the tag is expensive because a public key encryption function is complex and costly.

■Common Key Encryption Kinoshita *et al.* [9] propose "Common key encryption scheme", which uses a common key encryption. In this scheme, a common key encryption function, a ROM, and a pseudo-random number generator are embedded within each RFID tag. The server identifies the tag through the following protocol.

Step 1: RFID tag T_i generates a random number R, and sends $X = E_K(id_i \| R)$ to the server.
Step 2: The server decrypts X using the common key K and gets id_i.

The calculation of the common key encryption is smaller than that of the public key encryption; however, it is vulnerable to tampering because the common key must be shared among all tags. The reason why the common key must be shared is as follows. If each tag uses an individual key, the server must know which key to use for decrypting of the encrypted ID. However, how does the server determine the tag's ID before decrypting of the encrypted ID? Therefore, it is difficult to use individual common keys.

An exhaustive search of the key can solve this individual key problem; however the calculation load of the server is high. We describe the detail of the exhaustive search in Section 5 because a hash-based scheme also uses

Table. 1 Personal Information Protection Schemes for RFID systems

	Physical blocking	Rewritable tag	Smart tag		
			Public key	Common key	Hash
Service	**Not Available**	Available	Available	Available	Available
Anonymity	Satisfied	Satisfied	Satisfied	Satisfied	Satisfied
Unlinkability	Satisfied	**Partly satisfied**	Satisfied	Satisfied	Satisfied
Vulnerability	Tamper free	Tamper free	Tamper free	**Vulnerable**	Tamper free
Calc. on Server	Small	Small	Small	Small	**Large**
Cost of tag	Low	Low	**High**	Low	Low

the exhaustive search.

■Hash Function Hash-based schemes [3, 10–16] use a hash function as a cryptographic function. Since the hash calculation is a lightweight operation, the hash-based schemes are suitable for RFID systems, where the implementation cost of an RFID tag must be low. However, the calculation load of the server is high because the server needs to do exhaustive search. We describe the detail of the hash-based schemes in Section 5.

4.3 Comparison

Table 1 compares the personal information protection schemes for RFID systems. A bold font denotes the weakness of the scheme.

5 Hash-Based Scheme

In this section, we describe hash-based schemes and compare these schemes.

A hash-based scheme is one of schemes using the smart tag approach. A tag changes its output using hash function which is embedded on the tag. Since the hash function is a lightweight operation, a hash-based scheme is suitable for RFID systems, where the implementation cost of an RFID tag must be low.

We assume the hash function has 'one-way' and 'pseudo-random' properties. 'One-way' means it is computationally infeasible to calculate the input of the hash function from the output of the hash function. 'Pseudo-random' means the output of the hash function is computationally indistinguishable from a true random number.

5.1 Randomized Hash Lock Scheme [3]

In this scheme, a hash function H, a ROM, and a pseudo-random number generator are embedded within each RFID tag.

RFID tag T_i stores id_i in the ROM. The server stores the IDs id_i $(1 \leq i \leq N)$ of all tags. The server identifies the tag through the following protocol (see Figure 1).

Step 1: RFID tag T_i generates a random number R, and sends $X = H(id_i \| R)$ and R to the server.
Step 2: The server finds id_i that corresponds to X by checking $X = H(id_i \| R)$ for $1 \leq i \leq N$.

Since R changes every time, $X = H(id_i \| R)$ is not fixed. It is computationally infeasible to get id_i from X and R due to the one-way property of the hash function. Therefore, this scheme provides unlinkability against an adversary.

5.2 Hash-chain Scheme [10, 11]

In this scheme, two different hash functions H and G, a ROM, and a NVRAM are embedded within each RFID tag.

RFID tag T_i stores id_i in the ROM, and stores secret information $cs_i^1 \in \Sigma^{L'}$ in the NVRAM. The server stores the pair (id_i, cs_i^1) $(1 \leq i \leq N)$ of all tags. The server identifies the tag through the following protocol (see

