RFID Security: A Method for Tracking Prevention

Jarosław Bernacki(✉) and Grzegorz Kołaczek

Department of Computer Science, Wrocław University of Technology, Wrocław, Poland
{jaroslaw.bernacki,grzegorz.kolaczek}@pwr.edu.pl

Abstract. RFID-tags are very small and low-cost electronic devices that can store some data. The most popular are passive tags that do not have own power source, which allows for far-reaching miniaturization. The primary use of RFID-tags is to replace barcodes. Their industrial importance is constantly growing because in contrast to barcodes, manual manipulation of the object code is not required. RFID-tags are also used for detection and identification of objects. This enables tracking of objects in technological processes. At the moment, the most widespread use of RFID tags is identification of sold goods. However, the possibility of tracking carries the risk that improper subject can track the tags and consequently track a person who is in possesion of tagged subject. Therefore in this paper a method for tracking prevention is considered.

Keywords: Internet of Things · RFID · Privacy protection · Tracking prevention

1 Introduction

Internet of Things (IoT) is the convergence of Internet with Radio Frequency IDentification (RFID), Sensor and smart objects. IoT can be defined as "things belonging to the Internet" to supply and access all of real-world information [13]. RFID is said to give rise to the IoT. RFID are systems that consist of three fundamental elements: *tags*, *reader* and a database system. Tags (also called *transponders*) are "small" electronic devices, highly constrained. They usually do not have own power source and are inductively powered during communication with the reader. They are not capable to perform strong crypto operations (even symmetric encryption). Reader (*transceiver*) is a device with quite big computational and energetic capabilities. Readers communicate with the tags via radio channel. The last part of RFID system is a database that stores information related with tags. Usually reader communicating with tags, uses a database system.

Unfortunately, RFID technology entails some privacy threats. One of them is tracking. For example, if a person is carrying an RFID-tag with static ID with no encryption or blinding, then tracking is easy [4]. In this case tracking

© The Author(s) 2016
T. Czachórski et al. (Eds.): ISCIS 2016, CCIS 659, pp. 241–249, 2016.
DOI: 10.1007/978-3-319-47217-1_26

is understood as a possibility of identifying the tag. Another problem is that authentication here does not help much, because it is generally used in order to prevent revealing tag's stored data [9]. Tag's ID is usually not "masked". Thus learning tag's ID is quite easily achievable and sufficient for tag tracking.

In this paper a method for tracking prevention is described. We propose that tags has a dynamic ID. For this purpose, a tag should have built-in random number generator. We assume that tag's ID can be modified, for instance after every tag activation. Then the tag generates new ID and sends it to the reader which saves it in the system database. Considered is a passive model of an adversary who eavesdrops all the traffic, but not all the time [10]. If the adversary misses several changes of tag's ID, it may be not possible to identify again targeted tag. History of all tags IDs is stored in the backend database.

The rest of the paper is organized as follows: next section gives a short overview of methods for privacy preserving/tracking protection in RFID systems. Section 3 presents proposed method for tracking prevention. In Sect. 4 preliminary experimental evaluation of proposed method is presented; finally the last section concludes this work and gives possible future directions.

2 Related Works

The risk associated with privacy has been recognized quite quickly [2]. Unfortunately, some RFID systems do not use any security mechanisms, so tags can be read by any reader, which is an obvious threat to privacy [12]. For instance, an ability to identify a tag, can deliver information about its owner. It is then possible to create a profil of an user, based on information collected from tags [7]. Thus so far many techniques for privacy protection have been proposed. In [9], there is proposed a method for tracking prevention. Considered is a model, where an attacker monitors a large fraction of interactions, but not all of them. Authors propose to make small changes with the tag's identifier. Tag does not have to perform any cryptographic functions.

Another method is "masking" tags, described in [4,14]. It assumes that a tag stores a list of pseudonyms $p_1, p_2, \ldots, p_k$ and every now and then changes them. An adversary would not know that for example p_i and p_j belong to the same tag, therefore such approach can effectively complicate recognizing a tag. However, if an adversary intercepts tag's list of pseudonyms, the whole idea is compromised. Another question worth considering is how many pseudonyms should have store. Should be taken into account that tag has strongly limited memory resources [4].

Popular method is the *kill* command which aim is to completely deactivate a tag [12]. However this approach strongly reduces functionality of the system [8]. Another possible solutions are: screening with Faraday Cage or physical destruction of antenna or other parts of a tag [8]. More advanced solution is called *active jamming*. It is based on actively broadcasting radio signals, what disrupts actions of any reader. However, this approach requires extra device [11].

