Cryptanalysis of Two RFID Authentication Protocols

Tianjie Cao[1] and Peng Shen[2]
(Corresponding author: Tianjie Cao)

School of Computer, China University of Mining and Technology[1]
Sanhuannanlu, Xuzhou, Jiangsu, 221116, China (Email: tjcao@cumt.edu.cn)
National Mobile Communications Research Laboratory, Southeast University[2]
Sipailou No.2, Nanjing, Jiangsu 210096, China
(Received Mar. 24, 2008; revised and accepted Aug. 22, 2008)

Abstract

Radio frequency identification (RFID) technologies have many advantages in applications such as object tracking and monitoring, ticketing, supply-chain management, contactless payment systems. However, the RFID system may bring about various security and privacy problems. In this paper we present our security analysis of the LAK protocol and the CWH protocol. First, we show that the LAK protocol cannot resist replay attacks, and therefore an adversary can impersonate a legal tag. Next, we present a full-disclosure attack on the CWH protocol. By sending malicious queries to a tag and collecting the response messages emitted by the tag, the full-disclosure attack allows an adversary to extract the secret information from the tag.

Keywords: Authentication, privacy, RFID

1 Introduction

Radio frequency identification (RFID) technology is a major enabler of ubiquitous computing environments which brings enormous productivity benefits in applications such as object tracking and monitoring, ticketing, supply-chain management, contactless payment systems [10]. Aggressive RFID deployments have raised many concerns about security and privacy.

The RFID system has three main components: a set of RFID tags, a set of RFID readers, and a back-end database. An RFID tag is the identification device attached to an object in an RFID system. An RFID reader is a device to communicate with the RFID tag. The RFID reader interrogates the tag, and then transmits the collected data to the back-end database. The Reader can either be handheld terminal or stationary device. The back-end database stores records of product information, tracking logs or key management information associated with RFID tags. After receiving data from the reader the back-end database provides certain services, such as product information etc, to a specific tag. Since the communication between the reader and the tag is performed in an insecure channel, the communicated data can easily be eavesdropped and tampered with by an attacker.

Authentication is an important role in RFID applications for providing security and privacy. Authentication means that an object proves its claimed identity to its communication partner. If an RFID tag tells its own unique identifier information to any RFID readers without any authentication, this will cause the privacy problems, such as spoofing, private information leakage and location tracking of objects. Spoofing means an adversary impersonates a legal tag. Replay attack is a kind of spoofing and allows an adversary impersonate the tag by retransmitting previously transmitted message between a tag and a reader. Information leakage means that the secret information of the object attached a tag can be read by any adversary. Location tracking means that an adversary can track a specific tag attached to an object.

The RFID tags are generally low cost with tightly constrained computational and memory resources, therefore they cannot perform standard cryptographic operations, such as symmetric encryptions and the public key algorithms. To secure the RFID systems, various lightweight RFID protocols have been designed, where mostly hash functions and random number generators are involved. In [6], Lee, Asano and Kim proposed an RFID mutual authentication protocol (the LAK protocol) which utilizes a hash function and synchronized secret information. Similar to Lee et al.'s protocol, Chien and Huang proposed a lightweight RFID authentication protocol based on random number generator [5]. In [2], Chen, Wang and Hwang proposed an ultra-lightweight RFID protocol (the CWH protocol) which only involves simple bitwise XOR operation and left rotate operation. The ultra-lightweight protocols only involve simple bit-wise operations (like XOR, AND, OR, etc.) on tags [2, 3, 4, 7, 8, 11, 12, 13]. However, de-synchronization attack and the full-disclosure attack against such protocols have been reported [4, 7, 8, 11]. The ultra-lightweight strong authentication and strong integrity protocol [3] also has two security vulnerabilities, namely denial-of-service attack and tracing attack based on a compromised tag [1].

In this paper we present our security analysis of the LAK protocol and the CWH protocol. First, we show that the LAK protocol is vulnerable to replay attack, and therefore an adversary can impersonate the tag. The adversary eavesdrops on communication between readers and tags. By eavesdropping, the adversary can copy authentication information and perform replay attack at a later time. Next, we present a full-disclosure attack on the CWH protocol. By sending malicious queries to the tag and collecting the response messages emitted by the tag, this attack allows an adversary to extract the secret information from the tag.

