Unlinkability and Real World Constraints in RFID Systems

Nohara, Yasunobu
Kyushu University

Inoue, Sozo
Kyushu University

Yasuura, Hiroto
Kyushu University

出版情報：Proc. of 5th IEEE International Conference on Pervasive Computing and Communications, pp.371-376, 2007-03. IEEE International Conference on Pervasive Computing and Communications
バージョン：
権利関係：

Unlinkability and Real World Constraints in RFID Systems

Yasunobu NOHARA, Sozo INOUE, Hiroto YASUURA
Kyushu University
744 Motooka Nishi-ku Fukuoka 819-0395 JAPAN
{nohara,sozo,yasuura}@c.csce.kyushu-u.ac.jp

Abstract

Unlinkability, the property that prevents an adversary recognizing whether outputs are from the same user, is an important concept in RFID. There are many proposed schemes that provide unlinkability, however most of the schemes don't consider constraints of the real world. In this paper, we discuss the unlinkability and the real world constraints in RFID systems, and simulate the attack by an adversary. The simulation results show the real world constraints have possibility that break the unlinkability. We also analyze the simulation results from three viewpoints.

1. Introduction

RFID (Radio Frequency IDentification) systems are systems which, with wireless communication, identify small IC chips (RFID tags) attached to people and/or objects in the real world. RFID Systems enable information systems to relate people or objects in the real world with databases in the virtual world, and are therefore acknowledged as fundamental systems for pervasive computing.

Since RFID Systems utilize wireless communication, an adversary can easily read the IDs of primitive RFID tags. Therefore, there appears a problem that an adversary can track users' movements, by setting readers everywhere and extracting the log for an ID. For the problem, *unlinkability*, a concept that one can not decide entries in the log as from a single user or not, and its implementations [1, 2, 3, 4] have been proposed.

However, no matter if we use the implementations above, vulnerability of unlinkability still remains if an adversary can utilize the constraints in the real world [5, 6, 7]. For example, if an adversary knows the location of the readers, then he/she can infer the correlation between tuples of each (time, location), such as "*I don't know the ID of you, but I know you were in the room a minute ago, since everyone on this floor is over a minute from the room.*".

In this paper, we discuss the unlinkability and the real world constraints in RFID systems. Firstly, we define a link expression with real world constraints and propose a location tracking model. Secondly, we define the link problem and propose its solution. Finally, we simulate a link successful rate of the link problem using our laboratory as simulation environment. The simulation results show the real world constraints have possibility that break the unlinkability. We also analyze the simulation results from three viewpoints.

The remainder of this paper is organized as follows. Section 2 describes previous work and defines a link with real world constraints. Section 3 proposes a location tracking model and defines location tracking problems. Section 4 discusses the link problem, one of the location tracking problems and proposes its solution. Section 5 describes simulation results of the link problem. Section 6 discusses the link attack with real world constraints based on the simulation results. Section 7 concludes this paper.

2. Link with Real World Constraints

In this section, we describe previous work and define a link expression with real world constraints.

2.1. Link Attack

For simplicity, we assume that every RFID tag belongs to and follows a user. In reality, an RFID tag might belong to no/multiple users, or placed to an independent place of the user, but we ignore these cases, where the problem is relaxed as for unlinkability.

A *link attack* (or simply, *link*) is an attack by an adversary to know the movement of a (sometimes unknown) user, performed as follows:

Phase 1: An adversary installs readers on any locations known to him/her,

Phase 2: obtains the output of an RFID tag when it is around a reader, and records the tuple: (output, time, location), and,

Phase 3: retrieves a set of tuples: (time, location) for a single RFID tag inferring from the record, which corresponds to a movement history of a user.

Supposing RFID tags always output static IDs, an adversary can perform successful link, since **Phase 3** is possible by retrieving a set of tuples: (time, location) for a single 'output'.

Against link attacks, methods to have dynamic IDs on each RFID tag, using such as hash functions, are proposed [1, 2, 3, 4]. These are assumed to achieve unlinkability against adversaries, since they cannot perform **Phase 3** in which each 'output' cannot be used as a retrieval key for the same RFID tag.