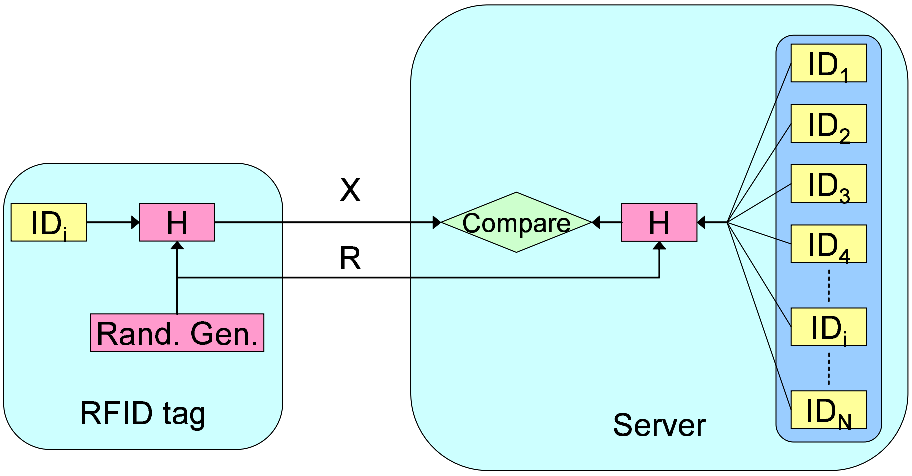

Fig. 1 Randomized Hash Lock Scheme

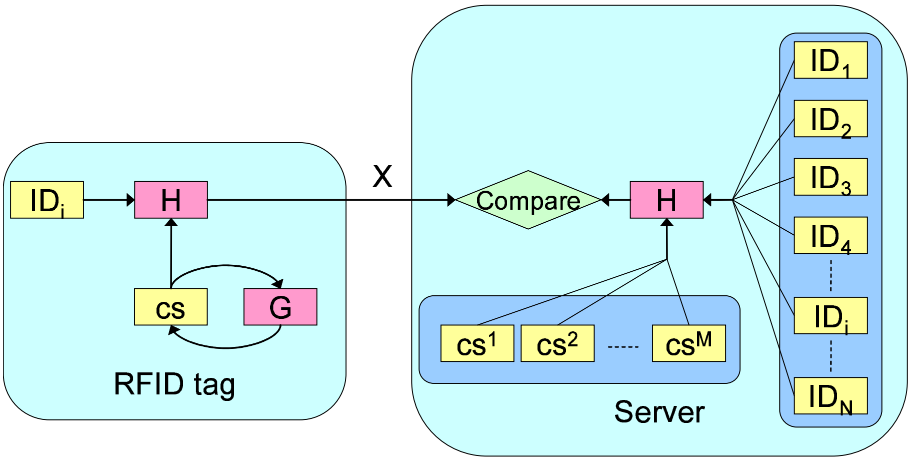

Fig. 2 Hash-chain Scheme

Figure 2).

Step 1: RFID tag T_i sends $X = H(id_i \| cs_i^l)$ to the server. RFID tag T_i updates $cs_i^{l+1} \leftarrow G(cs_i^l)$.
Step 2: The server finds the id_i corresponding to X by checking $X = H(id_i \| cs_i^l)$ for all $1 \le i \le N$ and all $1 \le l \le M$ (where M is the maximum length of the hash chain).

Since cs_i^l changes every time, $X = H(id_i \| cs_i^l)$ is not fixed. It is computationally infeasible to get id_i from X due to the one-way property of the hash function. Therefore, this scheme provides unlinkability against an adversary.

Moreover, it is computationally infeasible to get $cs_i^{l'} (l' < l)$ even if id_i and cs_i^l are tampered with. Therefore, the scheme provides forward security, meaning that no RFID tag can be traced from past ID information even if the secret information in the tag is tampered with.

5.3 K-steps ID Matching Scheme [12], Tree-based Scheme [13]

In these schemes, a hash function H, a ROM, and a pseudo-random number generator are embedded within each RFID tag. These schemes use a tree ID structure. In this paper, we explain the K-steps ID Matching Scheme [12].

