Information Assurance and Security Letters 1 (2010) 006-011

Improvement of Two Lightweight RFID Authentication Protocols

Kuo-Hui Yeh[1] and N.W. Lo[2]

National Taiwan University of Science and Technology, Department of Information Management,
#43, Sec.4, Keelung Rd., Taipei, 106, Taiwan, R.O.C
[1]*D9409101@mail.ntust.edu.tw*; [2]*nwlo@cs.ntust.edu.tw*

***Abstract*:** Recently, lightweight RFID authentication protocol has been investigated due to the awareness of practical requirements on individual privacy, system security and resource limitation of low cost tags. Research community demonstrates major advancements in this interesting research area of designing a robust access control and information retrieval mechanism for RFID system. In 2008, two well-studied lightweight RFID authentication mechanisms are introduced by Burmester et al. [2] and Peris-Lopez et al. [25] to support tag anonymity, data confidentiality and forward security in which only primitive computation modules such as pseudo random number generator and simple bitwise operation are required. Nevertheless, based on our analysis, both of these two schemes are vulnerable to desynchronization attack. The secret key value, which is shared between the tag and the backend database, can be out of synchronization by just performing a series of challenge-response operations. To remedy this authentication flaw, we are motivated to develop two countermeasure mechanisms which deliver stronger security robustness.

***Keywords*:** Authentication, Privacy, RFID, Security

1. Introduction

As Radio Frequency IDentification (RFID) technology has been widely adopted in lots of novel applications and innovations, the security issues and privacy concerns on RFID system are promptly focused by individual, industry and even academic. In comparison with traditional wireless transmission technology, the nature, i.e. restricted computation ability and limited memory space, of low-cost tags makes existing RFID systems vulnerable to many security attacks [28]. The insecure communication environment also paves the way for new privacy threats such as industrial espionage attack, tag carrier tracking, breadcrumb threat and RFID cloning in RFID based applications [11]. Hence, over the last few years, significant efforts have been devoted to RFID security research area and one of the most interesting studies is to develop a secure authentication scheme for RFID system with resource constrained tags.

So far, existing RFID authentication protocols can be briefly classified into four classes [5], i.e. Full-fledged, Simple, Lightweight and Ultralightweight. In Full-fledged and Simple classes [3, 7-8, 15, 19], tags are assumed to support conventional cryptographic functions such as symmetric encryption, one-way hash function and even public key cryptosystem technology. Such kind of tag is too costly to be adopted in inventory management, supply-chain logistic and retailer operations which are envisioned as major applications of RFID technology. It is obvious that Lightweight and Ultralightweight based authentication schemes are more suitable for RFID application systems as tags only require to perform primitive computation modules such as random number generator, cyclic redundancy code checksum and arithmetic bitwise operations. The characteristic of such low-cost tag is very helpful to achieve the pervasive usage and development of RFID technology and applications. The research community also shows this tread with numerous significant efforts [1-2, 4-6, 9, 13-14, 16-18, 20-27]. Hence, we argue that Lightweight and Ultralightweight based authentication mechanism will be the best candidate technology for securing existing or future RFID system due to the tradeoff between security robustness, system efficiency and tag practicality. Moreover, in our opinion all mechanisms from these two categories can be referred to "lightweight protocol", since all involved operations, i.e. random number generator, cyclic redundancy code checksum and arithmetic bitwise operations, are currently affordable by a low-cost passive tag [10, 22-26].

To foster and promote the interoperability of RFID technology, EPC Gen2 standard [10], which provides a platform for building interoperable RFID protocols, is well known and supported by enterprises all over the world. To pursue the best balance between tag cost and system functionality, EPC Gen2 supports several efficiency and simple security guarantees such as efficient tag-identifying scheme, memory standardization, on-chip 16-bit pseudo-random number generator and 16-bit cyclic redundancy code. However, for the aspect of information security, EPC Gen2 standard does not thoroughly consider privacy invasion problems and data security issues. Hence, in order to protect privacy related information between communicating parties and defend counterfeit data attack against RFID systems, several authentication schemes [2, 4, 6, 9, 13, 18] conformed to EPC Gen2 standard have been proposed to support stronger security robustness. In 2008, Burmester and Medeiros [2] developed a well-studied EPC Gen-2 conforming authentication protocol, called TRAP-3, which adopts a robust pseudo random function (PRF) instead of weak 16-bit random number generator (RNG) to achieve forward security and tag anonymity in RFID system. Note that PRF is based on RNG and the EPC Gen2 compliant authentication scheme belongs to Lightweight category [5]. However, according to the security analysis conducted by us, TRAP-3 is insecure against a desynchronizaiton attack. The secret key value, which is shared between the tag and the backend server/database in TRAP-3, can be out of synchronization by just performing a series of challenge-response operations. On the other hand, since Peris-Lopez et al. published a series of Ultralightweight based RFID authentication protocols [22-24] to pursue strong tag anonymity and data security, various studies [5, 14, 16-17, 25, 27] had been developed in which only very lightweight arithmetic bitwise operations are required at tag end. Among