2.2. Real World Constraints

Although the link attack introduced above is upon the outputs of RFID tags, the information an adversary can use is not limited to them in reality. In general, the time and the location of a read event of a reader have the following correlations [5, 6, 7]:

- An RFID tag can only move to some length in limited time, and not be in far away.
- An RFID tag sometimes has to go through some place to get to a goal. For example, a product in a supermarket has to pass a cashier before exits.

Utilizing these correlations, vulnerability of unlinkability still exists as the following [7]: Suppose a system with two RFID tags, and suppose reader A and B read a tag in time t, and reader C and D read in time $t + k$, where A and C is far enough not to move in time k. Then, one can infer that a tag has moved from B to C.

2.3. Probabilistic Expression of Link

The link attacks in Section 2.1 only succeed in either (1) always successful as static IDs, or (2) always unsuccessful as difficult as random attacks.

However, in case of utilizing real world constraints where the success in link attack comprehends uncertainty, we should introduce probabilistic expression of link. For instance, we should express such as an RFID tag observed in time t at place A will be at place B in the possibility $p^k(\mathrm{A,B}) = 0.3$ and at C in $p^k(\mathrm{A,C}) = 0.1$, at time $t + k$.

With the expression, we can represent various constraints in the real world, such that the possibility $p^k(\mathrm{A,B}) = 0$ of link means a tag cannot move from a place A to B in time k.

Table 1. Link Attack without/with Real World Constraint

	without	with
Key for Link	Output of RFID Tag	Plus, Location, Time
Expression of Link	Success or Failure	Probability (0–1)

Table 1 is a comparison of link attacks in traditional methods and those with real world constraints. With real world constraints, link attacks are done with not only the output of RFID tags, but also with location and time, and the expression of link is probabilistic.

3. Location Tracking Model

In this section, we propose a location tracking model to evaluate the location tracking ability of an adversary. Since the link attack is one of the methods of the location tracking, we can evaluate the attack by this model.

3.1. Target System for Modeling

We assume users have an RFID tag and an adversary try to know movements of users by RFID.

Let N be the number of RFID tags in an RFID system. L denotes a set of locations that users can move. L_R denotes a subset of L where the adversary installed a reader. We assume $|L| = m$ and $|L_r| = r$.

When an RFID tag moves into a location $l_i \in L$, the reader either reads the output of the RFID tag correctly, or cannot detect tag due to communication errors. We denote the probability of former action by $1-P_{err}$, and that of later action is P_{err}.

We assume that there are no false positive reads, where RFID tags might be read when they are outside of the reader range.

3.2. Location Tracking Model

Figure 1 shows a *location tracking model* that we proposed. Our model consists of N-tuples (RFID tag, location information source), a group of readers, and a decoder.

In the following, we explain each element.

3.2.1 RFID Tag and Location Information Source

Here, *Location Information Source* L_i is an information source that represents the location of RFID tag M_i. O_i de-

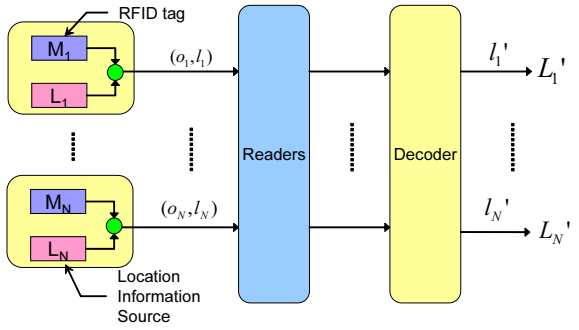

Figure 1. Location Tracking Model

notes a set of outputs of RFID tag M_i.

We express an existence of an RFID tag M_i by a location $l \in L_i$ as a pair (o, l), where $o \in O_i$.

3.2.2 Readers

Readers are modeled as an algorithm that outputs N-tuples $(O_i \cup \{*\}, L_i \cup \{*\})$ from N-tuples input (O_i, L_i). '$*$' denotes unknown elements of O_i or L_i. The outputs of this algorithm correspond to tuples (output, location, time), that is stored in the database in **Phase 2**. This algorithm is described as follows:

Step 1: let $i \leftarrow 1$.