In [6] there is proposed an extension of method from [15], where tag can be temporarily switched off and another tag is simulating tags of all possible IDs. Hence a reader is not able to determine a tag which established a connection.

Golle et al. proposed in [5] a method called *universal re-encryption*. This solution is based on the classical scheme ElGamal which allows for re-encryption of a ciphertext without knowledge about public key. Thereby computationally powerful devices can read from a tag its content, then re-encrypt it and save it back in the tag. In this case only tag's owner, who knows the proper private key, is able to track the tag. Further development of this idea was proposed in [1].

3 A Method for Tracking Prevention

3.1 System and Privacy Model

We assume that RFID system consists of several tags, a reader and the backend database. More formal definition is presented in Definition 1.

Definition 1 (RFID system). *Let $\mathcal{S}$ denote RFID system. $\mathcal{S}$ consists of reader $\mathcal{R}$, finite set of i tags (transponders) $\mathcal{T} = \{T_1, T_2, \ldots, T_i\}$ and database $\mathcal{DB}$ which stores information related with the tags. $\mathcal{DB}$ also stores for each tag $\mathcal{ID} = \{ID_1, ID_2, \ldots ID_n\}$ which is the history of all tags' IDs. ID_n is defined as history of IDs of tag's n: $ID_n = \{ID_n^1, ID_n^2, \ldots, ID_n^k\}$, where ID_n^k is the k-th ID of the n-th tag.*

It is assumed that tags are passive (powered only during the communication with the reader).

In Definition 2 we introduce a simple model of an adversary and his goals. We define adversary's goal similarly as in the scheme proposed in [3]. A passive adversary $\mathcal{A}$ eavesdrops all the communication between RFID system components (i.e. the *forward* and *backward channel*), but not all the time.

Definition 2 (Adversary's goal – *unlinkability* game). *Suppose that there exists list of n tags IDs: $\mathcal{ID} = \{ID_1, ID_2, \ldots ID_n\}$, where ID_n is defined as in Definition 1. Then, it is choosed $ID_x^k \in \mathcal{ID}$ which is the currently used ID of some tag $T_x \in \mathcal{T}$. The goal of the adversary is to guess x with the probability greater than $\frac{1}{n}$.*

In our approach we assume that adversary observing the communication between reader and a tag, can "miss" several queries. The goal of the adversary is to identify the tag, i.e. not to "lose" its ID.

3.2 Tracking Prevention

We propose a method ChangeID which can be used to make more difficult recognition a particular tag. This method assumes that a tag simply changes its own identifier by generating a new one. Then, a new ID is transferred to the reader which saves it in the backend database. This makes possible later identifying the tag. Below is presented an idea of method ChangeID.

1. Tag has a n-bit binary sequence which stands for its ID: $(b_1, \ldots, b_n) \in \{0,1\}^n$;
2. Next n bits are overwritten at random: a new sequence is created $(b_{i_1}, \ldots, b_{i_n})$, where for all $j \leq n$, $b_{i_j} \leftarrow b \in_U \{0,1\}$ is substituted from a uniform distribution.

This procedure can be performed after each activation of tag or, for instance at specified intervals. Note that none of sensitive data is transferred through the forward channel which is assumed to be easily eavesdropped [11,15]. It is likely that at average $n/2$ bits could remain unchanged.

Formally, this approach can be described as Algorithm 1.

ChangeID
Input: $(b_1, \ldots, b_n) \in \{0,1\}^n$
Output: $(b_{i_1}, \ldots, b_{i_n})$

for $j \leq n$ **do**
  $b_{i_j} \leftarrow b \in_U \{0,1\}$
end

Algorithm 1: ChangeID procedure

Note that this procedure has low requirements in terms of computational complexity.

3.3 Problem of Ambiguity

One should consider that generating random IDs may cause generation of two (or more) the same IDs. Such a situation is undesirable in most systems and sometimes can be critical to their functioning. Although intuitively the probability of happening such situation is quite small, one can assume that the reader (after each changing tag's ID) checks in the backend database, if generated ID already exists. If does, then tag simply could be asked to perform another ChangeID operation. Similarly, if new generated ID is the same as the previous one, another performance of ChangeID could be done. In this case we assume that considered is a sequential access model. This situation is presented in Table 1.

4 Preliminary Experimental Evaluation

We conducted a simple experiment in which we implemented a function generating different lengths random sequences (strings) that could act as a tag identifier. We checked the possible links between distances of these sequences and examined Hamming distances between them.