Received November 20, 2009

them, the authentication scheme in [25], called Gossamer protocol, is a more robust and efficient candidate technique for securing RFID system with very low cost tags. Nevertheless, Gossamer protocol is vulnerable to synchronizaiton attack also. To remove this serious security flaw from TRAP-3 and Gossamer protocols, in this study we are motivated to propose a novel key updating mechanism. With our newly proposed countermeasure scheme, both of TRAP-3 and Gossamer protocols could be better convinced. In brief, the purpose of this article is to explore further into RFID authentication research field by providing the security analysis of two recently published authentication mechanisms, i.e. TRAP-3 and Gossamer protocols, and its improvement.

2. Cryptanalysis of TRAP-3

2.1 Review of TRAP-3

In TRAP-3, the authors utilize a 32-bit PRF and a 48-bit key to achieve the secure message transmission between the tag and the reader. The PRF is defined as follows [2, 12].

Let G be an n-bit RNG and I be an n-bit number. Let $G_0(I)$ be the first n-bit number output by G and $G_1(I)$ be the next n-bit number. Let $X= X_0, X_1, X_2, \ldots, X_t$, $t \geq n$, be a t-bit number and $G_X(I)=G_{X_t}(Z_{t-1})$, where $Z_{t-1}=(G_{X_{t-1}}(\ldots(G_{X_1}(G_{X_0}(I)))\ldots)$. Define PRF $f_l(\ ): \{0, 1\}^t \rightarrow \{0, 1\}^n$ by $f_l(X) = G_X(I)$.

In TRAP-3, each tag T stores a 16-bit number P and a 48-bit key $K=k_0||k_1$, where the length of k_0 is 32-bit. The server S maintains for each tag T in a database (DB) an entry with two 48-bit keys K^{old}, K^{cur} and the identity of T, i.e. $<K^{old}, K^{cur}, id(T)>$. Note that the communication channel between the reader (R) and the server (S) is assumed to be secure. Initially, P is assigned a random value, K and K^{cur} are set to the same random value, and K^{old} is null. The detailed procedures of TRAP-3 are described as follows (Figure 1).

- $S \rightarrow R \rightarrow T$: a 16-bit random nonce N.

R first sends a request to S to ask a random nonce N which will be forwarded to T. Once receiving N, T computes $L=(k_1 \oplus P)||N$, and draws three 32-bit numbers M_1, M_2, M_3 from $f_{k_0}(L)$. Next, T parses $M_1=M_{10}||M_{11}$, $M_2=M_{20}||M_{21}$ and $M_3=M_{30}||M_{31}$ into several 16-bit numbers and sends $\{P, M_{10}\}$ as a response back to R. Then, T updates $P=M_{11}$.

- $T \rightarrow R \rightarrow S$: P, M_{10}

R forwards $\{P, M_{10}\}$ to S. After S receives the response message $\{P, M_{10}\}$, S iteratively retrieves the stored key values K^{old} and K^{cur} from each tuple in DB and performs the following computations to examine which tag (or which key value) is currently involved. If S cannot find a corresponding record for $\{N, P, M_{10}\}$, S terminates current session. Otherwise, S sets $K^j=K^{old}$ or $K^j=K^{cur}$, and continues the next processes of current session.

(1) Calculate $L'=((k_1 \text{ from } K^{old}) \oplus P)||N$ or $L'=((k_1 \text{ from } K^{cur}) \oplus P)||N$.
(2) Draw three 32-bit numbers M_1', M_2', M_3' from $f_{k_0}(L')$.
(3) Parse $M_1'=M_{10}'||M_{11}'$, $M_2'=M_{20}'||M_{21}'$ and $M_3'=M_{30}'||M_{31}'$.
(4) Check $M_{10}=M_{10}'$?

- $S \rightarrow R$: "Details of T", M_{20}', "end session"

S sends details regarding the tag T, M_{20}', and end session command to R. Next, R forwards M_{20}' to T. Meanwhile, S updates $K^{old}=K^j$ and $K^{cur}=M_{21}'||M_3'$.