Step 2: let o be an input from M_i and l be a input of L_i.

Step 3: if $l \notin L_R$ then let $Z_i \leftarrow (*, *)$ and go to Step 6.

Step 4: let $x \leftarrow GenRandReal(0, 1)$.

Step 5: if $0 \leq x < P_{err}$ then let $Z_i \leftarrow (*, *)$; otherwise let $Z_i \leftarrow (o, l)$.

Step 6: if $i \neq N$ then let $i \leftarrow i + 1$ and return to Step 2.

Step 7: let $i \leftarrow 1$.

Step 8: let $j \leftarrow GenRandInt(i + 1, N)$.

Step 9: swap with Z_i and Z_j.

Step 10: if $i = N - 1$ then output $Z_1, \cdots, Z_N$; otherwise let $i \leftarrow i + 1$ and return to Step 8.

$GenRandReal(x, y)$ denotes a random number generator that outputs a real number $r \in [x, y]$, and $GenRandInt(i, j)$ denotes a random number generator that outputs an integer $m \in [i, j]$.

When an RFID tag is in a location where no reader is installed, the adversary cannot get the output of the tag and its location. This is represented in Step 2 and Step 3.

When an RFID tag is in a location where a reader is installed, the adversary can get the output of the tag and its location if read error doesn't occur. This is represented in Step 4, Step 5 and Step 6.

From Step 7 to Step 10 shuffle the data not to cause the correlation of location information L_i.

3.2.3 Decoder

A decoder is modeled as a finite state machine which outputs N-tuples location information L_i from $(O_i \cup \{*\}, L_i \cup \{*\})$ inputs. An adversary tries to close the outputs of the decoder to the output of location information source, by data correction. The approximation level of the outputs shows the tracking ability of the adversary.

3.3. Location Tracking Problem

We can define some problems related to location tracking by location tracking model.

1. How to construct a decoder to maximize the tracking ability
2. How to allocate readers to maximize the tracking ability under some restrictions

In next section, we discuss former problem that is restricted condition $P_{err} = 0$.

4. Link Problem

In this section, we define the link problem, one of the location tracking problems, and proposes its solution.

4.1. Problem Definition

Let S be a set of the locations of N tags at time t, and E be a set of the locations at time $t+k$. We suppose $P_{err} = 0$. The *link problem* is finding a correct combination of the locations which each person moved. An adversary can get people's moving history by solving a series of link problem changing time interval k.

Let F be a set of bijection $S \rightarrow E$. We use a bijection $f \in F$ to express combinations of the locations. Let strength of the link between location i at time t and location j at time $t+k$ be $p^k(i, j)$. According to Bayes' theorem, we can compute the probability that $f \in F$ is a correct combination, as such:

$$\frac{\prod_{s \in S} p^k(s, f(s))}{\sum_{g \in F} \{\prod_{s \in S} p^k(s, g(s))\}} \quad (1)$$

An adversary use f, which maximizes Eq.(1), to construct a decoder. Since denominator of Eq.(1) is a constant whatever $f \in F$, we can define the link problem as follows.

Link Problem

Given two sets, S and E, of equal size n, together with a weight function $P : S \times E \rightarrow R^+$. Find a bijection $f : S \rightarrow E$ such that the cost function:

$$C(f) = \prod_{s \in S} P(s, f(s)) \quad (2)$$

is maximized.

4.2. Solution

The link problem can be solved by brute force method. However, $O(n!)$ calculation is needed.

If Eq.(2) is converted by logarithm, we have

$$\log\{C(f)\} = \sum_{s \in S} \log\{P(s, f(s))\} \quad (3)$$

Eq.(3) is a same form of assignment problem, which can be solved $O(n^3)$ calculation by Munkres algorithm [8].

$\hat{f} \in F$, which maximizes Eq.(3), also maximizes Eq.(2). Since $x < y \iff \log(x) < \log(y)$, where $\forall x, y \in R^+$.

Therefore, the link problem can be solved $O(N^3)$ calculations by Munkres algorithm.

5. Simulation

Using the formula obtained in the previous section, we simulated the link successful rate without ID information.