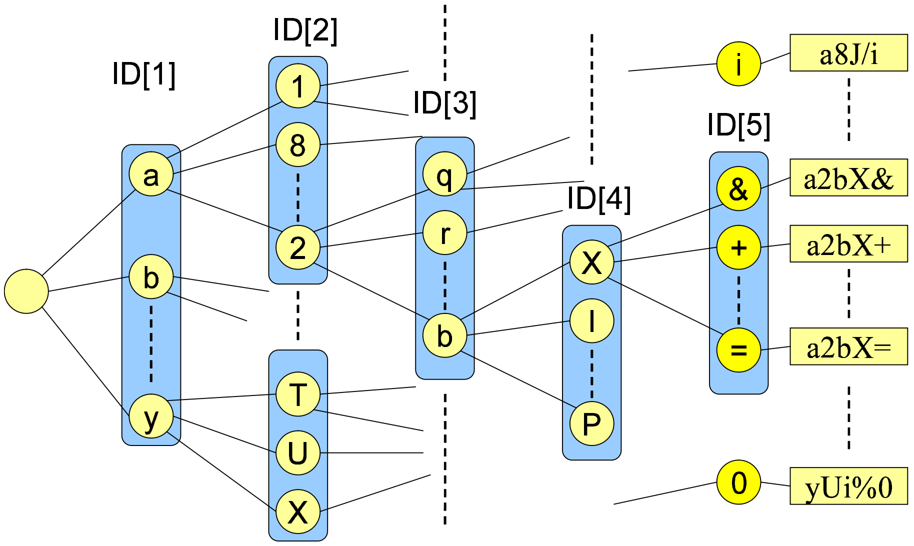

Fig. 3 An ID structure for the K-steps ID matching scheme

5.3.1 ID Configuration

We use a labeled tree of depth K, such as the tree shown in Figure 3. The tree has N leaves, and each leaf corresponds to an RFID tag. Each node has a unique label. ID id_i of an RFID tag corresponding to a leaf node is defined as the sequence of labels from the root node to the leaf node (e.g., a2bX& in Figure 3).

In the following, the k-th $(1 \leq k \leq S_i)$ label of T_i is denoted by $id_i[k]$, where S_i is the depth of leaf i, and $1 \leq S_i \leq K$.

5.3.2 Protocol

In the K-steps ID matching scheme, the server recognizes an ID from the output of an RFID tag through the following protocol.

Step 1: RFID tag T_i generates a random number R. T_i then sends $(R, X_1, X_2, \ldots, X_K)$ to the server, where X_k is $H(id_i[k]\|R)$ if $1 \leq k \leq S_i$ and a random number R_k if $S_i + 1 \leq k \leq K$.

Step 2: The server operates as follows:

STEP2-1: let Z be the root of the labeled tree and let $k \leftarrow 1$;

STEP2-2: find L_i s.t. $H(L_i\|R) = X_k$ by computing $H(L_i\|R)$ for each child L_i of Z, and update $Z \leftarrow L_i$;

STEP2-3: output the label corresponding to Z as the ID of the RFID tag if Z is a leaf; otherwise, let $k \leftarrow k+1$ and return to STEP2-2.

In Step 1, RFID tag T_i sends a random number as X_k for $S_i + 1 \leq k \leq K$, which hides the depth of the leaf S_i to prevent weakening the unlinkability against an adversary.

When $K = 1$, the protocol and the ID structure of the protocol correspond to those of the randomized hash lock scheme [3]. If some procedures of the protocol are changed, it becomes a protocol corresponding to the hash-chain scheme [10, 11].

5.4 Avoine's scheme [14, 15]

Avoine *et al.* [14, 15] developed a specific time-memory trade-off that reduces the amount of computation in the hash-chain scheme [10, 11]. This time-memory trade-off reduces the hash calculations on the server with help of pre-computation results. However, heavy pre-calculation is needed with Avoine's scheme [14, 15].

Table. 2 Classification of Hash-Based Schemes

	Base Model	ID Struct.	Time-memory
Hash Lock [3]	Hash Lock	Normal	No
K-step [12], Tree-based [13]	Hash Lock	Tree	No
Hash-chain [10, 11]	Hash Chain	Normal	No
Avoine's scheme [14, 15]	Hash Chain	Normal	Yes
Yeo's without pre-comp. [16]	Hash Chain	Tree(K=2)	No
Yeo's with pre-comp. [16]	Hash Chain	Tree(K=2)	Yes

5.5 Yeo's scheme [16]

Yeo's scheme [16] is one of the hash-chain schemes, and the same ID structure as in K-steps ID matching scheme is used to reduce the server complexity. Yeo *et al.* propose two types of scheme. One is a scheme without pre-computation which uses only a grouping technique. The other is a scheme with pre-computation which uses both a grouping technique and a time-memory trade-off technique [14].

5.6 Comparison

We compare the hash-based schemes from the viewpoints of security, hash calculation time, the amount of memory needed, and the amount of communication.