- $R \rightarrow T$: M_{20}'

When T obtains M_{20}', it checks whether $M_{20}'=M_{20}$ or not. If it holds, T updates $K=M_{21}||M_3$. Otherwise, T terminates current session.

2.2 Desynchronization attack on TRAP-3

In this section, we demonstrate that TRAP-3 cannot defend against a desychronzation attack. The detailed processes of this vulnerability are described as follows.

First of all, in Figure.2 a given synchronized tag in which the secret information $<id(T), K>$ maintained at tag side equals to the values $<id(T), K^{old}, K^{cur}>$ stored in database (DB) is assumed. Note that K^{old}=null and $K^{cur}=K=k_0||k_1$. Now we suppose an adversary C, who possesses two legal readers A and B, intends to desynchronize the secret key K shared between T and S. Adversary C first utilizes its own legal reader A to issue a normal request to server, and obtains a random number N_1 which is soon forwarded to T. Next, T sends back a response message $\{P, M_{10}\}$. Then, C abandons current session with T.

Secondly, in Figure.3 C uses it's another legal reader B to invoke a normal session of TRAP-3 with victim tag T. After successfully performing all normal procedures of TRAP-3 scheme, C recognizes that current secret key value K shared between DB and T are as follows.

Secret key value K stored at T side = $M_{21}'||M_3'$
Secret key value K^{old} stored at DB side = $k_0||k_1$
Secret key value K^{cur} stored at DB side = $M_{21}'||M_3'$

After that, in Figure.4 adversary C continues the previous uncompleted procedure in Figure.2 to issue the previously obtained message $\{P, M_{10}\}$ to S. When receiving $\{P, M_{10}\}$, S examines the validity of this incoming message with random number N_1. Obviously, the verification of $\{N_1, P, M_{10}\}$ will be successfully examined with the help of old secret key value $K^{old}=k_0||k_1$. More concretely, S misunderstands that T utilizes the old key values, i.e. $K^{old}=k_0||k_1$, to communicate with it as the message $\{P, M_{10}\}$ is involved with the old secret key value $K^{old}=K=k_0||k_1$.

After executing the authentication procedures, S updates the corresponding secret key values of T to new ones, i.e. $K^{old}=k_0||k_1$ and $K^{cur}=M_{21}||M_3$. Obviously, the secret key value shared between T and DB is out of synchronization, where $\{M_{21}'||M_3'\}$ and $\{M_{21}||M_3\}$ is involved with different values set $\{N_2, P'\}$ and $\{N_1, P\}$, respectively.

Secret key value K stored at T side = $M_{21}'||M_3'$
Secret key value K^{old} stored at DB side = $k_0||k_1$
Secret key value K^{cur} stored at DB side = $M_{21}||M_3$

3. Countermeasure for the security vulnerability identified on TRAP-3

In previous section, our proposed attack procedures show that in TRAP-3 the secret key value, which is shared between T and DB, can easily be out of synchronization. This weakness results from that an adversary can appropriately utilize legitimate messages acquired in previous sessions to impersonate victim entity T and communicate legally with backend server S. Without a carefully-designed key updating mecha-

nism, both of T and S are easily to be confused with current state of stored secret key value. Based on this observation, we develop a novel key updating mechanism to remedy this security vulnerability in TRAP-3.

- Key updating mechanism at S side:

Once S intends to update current secret key value, S computes $f_{k0}(k_1||k_1)$ and draws two 32-bit numbers M_4=$M_{40}||M_{41}$ and M_5 from the computation result $f_{k0}(k_1||k_1)$. Next, S updates K^{old}=K^j and K^{cur}=$M_{40}||M_5$.

- Key updating mechanism at T side:

Once T wants to update current secret key value, T computes $f_{k0}(k_1||k_1)$ and draws two 32-bit numbers M_4=$M_{40}||M_{41}$ and M_5 from the computation result $f_{k0}(k_1||k_1)$. Next, T updates K^{cur}=$M_{40}||M_5$.

In our proposed key updating mechanism, each new secret key value K^{cur} are derived from $f_{k0}(k_1||k_1)$ which is based on current used key value K^j=$k_0||k_1$. Hence, in Figure.4 even if the malicious attacker utilizes the previous response message $\{P, M_{10}\}$ involved with the old secret key value K^{old} to communicate legally with S, the new updated key value K^{cur} at both of T and DB sides will be the same. Obviously, our proposed key updating mechanism can resist to desynchronizaiton attack.