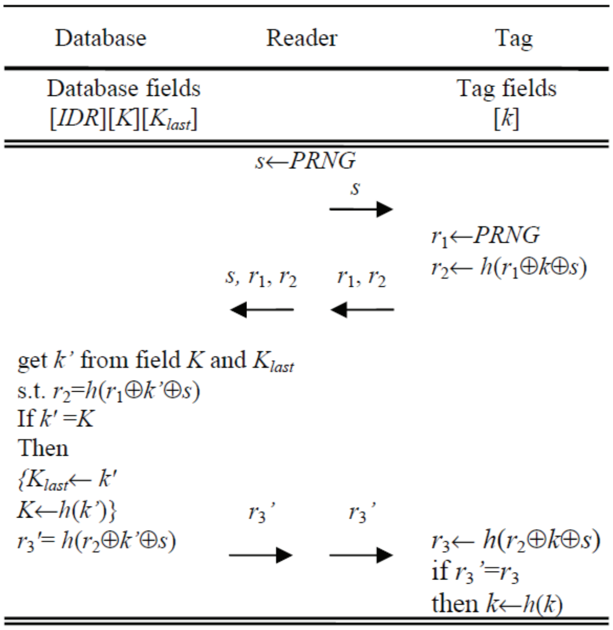

Figure 1: The LAK protocol

2 Notations

To simplify description throughout the paper, we use the following notations.

$Query$:	Requesting the response of the tags.
T:	RFID tag.
R:	RFID reader.
DB:	Back-end database.
$h()$:	One-way hash function.
$PRNG$:	Pseudo Random Number Generator.
$\oplus$:	Exclusive-or (XOR) function.
$\|\|$:	Concatenate function.
$weight(r)$:	$weight(r)$ is the number of binary value "1" of number r.
$Rot(x,y)$:	left rotate operations. $Rot(x,y)$ left rotates the value of x with y bits.

3 Security Analysis of the LAK Protocol

3.1 Review of the LAK Protocol

In the LAK protocol, DB and T can operate the XOR calculation and a common one-way hash function, $h : \{0,1\}^* \rightarrow \{0,1\}^l$. R has a $PRNG$. T also has a $PRNG$, which need not be the same as one of R.

The secret value k whose length is l-bit is saved in non-volatile memory of T. k is used in order to identify ID of T, so k must be different among all T's all the time. The initial value of k of each T is assigned by pre-calculation to guarantee each k of T to be always different.

DB has fields IDR, K, and K_{last}, which save the ID, the current k, the preceding k (the previous secret information which is replaced by the current k), respectively. Initially, IDR and K are set up with ID and initial k of each T, respectively, and all values of the field K_{last} are null. The role of K_{last} is to prevent de-synchronization.

Figure 1 shows the process of the LAK protocol, and the following is a detailed description of each step:

1) R generates a pseudorandom number s by utilizing $PRNG$, and sends s to T.

2) T generates a pseudorandom number r_1 and computes $r_2 = h(r_1 \oplus k \oplus s)$. T sends r_1 and r_2 to R.

3) R delivers responses of T with the value s to DB, i.e., s, r_1, and r_2.

4) In order to find ID of T, DB searches k' from the fields K and K_{last} which satisfies the equation $r_2 = h(r_1 \oplus k' \oplus s)$.

5) DB updates information of T. If k' is found in the field K of a record, k' is copied to the field K_{last} of the record and the field K of the record is set to $h(k')$. If k' is found in the field K_{last}, DB does not update information.

6) DB calculates $r'_3 = h(r_2 \oplus k' \oplus s)$, and sends r'_3 to R. R transfers r'_3 to T.

7) T checks whether or not $r'_3 = h(r_2 \oplus k \oplus s)$. If correct, T updates k to $h(k)$.

3.2 Replay Attack

In this subsection, we show that the LAK protocol is vulnerable to replay attack. In LAK protocol, an adversary can easily eavesdrop on the communications

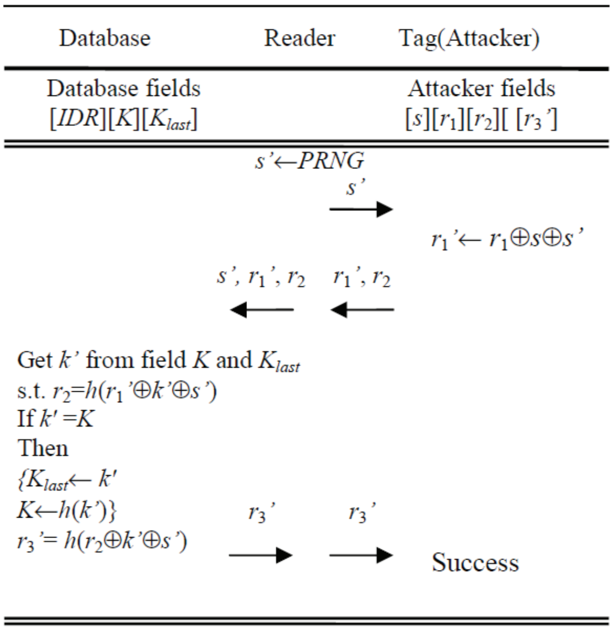

Figure 2: Replay attack

from a legal tag, modify the data, and then replay the messages to masquerade as the legal tag. The replay attack consists of two stages: Copying the messages stage and replaying the messages stage (see Figure 2).