We used our laboratory as simulation environment. Figure 2 shows the rough sketch of our laboratory. We suppose $m = r = 9$. State transition matrix $\Pi = \{p_{i,j}\}$ is given as follows.

$$\Pi = \begin{bmatrix} 0.1 & 0.2 & 0.7 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.7 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.2 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.0 & 0.0 & 0.1 & 0.6 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.3 & 0.6 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.1 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.2 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.1 & 0.0 & 0.6 & 0.2 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.6 \end{bmatrix}$$

State transition matrix Π includes real world constraints as follows. Firstly, when people in a room(L2, L4, L5, L6, L8, L9), the probability of staying in the same room is high(e.g. $p_{4,4} = 0.6$). Secondary, when people in a corridor(L1, L3, L7), the probability of moving to another place is high(e.g. $p_{1,3} = 0.7$). Finally, people cannot move from a room to another room directly without passing a corridor(e.g. $p_{5,6} = 0$).

We simulated the link successful rate varying N and k by a simulation program. Figure 3 shows the simulation

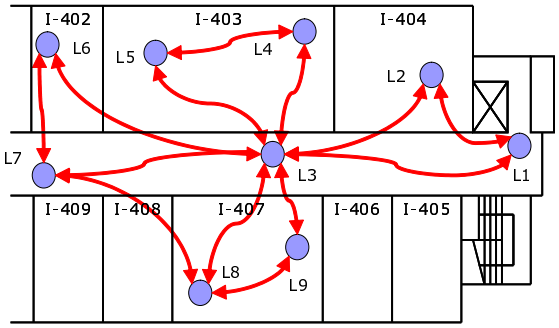

Figure 2. Rough sketch of our laboratory and readers allocation

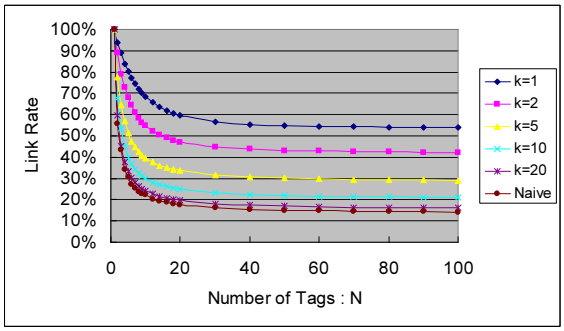

Figure 3. Simulation results for link successful rate

results. Naive scheme uses a decoder that doesn't consider real world constraints. The more the number of tags N and the time interval k increase, the more the link successful rates decreases. If the number of tag N exceeds about 40, the link successful rate is converged. When k is 20, the link successful rate of our scheme is a same level as that of naive scheme.

When the conditions are $N = 20$, $k = 1$, the link successful rate of the naive scheme is 17.5%. However, that of our scheme is 59.6% in the same conditions. This means the real world constraints have possibility that break the unlinkability.

6. Discussion

In this section, we discuss a link attack considering real world constraints from the viewpoints of prevention, acceptance, and application.

6.1. Prevention of Link Attack

In this subsection, we discuss how to prevent a link attack by an adversary.

6.1.1 No Response to Adversary's Reader

The best way to avoid the link attack is preventing an adversary from getting any information about a tag. Yamada *et al.* proposed a scheme that tags don't response to adversary's readers at any time. In Yamada's scheme, a tag authenticates a reader, and the tag responds to the reader if and only the authentication is successful [9].

Challenge-response protocol is widely used in authentication, since the protocol can avoid replay attacks. However, challenge-response protocol cannot be used to Yamada's scheme, since the tag cannot send a challenge without revealing its existence. Then, Yamada's scheme introduces a timestamp based authentication. A timestamp can be shared between tags and readers without communication. Therefore, the tag can authenticate the reader without revealing its existence.

However, the tag must have a battery, since the tag needs a clock to get a timestamp. Therefore, Yamada's scheme cannot apply to a passive tag, which doesn't contain a battery.

6.1.2 Increasing Time Interval

Increasing the time interval k is one of the ways to prevent the link attack. Yamane *et al.* proposed the concept of a silent period [7]. The silent period is a transition period that starts from the last output of the tag. During the silent period, the tag doesn't respond to any reader.