4. Cryptanalysis of Gossamer protocol

4.1 Review of Gossamer protocol

In 2008, Peris-Lopez et al. [25] developed an ultralightweight authentication mechanism, called Gossamer protocol, which is inspired by SASI scheme [5]. The Gossamer protocol is developed to eliminate the security vulnerabilities, i.e. desynchronization and disclosure attacks [14, 27], on SISA protocol. Here we describe the message exchanged between the reader (backend database) and the tag in a normal session of Gossamer protocol.

In Gossamer protocol, each tag stores a static identifier (ID), two records (old/new) of an index-pseudonym (IDS) and two keys (k_1 and k_2) which are two-records design (old/new). In the backend database, a static identifier (ID), an index-pseudonym (IDS) and two keys (k_1 and k_2) are required to be maintained. Note that the authors assume that the tag can operate several simple functions such as bitwise XOR ($\oplus$), bitwise AND ($\vee$), bitwise OR ($\wedge$), $Addition$ mod 2^m (+), circular shift rotation ($Rot(x, y)$) and $MixBits$ functions. In addition, random number generation (i.e. n_1 and n_2) is required on the reader. The Gossamer protocol is divided into three stages: tag identification phase, mutual authentication phase and updating phase. In the identification phase, the reader sends a *hello* message to the tag, and the tag responds with its IDS. Based on the received IDS, the reader can probe the corresponding information of the tag (ID, k_1 and k_2), and the protocol then turns into the mutual authentication phase. In this phase, the reader and the tag can authenticate each other, and the IDS and keys are subsequently updated in the next updating phase.

Reader → *Tag*: *Hello*
Tag → *Reader*: *IDS*
Reader → *Tag*: $A||B||C$

The reader generates two nonce values n_1 and n_2, and builds $A||B||C$ which will be sent to the tag.

$$A=Rot((Rot(IDS+k_1+\pi+n_1, k_2)+k_1, k_1);$$
$$B=Rot((Rot(IDS+k_2+\pi+n_2, k_1)+k_2, k_2);$$
$$n_3=MixBits(n_1, n_2);\ n_1'=MixBits(n_3, n_2);$$
$$k_1^*=Rot((Rot(n_2+k_1+\pi+n_3, n_2)+k_2\oplus n_3, n_1)\oplus n_3;$$
$$k_2^*=Rot((Rot(n_1+k_2+\pi+n_3, n_1)+k_1+n_3, n_2)+n_3;$$
$$C=Rot((Rot(n_3+ k_1^*+\pi+n_1', n_3)+k_2^*\oplus n_1', n_2)\oplus n_1';$$
$$\pi=0x3243F6A8885A308D313198A2.$$

From message A and B, the tag can obtain two nonce values n_1 and n_2 respectively. Then the tag computes C' and checks whether the result is equal to the received value C. If these two values are the same, the tag sends D and updates the values of IDS, k_1 and k_2. Note that the tag is given the added requirement of storing the old value of IDS, k_1 and k_2 to avoid the desynchronization attack.

$$C'=Rot((Rot(n_3+ k_1^*+\pi+n_1', n_3)+k_2^*\oplus n_1', n_2)\oplus n_1';$$
$$D=Rot((Rot(n_2+ k_2^*+ID+n_1', n_2)+k_1^*+n_1', n_3)+n_1';$$
$$n_2'=MixBits(n_1', n_3);$$
$$IDS_{old}=IDS;\ k_{1\_old}=k_1;\ k_{2\_old}=k_2;$$
$$IDS_{new}=Rot((Rot(n_1'+k_1^*+IDS+n_2', n_1')+k_2^*\oplus n_2', n_3)\oplus n_2';$$
$$k_{1\_new}=Rot((Rot(n_3+k_2^*+\pi+n_2', n_3)+k_1^*+n_2', n_1') +n_2';$$
$$k_{2\_new}=Rot((Rot(IDS_{new}+k_2^*+\pi+k_{1\_new}, IDS_{new})+k_1^*+k_{1\_new}, n_2')+k_{1\_new};$$

Tag → *Reader*: D

The reader computes D' value. If the computed D' value is equal to the received D value, updates IDS, k_1 and k_2 in the same way as the tag.

$$D'=Rot((Rot(n_2+ k_2^*+ID+n_1', n_2)+k_1^*+n_1', n_3)+n_1';$$
$$n_2'=MixBits(n_1', n_3);$$
$$IDS=Rot((Rot(n_1'+k_1^*+IDS+n_2', n_1')+k_2^*\oplus n_2', n_3)\oplus n_2';$$
$$k_1=Rot((Rot(n_3+k_2^*+\pi+n_2', n_3)+k_1^*+n_2', n_1') +n_2';$$
$$k_2=Rot((Rot(IDS+k_2^*+\pi+k_1, IDS)+k_1^*+k_1, n_2')+k_1;$$

4.2 Desynchronization attack on Gossamer protocol

Similarly, Gossamer protocol cannot defend against the desynchronization attack. The corresponding malicious procedures are as follows.