5.6.1 Classification of Hash-Based Schemes

For our comparison, we classify hash-based schemes with regard to three characteristics:

- Base model (hash lock or hash chain)
- ID structure (normal or tree)
- Introduction of a time-memory trade-off technique [14] (yes or no)

Table 2 shows the classification results.

There are no proposed schemes with the combination (Hash Lock, Normal, Yes) or (Hash Lock, Tree, Yes) because the responses of RFID tags in a hash lock scheme are randomized, which means a large memory space is needed to apply a time-memory trade-off technique [15].

5.6.2 Security

We compare the security of the hash-based schemes with respect to three concerns:

- Unlinkability
- Forward security
- Prevention of replay attacks

■Unlinkability We analyzed the unlinkability of the hash-based schemes by the *degree of unlinkability* [17]. The degree of unlinkability ranges from 0 to $\log_2 N$ [bit], and unlinkability becomes stronger as the degree of unlinkability increases. When an adversary has no ID information, each degree of unlinkability for the hash-based schemes is $\log_2 N$.

Since anonymity and unlinkability are closely related, this evaluation of unlinkability is also related to that of anonymity.

When an adversary obtains one ID, such as by tampering with an RFID tag, the degree of unlinkability of each scheme differs depending on its ID structure. The degree of unlinkability for the normal ID structure and that for the tree ID structure are given as follows [17].

$$U_{normal} = \frac{N-1}{N} \log_2 (N-1) \tag{1}$$

$$U_{tree} = \log_2 N + \frac{N-1}{N}\{\log_2 (N^{\frac{1}{K}} - 1) - \frac{N^{\frac{1}{K}}}{K(N^{\frac{1}{K}} - 1)} \log_2 N\} \tag{2}$$

The normal ID structure schemes enable user unlinkability, except for the tampered user, but the tree ID structure schemes cannot enable user unlinkability since some users share part of the ID of the tampered user. From Eqs. (1) and (2), we can see that the degree of unlinkability with the tree ID structure is lower than that with the normal structure.

However, the tree ID structure schemes provide the same level of unlinkability as the normal ID structure if $\alpha = N^{\frac{1}{K}}$ is large enough. Since the optimized K is much less than 10 even if N becomes 2^{100} [12], the tree ID structure decreases the degree of unlinkability only slightly.

Thus, the decrease in the degree of unlinkability with the K-steps ID matching scheme is only small [17] compared to that with the normal ID structure.

■Forward security *Forward security* is a property that means no RFID tag can be traced from past ID information even if an adversary tampers with the secret information in the tag.

Hash lock schemes, including K-step ID matching scheme, cannot provide forward-security because an adversary can easily get a random number R. On the other hand, hash-chain schemes can provide forward security since it is computationally difficult for an adversary to get $cs_i^{l'}$ $(l' < l)$ even if he has tampered with id_i and cs_i^l.

However, Juels *et al.* pointed out that hash-chain schemes create a security risk in that an adversary can guess a tag's count number [18]. We discuss this problem in Section 5.6.3.

■Prevention of Replay Attacks A replay attack is one in which a valid data transmission is maliciously or fraudulently repeated. The attack is carried out by an adversary who masquerades as a legitimate user.

Replay attacks must be prevented when a server has to authenticate as well as identify an RFID tag. One way to do this is to use a fresh challenge by the server. Hash lock schemes can prevent replay attacks if a step is added where the server sends a fresh challenge to the tag and includes the challenge in the hash calculations. In K-steps ID matching scheme, the protocol to prevent replay attacks is as follows.

```
Step 1: The server generates a random number $R_s$, and then sends $R_s$ to RFID tag $T_i$.
Step 2: RFID tag $T_i$ generates a random number $R_d$, and then sends $(R_d, X_1, X_2, \ldots, X_K)$ to the server, where
    $X_k$ is $H(id_i[k] \| R_s \| R_d)$ if $1 \le k \le S_i$, and a random number $R_k$ if $S_i + 1 \le k \le K$.
Step 3: The server operates as follows:
    STEP3-1: let $Z$ be the root of the labeled tree and let $k \leftarrow 1$;
    STEP3-2: find $L_i$ s.t. $H(L_i \| R_s \| R_d) = X_k$ by computing $H(L_i \| R_s \| R_d)$ for each child $L_i$ of $Z$, and update
        $Z \leftarrow L_i$;
    STEP3-3: output the label corresponding to $Z$ as the ID of the RFID tag if $Z$ is a leaf; otherwise, let $k \leftarrow k+1$
        and return to STEP 3-2.
```

Avoine *et al.* propose a modified hash-chain scheme which prevents replay attacks using a challenge [15]. This technique can be easily adopted in Yeo's scheme without pre-computation.