Stage 1. Copying the messages.

Supposing the system is working normally right now. The attacker eavesdrops in the insecure channel, collecting the messages between the reader and the tag.

1) R generates a pseudorandom number s, and sends s to T. The attacker records the message s.

2) T generates a pseudorandom number r_1 and computes $r_2 = h(r_1 \oplus k \oplus s)$. T sends r_1 and r_2 to R. The attacker records the messages r_1 and r_2.

Stage 2. Replaying the messages.

1) R generates a pseudorandom number s' and sends s' to T. The attacker receives the message s'.

2) The attacker computes $r'_1 = r_1 \oplus s \oplus s'$ and sends r'_1 and r_2 to R.

3) R delivers responses with the value s' to DB, i.e., s', r'_1 and r_2.

4) DB searches k' from the fields K and K_{last} which satisfies the equation $r_2 = h(r'_1 \oplus k' \oplus s')$.

5) DB updates information. If k' is found in the field K of a record, k' is copied to the field K_{last} of the record and the field K of the record is set to $h(k')$. If k' is found in the field K_{last}, DB do not update information.

6) DB calculates $r'_3 = h(r_2 \oplus k' \oplus s')$, and sends r'_3 to R. R transfers r'_3 to the attacker.

7) The attacker succeeds in authenticating him to R.

We check the validity of the massage s', r'_1, and r_2. In the first protocol run, DB searches k' from the fields K and K_{last} which satisfies the equation $r_2 = h(r'_1 \oplus k' \oplus s)$. We have $r_2 = h(r_1 \oplus k' \oplus s) = h(r'_1 \oplus k' \oplus s')$. We can see that the reader will finally accept the spoofing tag as the genuine tag.

An important observation of the LAK protocol is that the same reply message r_2 in different sessions can be accepted for any challenge s. Thus, the attacker can replay the message r_2 and disguise as a legal tag. To eliminate this vulnerability, we can modify the structure of $r_2 = h(r_1 \oplus k \oplus s)$ to $r_2 = h(r_1 \oplus k||s)$ in Step 2 of the LAK protocol. If an attacker wants to impersonate a tag in this improved protocol, it must be able to reply a valid response r_2 to the reader's challenge s. However, it is hard to compute such a valid value without knowledge of k.

In [5], Chien and Huang showed the replay attack and the secret disclosure problem of Li et al.'s protocol [9], and then propose a new lightweight protocol to improve the security. However, similar to the analysis of the LAK protocol, Chien and Huang's protocol is also vulnerable to replay attack.

4 Security Analysis of the CWH Protocol

4.1 Review of the CWH Protocol

Recently, Chen et al. proposed an ultra-lightweight RFID authentication (the CWH protocol), where the tags involve only simple bitwise operations like XOR and left rotation operation. The CWH protocol is very efficient and small space storage. Unfortunately, the CWH protocol is vulnerable to full-disclosure attack. For secret value made up of simple bit operation is easily estimated and is inclined to the certain value, the bit operation based protocol is weak to brute force attack.

The goal of the CWH protocol is that the tag stores ReaderlD of an authorized Reader beforehand, thus tags are able to identify authorized readers by their ReaderlD which are stored in both tags and readers. The procedures of the CWH protocol is shown in Figure 3 and described as follows.

1) R generates a 128-bit random number r and computes $s = ReaderID \oplus r$. R broadcasts *query* and s to tag.

2) T receives the s enclosed *query* and recovers the random number r by XOR logic operation with $ReaderID$, which is previously stored in tag. Tag computes $n = weight(r)$, where n is the number of

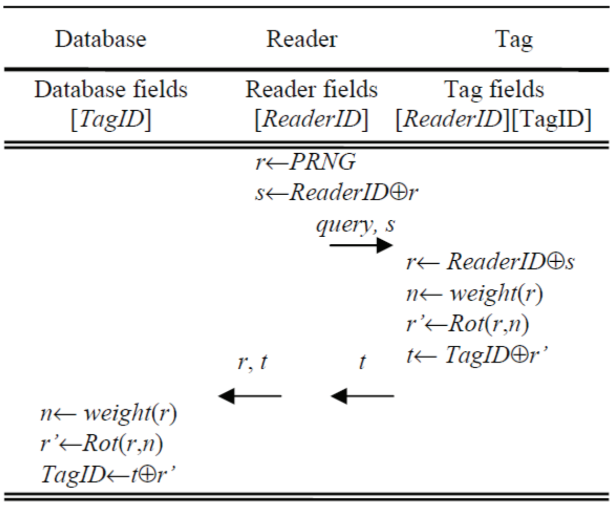

Figure 3: The CWH protocol

binary value "1" of random number r. After that T shifts r to left for n bits generating a new number r'. The tag computes $t = TagID \oplus r'$ and then transmits t to R.