First, a given synchronized tag in which the secret information (IDS_{new}, k_{1_new}, k_{2_new}) maintained at the tag side equals to the values (IDS, k_1, k_2) stored in the backend database is assumed. Now we suppose the reader intends to query the tag. During a normal operation process of Gossamer protocol, the attacker eavesdrops and records the transmitted messages $A||B||C$. At the end of the protocol, the attacker interrupts the message D and this results in that the backend database will not update the information (IDS, k_1, k_2) associated with the tag. However, the tag will update the secret information as follows. For clarity, we denote the old secret information as (IDS_1, k_{1_1}, k_{2_1}) and updated information as (IDS_2, k_{1_2}, k_{2_2}) at current session.

(IDS_{old}, k_{1_old}, k_{2_old}) stored in the tag=(IDS_1, k_{1_1}, k_{2_1})
(IDS_{new}, k_{1_new}, k_{2_new}) stored in the tag=(IDS_2, k_{1_2}, k_{2_2})
(IDS, k_1, k_2) stored in the server/database=(IDS_1, k_{1_1}, k_{2_1})

Next, the attacker lets the reader and the tag run the Gossamer protocol without being intervened. In this communication process, the tag will utilize the old values, i.e. IDS_{old}, k_{1_old} and k_{2_old}, to communicate with the reader as the IDS stored in the backend database is the old one. After performing all authentication procedures, the database will up-

date the corresponding values of the tag to a new one (IDS_3, k_{1_3}, k_{2_3}) due to two new random nonce values generated by the reader. At the tag side, the secret information will update as follows.

(IDS_{old}, k_{1_old}, k_{2_old}) stored in the tag=(IDS_1, k_{1_1}, k_{2_1})
(IDS_{new}, k_{1_new}, k_{2_new}) stored in the tag=(IDS_3, k_{1_3}, k_{2_3})

Finally, the attacker utilizes its own legal reader to inquire the tag. The tag first replies IDS_{new}, which is IDS_3, and then sends IDS_{old}, which is IDS_1, when the attacker pretends that he/she cannot find the IDS_3 in the backend database and requests the IDS_{old} value. The attacker then transmits the previously eavesdropped values $A||B||C$ to the tag. Since these values are computed by the legal reader with IDS_1 previously, the tag cannot distinguish whether these values are truly issued from a legal user or not, and accepts these values. After performing the update procedures, the secret information at the tag side will be as follows.

(IDS_{old}, k_{1_old}, k_{2_old}) stored in the tag=(IDS_1, k_{1_1}, k_{2_1})
(IDS_{new}, k_{1_new}, k_{2_new}) stored in the tag=(IDS_2, k_{1_2}, k_{2_2})

Obviously, the secret information stored in the tag side and in the backend database side is out of synchronization now.

(IDS, k_1, k_2) stored in the server/database=(IDS_3, k_{1_3}, k_{2_3})

5. Countermeasure for the security vulnerability identified on Gossamer protocol

In this section, we provide an enhanced key update mechanism to remedy the desynchronization attack on Gossamer protocol.

- Key updating mechanism at server/database side:

Once server intends to update current secret key value, server calculates $IDS_{new}=PRNG(IDS_{old})$, $k_{1\_new}=PRNG(k_1)$ and $k_{2\_new}=PRNG(k_2)$, where $PRNG$ is a lightweight pseudonym random generator proposed in [26].

- Key updating mechanism at tag side:

Once tag wants to update current secret key value, tag computes $IDS_{old}=IDS$, $k_{1\_old}=k_1$, $k_{2\_old}=k_2$, $IDS_{new}=PRNG(IDS_{old})$, $k_{1\_new}=PRNG(k_1)$ and $k_{2\_new}=PRNG(k_2)$.

Similarly, in our proposed key update mechanism, each new secret key value k_{1_new} and k_{2_new} are both derived from the old secret key k_{1_old} and k_{2_old}. Even if the malicious attacker intends to invoke a desynchronization attack, the newly updated key value k_{1_new} and k_{2_new} at tag and server/database sides will always be identical. It is obvious that our remedy can resist to the identified desynchronizaiton attack.