The simulation results (See Section 5) showed the link successful rate of our scheme is a same level as that of naive scheme, when k is 20. Therefore, if the length of the silent period is 20, we can ignore the real world constraints in this case.

During the silent period, even regular reader cannot read the tag. Therefore, a long silent period decreases the quality of service(QoS). It is necessary to set appropriate silent period that balances QoS and the danger of the link attack. The problem of setting appropriate silent period can be defined using the location tracking model (See Section 3).

6.1.3 Increasing Number of Tags

The simulation results show the link successful rate is high, when N is small. The more the number of tags N increases, the more the link successful rates decreases. Since the adversary can expect the user's future position at certain probability as far as he knows the user's past position, the link successful rate never decreases less than the probability, If the number of tag N exceeds the threshold value (in case of our simulation environment, about 40), the link successful rate is converged. We should set the system so that N can exceed the threshold value.

If RFID systems become more popular, we can easily satisfy the conditions such as 'the number of the tags in a building is more than 40'. However, it is difficult to satisfy the conditions on the first stage of RFID permeation. Therefore, there is a possibility that the spread of RFID stops due to privacy problems. One of the solutions to solve this permeation problem is increasing the N by installing dummy tags on various locations. And another solution is introducing the schemes described in previous subsections during the first stage of RFID permeation.

6.2. Acceptance of Link Attack

The simulation results show an adversary successes a link attack in short odds without ID information if the time interval k is short. In other words, we cannot avoid some link attack even if a tag changes its outputs every time. Then, we can choose the scheme that decreases the frequency of the update of the tag output if we tolerate some link attack. This scheme has a possibility that the cost of RFID system can be low without depraving the danger of the link attack.

6.3. Application of Link Attack

Up to here, we discussed the problem of the location tracking by an adversary. In this subsection, we discuss the location tracking for regular service.

A regular service provider can get tag ID from tag outputs using encryption key etc. Therefore, the situation of the link problem, in which service provider can't get a tag ID but can know the location of the tag, cannot be happened originally. However, there is a situation in which a reader fails to read the tag output due to Cyclic Redundancy Check(CRC) error, RFID tag collision, etc. In such a case, the reader cannot get the tag ID but can detect the existence of some tags. Therefore, we can use the link problem for error correction.

7. Conclusion

In this paper, we discussed the unlinkability and the real world constraints in RFID systems, and simulated the attack by an adversary. The simulation results showed the real world constraints have possibility that break the unlinkability. We also analyzed the simulation results from three viewpoints.

The next challenge is to solve a link problem removing the restriction $P_{err} = 0$. We want to discuss the link attack more deeply by solving the new link problem.

Acknowledgment

This work has been supported by the Grant-in-Aid for Creative Scientific Research No.14GS0218 and for Young Scientists No.18680009 of the Ministry of Education, Science, Sports and Culture (MEXT). We are grateful for their support.

References

[1] Y. Nohara, S. Inoue, K. Baba, and H. Yasuura. Quantitative evaluation of unlinkable ID matching schemes. In *2005 ACM Workshop on Privacy in the Electronic Society – WPES2005*, pages 55–60. ACM Press, Nov. 2005.

[2] Y. Nohara, T. Nakamura, K. Baba, S. Inoue, and H. 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.

[3] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to a privacy friendly tag. In *RFID Privacy Workshop@MIT*, Nov. 2003.

[4] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. 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.

[5] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[6] S. Inoue, D. Hagiwara, and H. Yasuura. Systematic error detection for RFID reliability. In *International Conference on Availability, Reliability and Security – ARES2006*, Apr. 2006.

[7] H. Yamane, L. Huang, M. Kanta, and S. Kaoru. Protection RFID location privacy through silent period. In *2006 Symposium on Cryptography and Information Security – SCIS2006*, Jan. 2006. in Japanese.

[8] F. Bourgeois and J.-C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–806, 1971.

[9] I. Yamada, S. Shiotsu, A. Itasaki, and S. Inano. Secure active RFID tag system. In *4th Workshop on UbiComp Privacy*, 2005.