Table 1. ChangeID protocol

Reader		Tag
	$\xrightarrow{\text{hello}}$	
		s =ChangeID
	$\xleftarrow{s}$	
`if` s exists in $\mathcal{DB}$ `then` query for another ChangeID `else` save s		

We divided an experiment into 5 trials, in each trial 80 sequences of the following lengths were generated:

1. 32 bits length;
2. 64 bits length;
3. 128 bits length;
4. 256 bits length;
5. 512 bits length.

We analyzed Hamming distances between sequences in each trial (for example, sequence (1) with sequence (2); (2) with (3), ...). For the clarity, we normalized results of Hamming distance on the interval $[0, 1]$.

4.1 Distances in 32 Bits Trial

In Fig. 1 there are presented distances between adjacent sequences in 32-bits trial. Similarity is mostly at the level 0.7–0.9.

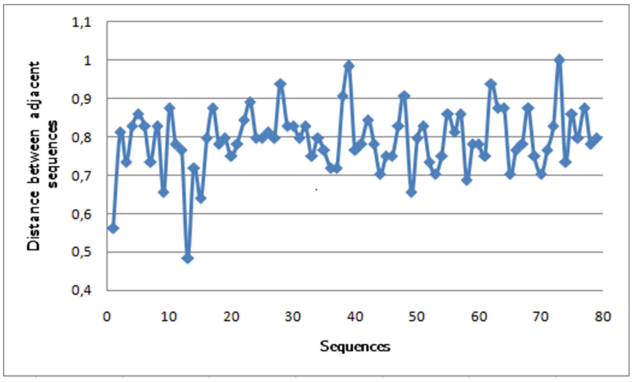

Fig. 1. Distances between adjacent sequences (total number of sequences: 80)

On the X-axis the are next sequences; Y-axis presents the normalized distance between adjacent sequences.

Table 2. Fragment of generated sequences for 32 bits trial

	Generated sequence	H_d	$Norm$
(1)	11011111001011011001010011001010		
(2)	10111100010110011100111100110100	21	0.66
(3)	10100010100111001110100010010111	27	0.84
(4)	11010100000100011111101000001101	18	0.56
(5)	11111010000100001100111000001010	21	0.66
(6)	11111110000100111000010011000011	22	0.69
(7)	10111010000010001011000000100011	28	0.88
(8)	11001000101011011101011110100000	25	0.78
(9)	11000010000100010110110110101111	27	0.84
...	...		
(79)	10111010001000111000110001111 1010		
(80)	11100010101001011011011001110000	23	0.72

In Table 2 there are presented several generated sequences and distances between adjacent sequences. H_d for i-th sequence stands for Hamming distance between the $i-1$ and i sequence, $Norm$ denotes value of normalization at $[0, 1]$. For instance, H_d between (1) and (2) equals 21; in normalized way: 0.66, and so on.

For the clarity, we do not present full results of this and the other trials.

4.2 Summary

The Table 3 shows minimum and maximum values of normalized at $[0, 1]$ distances in each trial.

Table 3. Minimum and maximum values of distances between sequences within each trial

	32 bits	64 bits	128 bits	256 bits	512 bits
Min	0.38	0.48	0.61	0.71	0.76
Max	1	1	0.97	0.92	0.88

Intuitively, the shortest sequence, the higher probability for generating two quite similar sequences (minimum distance for 32 bits is 0.38, for 64 bits – 0.48). The longer sequence, the greater differences (for instance, 0.76 for 512 bits sequences). These results are also showed in Figs. 2 and 3, respectively.

The longer tag's ID, the smaller probability of generating two the same sequences; however longer sequence requires more tag's memory.