6. Conclusion

In this paper, we have reported the security vulnerability on two well-studied RFID lightweight authentication protocols [2, 25] despite the authors' claim of security robustness. Based on our cryptanalysis, an adversary only requires performing a series of challenge-response operations to make the secret key value shared between the tag end and the server end out of synchronization. Our results indicate that the schemes proposed by Burmester et al. and Peris-Lopez et al. fail to commit their claimed security requirement and accordingly more detailed security analyses are required to be done by the designers. To eliminate this authentication flaw, we develop two novel key updating mechanisms as the countermeasure and achieve security enhancement in these two schemes. Instead of adopting a whole new value, the new secret key is always derived from the currently used one. This design prevents the shared secret key from being out of synchronization. With our proposed remedy, both of TRAP-3 and Gossamer protocols could be more convinced with better security intensity.

Acknowledgment

The authors gratefully acknowledge the support from TWISC project sponsored by the National Science Council, Taiwan, under the Grants No NSC 98-2219-E-011-001.

References

[1] J. Ayoade, "Security implications in RFID and authentication processing framework," *Computers & Security*, vol.25, no.3, May 2006, pp.207-212.

[2] M. Burmester and B. de Medeiros, "The Security of EPC Gen2 Compliant RFID Protocols, " *LNCS 5037*, in *Proc. of the 6th International Conference of Applied Cryptography and Network Security*, 2008, pp.490-506.

[3] Y. Chen, J.-S. Chou and H.-M. Sun, "A novel mutual authentication scheme based on quadratic residues for RFID systems," *Computer Networks*, vol.52, no.12, 2008, pp.2373-2380.

[4] C.-L. Chen and Y.-Y. Den, "Conformation of EPC class 1 generation 2 standards RFID system with mutual authentication and privacy protection," *Engineering Applications of Artificial Intelligence*, vol.22, no.8, Dec. 2009, pp.1284-1291.

[5] H.-Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," *IEEE Trans. on Dependable and Secure Computing*, vol.4, no.4, 2007, pp.337-340.

[6] H.-Y. Chien and C.-H. Chen, "Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards," *Computer Standards & Interfaces*, vol.29, no.2, 2007, pp.254-259.

[7] M. Conti, R. Di Pietro and L. V. Mancini, "RIPP-FS: an RFID identification, privacy preserving protocol with forward secrecy," in *Proc. of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2007, pp.229-234.

[8] M. Conti, R. Di Pietro, L. V. Mancini and A. Spognardi, "FastRIPP: RFID privacy preserving protocol with forward secrecy and fast resynchronization," in *Proc. of the 33rd Annual Conference of the IEEE Industrial Electronic Society*, 2007, pp.52-57.

[9] D.N. Duc, J. Park, H. Lee and K. Kim, "Enhancing Security of EPCglobal GEN-2 RFID Tag against Traceability and Cloning," in *Proc. of the 2006 Symposium on Cryptography and Information Security*, 2006.

[10] *EPC™ Radio-Frequency Identification Protocols Class 1 Generation-2 UHF RFID Protocol for Communication at 860-960 MHz Version 1.0.9*, EPCGlobal Inc., Dec. 2005.

[11] S.L. Garfinkel, A. Juels and R. Pappu, "RFID Privacy: an overview of problems and proposed solutions," *IEEE*

Security & Privacy Magazine, vol.3, no.3, 2005, pp.34-43.

[12] O. Goldreich, S. Goldwasser and S. Micali, "How to construct pseudorandom functions," *Journal of the ACM*, vol.33, no.4, 1986.

[13] D. Han and D. Kwon, "Vulnerability of an RFID authentication protocol conforming to EPC class 1 generation 2 standards," *Computer Standards & Interfaces*, vol.31, no.4, June 2009, pp.648-652.

[14] J.C. Hernandex-Castro, J.M. Estevex-Tapiador, P. Peris-Lopez and J.-J. Quisquater, "Cryptanalysis of the SASI ultralightweight RFID Authentication Protocol," *eprint arXiv:0811.4257*, Nov. 2008.

[15] T. V. Le, M. Burmester and B. de Medeiros, "Universally Composable and Forward-secure RFID authentication and Authenticated Key Exchange," in *Proc. of the 2nd Asian ACM Symposium on Information, Computer and Communications Security*, 2007, pp.242-252.

[16] T. Li and R.H. Deng, "Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol," in *Proc. of the 2nd International Conference on Availability, Reliability and Security*, 2007, pp.238-245.