However, the technique of using a fresh challenge cannot be applied directly to Avoine's scheme or Yeo's scheme with pre-computation since the randomization of the tag's response prevents the server using a time-memory trade-off (see Section 5.6.1). Therefore, the RFID tag must calculate a hash value(s) without a challenge and a hash value with the challenge [15]. The former value(s) enable(s) the server to identify the tag, while the latter one prevents replay attacks.

All of the hash-based schemes proposed so far have a countermeasure against replay attacks, and preventing replay attacks increases both the calculation complexity and the communication amount. We discuss this problem in Sections 5.6.3 and 5.6.4.

Table. 3 Comparison of required memory and time

	Hash Calc. on Device	Hash Calc. on Server	Pre-comp. on Server	Memory on Server
Hash Lock	1	N	0	0
K-step, Tree-based	K	$KN^{\frac{1}{K}}$	0	0
Hash-chain	2	MN	0	N
Avoine's scheme	2 [+1]	$\frac{3^3}{2^3}\frac{M^3\gamma}{c^3\mu^2}$ [+1]	$\frac{NM^2}{2}$	cN
Yeo's without pre-comp.	4	$2M\sqrt{N}$	0	N
Yeo's with pre-comp.	4 [+1]	$\left(\frac{2^5M^6\gamma}{c^3\mu^2}\right)^{\frac{1}{4}}$ [+1]	$\left(\frac{2^3c^3N^4\mu^2}{3^4M^2\gamma}\right)^{\frac{1}{4}}\frac{M^2}{2}$	cN

Table. 4 Communication cost

	not preventing replay attacks	preventing replay attacks
Hash Lock	$r+h$	$2r+h$
K-step, Tree-based	$r+Kh$	$2r+Kh$
Hash-chain	h	$r+h$
Avoine's scheme	h	$r+2h$
Yeo's without pre-comp.	$2h$	$r+2h$
Yeo's with pre-comp.	$2h$	$r+3h$

5.6.3 Comparison of memory and time

We compare the different schemes regarding the number of hash calculations on the RFID tag and on the server, the number of pre-computations on the server, and the memory required for the pre-computation results.

Table 3 compares the memory and the time needed for each scheme. In the table, M is the maximum length of the hash-chain, μ is the conversion factor, c is the memory size parameter for Avoine's scheme [14], and γ is the rate of successful search parameter in that scheme. For example, the success rate is 99.9% when $\gamma = 8$.

In the table, ‘Memory on server’ denotes the amount of secret information to be stored cs_i^l in hash-chain schemes. Note that the memory amount does not include the space for the ID list, which is required for every scheme. The additional number of calculations for the scheme to prevent replay attacks is given in brackets.

As the table shows, the number of hash calculations on the server in a time-memory trade-off scheme includes M^3 or $M^{1.5}$, while that in the K-step ID matching scheme includes N. Therefore, K-steps ID matching scheme might be disadvantageous in terms of the required time if M^3 or $M^{1.5}$ is sufficiently smaller than $N^{\frac{1}{K}}$.

The server cannot identify the RFID tag when the tag number is larger than M because it will be outside of the search range. In addition, there is a security risk in that an adversary can guess a tag's count number if M is small [18]. Therefore, M must be sufficiently large.

Avoine *et al.* pointed out that replacing cs_i^1 by cs_i^k in the database regularly expands the search range of the server [14]. However, the problem of a count number leakage remains, and heavy pre-computation (e.g., $M^2N/2$) is needed for every replacement.

5.6.4 Communication cost

Table 4 compares the communication cost for each scheme in terms of the amount of communication data. The costs are shown in each case of preventing or not preventing replay attacks. In the table, r is the length of the random value for the challenge, and h is the length of the hash output.

In tree ID structures, including those of the K-step ID matching scheme, the communication cost increases in proportion to the tree depth. For the K-step ID matching scheme, we measured the practical time for the entire execution (including the communication time between the server and the RFID tag), and found it is shorter than that of a naive scheme [3] when N is sufficiently large.