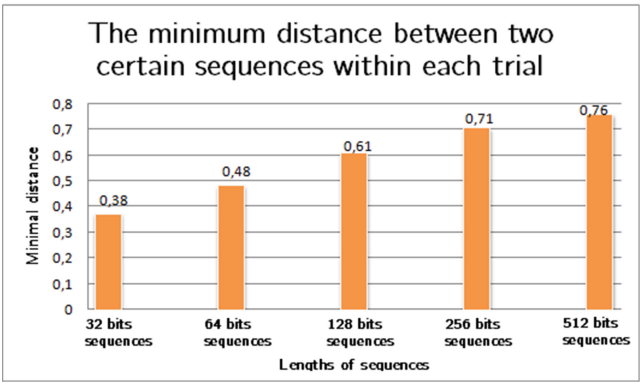

Fig. 2. The minimum (normalized) Hamming distance within each trials

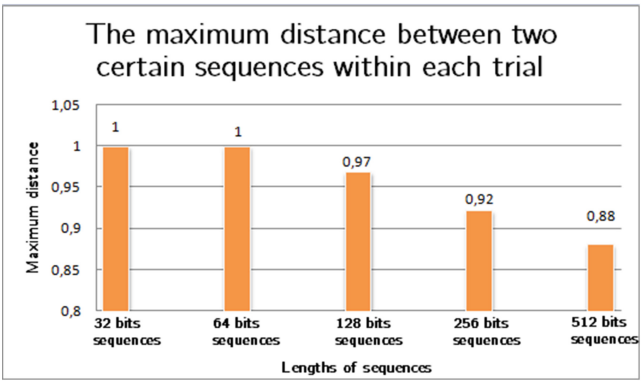

Fig. 3. The maximum normalized Hamming distance within each trials

5 Conclusion and Future Works

In this paper, a method for tracking prevention for RFID-tags was proposed. It was assumed that tag is able to change its own identifier by generating a random sequence and replacing earlier ID. If an adversary is not able to monitor the tag all the time, this method after a certain amount of execution can effectively complicate recognition of the tag. Preliminary experimental evaluation showed that unlinkability between tags IDs is at satisfactory level.

If future works it is planned to give a formal estimation of minimal number of ID modification in order to achieve good level of privacy. Also a simulation of implementation is considered to be carried out. Another problem to consider is to propose a method for settlement of the ambiguity of tags' IDs not in the sequential access model but in situation of independent and parallel operations of (several) readers.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. Ateniese, G., Camenisch, J., de Medeiros, B.: Untracable RFID tags via insubvertible encryption. In: Proceedings of 12th ACM Conference on Computer and Communications Security (2005)
2. Chan, H., Perrig, A.: Security and privacy in sensor networks. Computer **36**(10), 103–105 (2003)
3. Cichoń, J., Klonowski, M., Kutyłowski, M.: Privacy protection for RFID with hidden subset identifiers. In: Indulska, J., Patterson, D.J., Rodden, T., Ott, M. (eds.) PERVASIVE 2008. LNCS, vol. 5013, pp. 298–314. Springer, Heidelberg (2008)
4. Garfinkel, S.L., Juels, A., Pappu, R.: RFID privacy: an overview of problems and proposed solutions. IEEE Secur. Priv. **3**(3), 34–43 (2005)
5. Golle, P., Jakobsson, M., Juels, A., Syverson, P.F.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
6. Juels, A., Rivest, R.L., Szydlo, M.: The blocker tag: selective blocking of RFID tags for consumer privacy. In: ACM Conference on Computer and Communications Security, pp. 103–111 (2003)
7. Karthikeyan, S., Nesterenko, M.: RFID security without extensive cryptography. In: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 63–67. ACM. New York (2005)
8. Klonowski, M.: Algorytmy zapewniajace anonimowość i ich matematyczna analiza. PhD Dissertation (in Polish), Wrocław University of Technology, Poland (2009)
9. Klonowski, M., Kutyłowski, M., Syga, P.: Chameleon RFID and tracking prevention. In: Radio Frequency Identification System Security, RFIDSec Asia 2013, pp. 17–29 (2013)
10. Kutyłowski, M.: Anonymity and rapid mixing in cryptographic protocols. In: The 4th Central European Conference on Cryptology, Wartacrypt (2004). http://kutylowski.im.pwr.wroc.pl/articles/warta2004.pdf. Accessed 13 Feb 2016
11. Luo, Z., Chan, T., Li, J.S.: A lightweight mutual authentication protocol for RFID networks. In: Proceedings of 2005 IEEE International Conference on e-Business Engineering (ICEBE 2005), IEEE Xplore, pp. 620–625 (2005)
12. Medaglia, C.M., Serbanati, A.: An overview of privacy and security issues in the Internet of Things. In: Giusto, D., Iera, A., Morabito, G., Atzori, L. (eds.) The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications, pp. 389–395. Springer, New York (2010)

13. Singh, D., Tripathi, G., Jara, A.J.: A survey of Internet-of-Things: future vision, architecture, challenges and services. In: 2014 IEEE World Forum on Internet of Things (WF-IoT), pp. 287–292 (2014)
14. Vajda, I., Buttyan, L.: Lightweight authentication protocols for low-cost RFID tags. In: Laboratory of Cryptography and Systems Security (CrySyS) (2003)
15. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)