[17] T. Li and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols," in *Proc. of IFIP International Federation for Information Security*, 2007, pp.108-120.

[18] N.W. Lo and K.-H. Yeh, "An Efficient Mutual Authentication Scheme for EPCglobal Class-1 Generation-2 RFID System," *LNCS 4809*, in *Proc. of the 2nd International Workshop on Trustworthiness, Reliability and services in Ubiquitous and Sensor networks*, 2007, pp.43-56.

[19] N.W. Lo and K.-H. Yeh, "Novel RFID authentication schemes for security enhancement and system efficiency," *LNCS 4721*, in *Proc. of the 4th VLDB Workshop on Secure Data Management*, 2007, pp.203-212.

[20] J. Munilla and A. Peinado, "*HB-MP: A Further Step in the HB-Family of Lightweight Authentication Protocols*," *Computer Networks*, vol.51, no.9, June 2007, pp.2262-2267.

[21] K. Oua and Raphael C.-W. Phan, "Privacy of recent RFID authentication protocols," *LNCS 4991*, *in Proc. of 4th International Conference on Information Security Practice and Experience*, 2008, pp.263-277.

[22] P. Peris-Lopez, J.C. Hernandex-Castro, J.M. Estevex-Tapiador, and A. Ribagorda, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-Cost RFID Tags," in *Proc. of the 2nd Workshop RFID Security*, 2006.

[23] P. Peris-Lopez, J.C. Hernandex-Castro, J.M. Estevex-Tapiador, and A. Ribagorda, "EMAP: An Efficient Mutual Authentication Protocol for Low-Cost Tags," in *Proc. of the OTM Federated Conferences and workshop: IS Worksho*p, 2006.

[24] P. Peris-Lopez, J.C. Hernandex-Castro, J.M. Estevex-Tapiador, and A. Ribagorda, "M2AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags," *LNCS 4159*, *in Proc. of International Conference on Ubiquitous Intelligence and Computing*, 2006, pp.912-923.

[25] P. Peris-Lopez, J.C. Hernandex-Castro, J.M. Estevex-Tapiador and A. Ribagorda, "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol," *LNCS 5379*, in *Proc. of the 9th International Workshop of Information Security Applications*, 2008, pp.56-68.

[26] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda, "LAMED - a PRNG for EPC class-1 generation-2 RFID specification," *Computer Standards & Interfaces*, vol.31, no.1, 2009, pp.88-97.

[27] H.-M. Sun, W.-C. Ting, & K.-H. Wang, "On the Security of Chien's ultralightweight RFID Authentication Protocol," *Cryptology ePrint Archive*, report 83, 2008.

[28] X. Zhang and B. King, "Security requirements for RFID computing systems," *International Journal of Network Security*, vol.6, no.2, 2008, pp.214-226.

Author Biographies

Kuo-Hui Yeh received his B.S. degree in Mathematics from the Fu Jen Catholic University, Taipei County, Taiwan, in 2000, and the M.S. degree in Information Management from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2005. He is currently a Ph.D. candidate in Department of Information Management at the National Taiwan University of Science and Technology. His research interests include RFID applications and security, wireless network protocol and security.

Nai-Wei Lo received his B.S. degree in engineering science from the National Cheng-Kung University, Tainan, Taiwan, in 1988, and the M.S. and Ph.D. degrees in computer science and electrical engineering from the State University of New York at Stony Brook, NY, in 1992 and 1998, respectively. He is currently an assistant professor of Department of Information Management at the National Taiwan University of Science and Technology. His research interests include RFID applications and security, wireless network routing and security, Web technology, and fault tolerance.