Thus, we expect the communication cost of K-steps ID matching scheme to be negligible in a practical situation. However, further evaluation is required since we used contact smart cards in our experiment. With contact-less smart cards or RFID tags, the communication cost might increase because of communication failures.

6 Conclusion

In this paper, we explained personal information protection in RFID systems. Firstly, we explained a distinction between privacy protection and personal information protection, and introduced two properties –anonymity and unlinkability– for personal information protection. Secondly, we surveyed the personal information protection schemes which realized anonymity and unlinkability. Finally, we described the detail of the hash-based schemes, one of the smart tag approaches.

References

[1] *ISO/IEC 15408 - INTERNATIONAL STANDARD Information technology - Security techniques - Evaluation criteria for IT security - Part2: Security functional requirements*, 1999.

[2] Sozo Inoue and Hiroto Yasuura. RFID privacy using user-controllable uniqueness. In *RFID Privacy Workshop@MIT*, Nov. 2003.

[3] Stephan A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *1st International Conference on Security in Pervasive Computing – SPC2003*, volume 2802 of *LNCS*, pages 201–212. Springer, 2004.

[4] EPCglobal. http://www.epcglobalinc.org/.

[5] Ari Juels, Ronald L Rivest, and Michael Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In *10th ACM Conference on Computer and Communications Security – CCS2003*, pages 103–111. ACM Press, Oct. 2003.

[6] Günter Karjoth and Paul A. Moskowitz. Disabling RFID tags with visible confirmation: Clipped tags are silenced. In *Proc. of 2005 ACM Workshop on Privacy in the*, pages 27–30, 2005.

[7] Ari Juels and Ravikanth Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *Financial Cryptography – FC2003*, volume 2742 of *LNCS*, pages 103–121, Jan 2003.

[8] Shingo Kinoshita, Fumitaka Hoshino, Tomoyuki Komuro, Akiko Fujimura, and Miyako Ohkubo. Low-cost RFID privacy protection scheme. *IPSJ Journal*, 45(8):2007–2021, 2004. in Japanese.

[9] Shingo Kinoshita, Miyako Ohkubo, Fumitaka Hoshino, Gembu Morohashi, Osamu Shionoiri, and Atsushi Kanai. Privacy enhanced active rfid tag. In *Proc. of 1st International Workshop on Exploiting Context Histories in Smart Environments – ECHSE2005*, 2005.

[10] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to a privacy friendly tag. In *RFID Privacy Workshop@MIT*, Nov. 2003.

[11] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Hash-chain based forward-secure privacy protection scheme for low-cost RFID. In *2004 Symposium on Cryptography and Information Security – SCIS2004*, volume 1, pages 719–724, Jan. 2004.

[12] Yasunobu Nohara, Toru Nakamura, Kensuke Baba, Sozo Inoue, and Hiroto Yasuura. Unlinkable identification for large-scale RFID systems. *IPSJ Journal*, 47(8):2362–2370, Aug. 2006. Online version : IPSJ Digital Courier, Vol. 2, pp.489–497.

[13] David Molnar and David Wagner. Privacy and security in library : RFID issues, practices, and architectures. In *11th ACM Conference on Computer and Communications Security – CCS2004*, pages 210–219. ACM Press, Nov. 2004.

[14] Gildas Avoine and Philippe Oechslin. A scalable and provably secure hash-based RFID protocol. In *2nd International Workshop on Pervasive Computing and Communications Security – PerSec2005*, pages 110–114. IEEE Computer Society Press, Mar. 2005.

[15] Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing time complexity in RFID systems. In *12th Annual Workshop on Selected Areas in Cryptography – SAC2005*, volume 3897 of *LNCS*, pages 291–306. Springer, 2005.

[16] Sang-Soo Yeo and Sung Kwon Kim. Scalable and flexible privacy protection scheme for RFID systems. In

2nd European Workshop on Security in Ad-Hoc and Sensor Networks – ESAS2005, volume 3813 of *LNCS*, pages 153–163. Springer, 2005.

[17] Yasunobu Nohara, Sozo Inoue, Kensuke Baba, and Hiroto Yasuura. Quantitative evaluation of unlinkable ID matching schemes. In *Proc. of 2005 ACM Workshop on Privacy in the Electronic Society – WPES2005*, pages 55–60. 2005 ACM Workshop on Privacy in the Electronic Society – WPES2005, ACM Press, Nov. 2005.

[18] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *IACR Cryptology ePrint Archive Report*, number 2006-137, 2006.