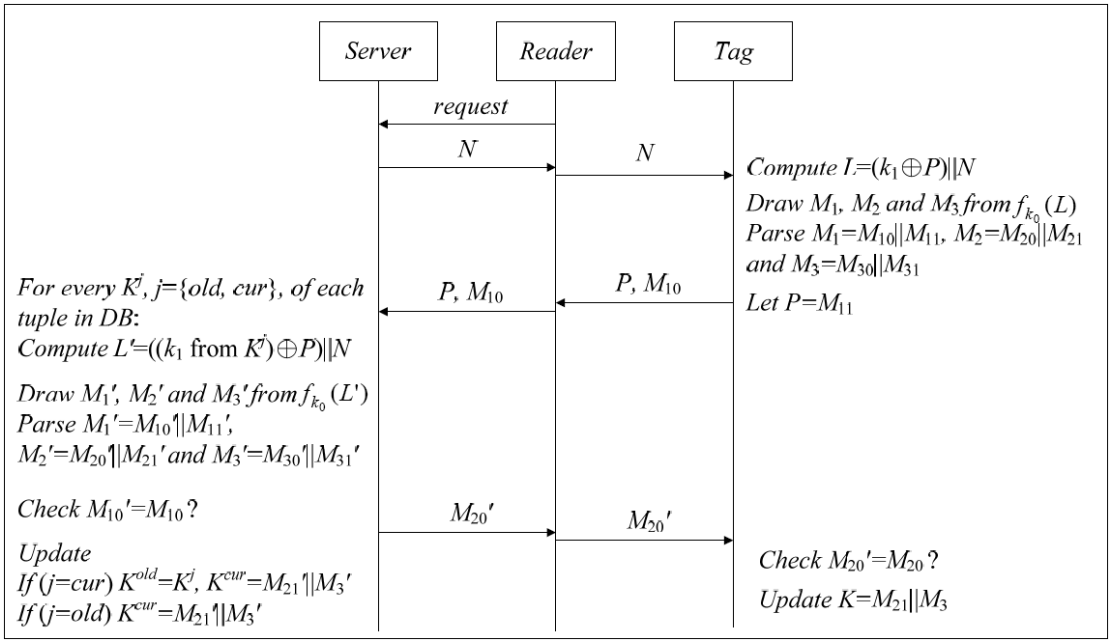

Figure 1. TRAP-3

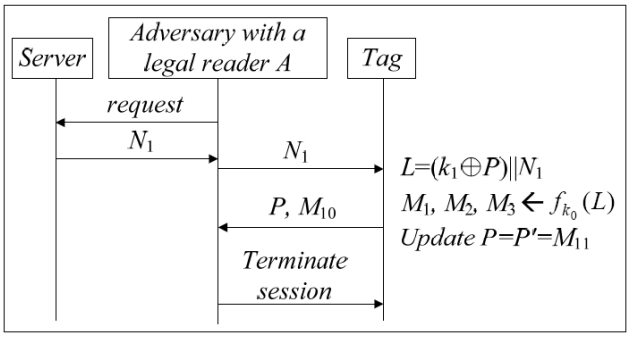

Figure 2. Step 1 of desynchronization attack on TRAP-3

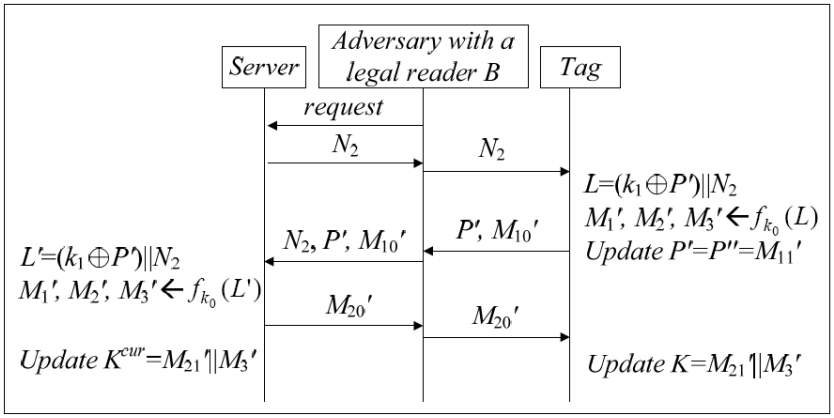

Figure 3. Step 2 of desynchronization attack on TRAP-3

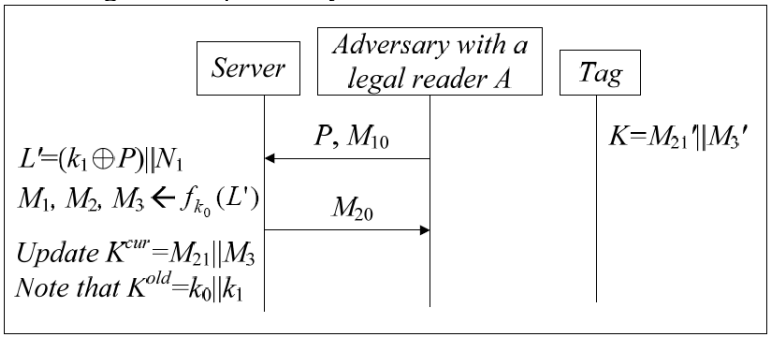

Figure 4. Step 3 of desynchronization attack on TRAP-3