3) R passes the messages t and r to DB. DB computes $n = weight(r)$ and $r' = rot(r, n)$. $TagID$ could finally be found by XOR logic operation of r with t.

4.2 Full Disclosure Attack

We find that the CWH protocol cannot resist the full-disclosure attack. The full-disclosure attack can fully compromise the secret data on tags. The full-disclosure attack on the CWH protocol is described as follows.

1) Supposing the system is working normally right now. The attacker eavesdrops in the insecure channel, collecting the messages s and t when the reader authenticates the tag. We have:

$$\begin{aligned} s &= ReaderID \oplus r && (1)\\ n &= weight(r) \\ r' &= Rot(r, n) \\ t &= TagID \oplus r' \\ &= TagID \oplus Rot(r, n) && (2) \end{aligned}$$

2) The attacker sends $s_1 = s \oplus [I]_0$ to the tag, where $[I]_0 = [000 \dots 001]$ (set the first 127 most significant bits of I as 0 and the least significant bit as 1). The tag responds the message t_1. The attacker records the message t_1. We have:

$$\begin{aligned} r_1 &= ReaderID \oplus s_1 \\ &= ReaderID \oplus s \oplus [I]_0 \\ &= r \oplus [I]_0 \\ n_1 &= weight(r_1) \\ &= weight(r \oplus [I]_0) \\ &= n \pm 1 && (3) \end{aligned}$$

If the least significant bit of r is 0 then the sign in Equation (3) is "+" else "-".

$$\begin{aligned} r'_1 &= Rot(r_1, n_1) = Rot(r \oplus [I]_0, n \pm 1) \\ t_1 &= TagID \oplus r'_1 \\ &= TagID \oplus Rot(r \oplus [I]_0, n \pm 1) && (4) \end{aligned}$$

3) The attacker sends a random number s_2, and records the responding message t_2. We have

$$\begin{aligned} t_2 = {} & TagID \oplus Rot(ReaderID \oplus s_2, \\ & weight(ReaderID \oplus s_2)) && (5) \end{aligned}$$

4) After obtains (s, t) and (s_1, t_1), the attacker can recover all the candidate secrets $ReaderID$ and $TagID$, and then checks the validity through the Equation (5). From Equations (2) and (4), we have

$$\begin{aligned} t \oplus t_1 &= TagID \oplus r' \oplus TagID \oplus r'_1 \\ &= r' \oplus r'_1 \\ &= Rot(r, n) \oplus Rot(r \oplus [I]_0, n \pm 1) && (6) \end{aligned}$$

We give a detailed description about every bit value of r, r', r'_1 and $t \oplus t_1$ in Table 1. In the following description, we omit "mod 128" in subscript.

Now we introduce an algorithm to recover r from Equation (6) and then recover $ReaderID$ and $TagID$ from Equations (1) and (2), then the attacker guesses all possible values of $n = 0, 1, \dots 127$ and checks whether or not the guessed value is correct. The detail full-disclosure attack is described in Table 2.

In our full-disclosure attack, the attacker can interact with the tag two times to attain the responds t, and then the attack can easily derive the secret information $ReaderID$ and $TagID$.

In ultra-lightweight protocols, only readers need to generate pseudo random numbers, tags only use them for creating fresh messages. For some prepared queries, the response messages emitted by the tag may disclose secret information. Thus, it is difficult to prevent information leakage.

5 Conclusions

In this paper we have presented our security analysis of the LAK protocol and the CWH protocol. First, we

Table 1: The values of r, r', r'_1 and $t \oplus t_1$

	127		$n+1$	n	$n-1$		0
r	$r[127]$		$r[n+1]$	$r[n]$	$r[n-1]$		$r[0]$
r'	$r[127-n]$		$r[1]$	$r[0]$	$r[127]$		$r[128-n]$
$r'_1(r[0]=0)$	$r[126-n]$		$r[0] \oplus 1 = 1$	$r[127]$	$r[126]$		$r[127-n]$
$t \oplus t_1(r[0]=0)$	$r[127-n] \oplus r[126-n]$		$r[1] \oplus 1$	$r[0] \oplus r[127]$	$r[127] \oplus r[126]$		$r[128-n] \oplus r[127-n]$
$r'_1(r[0]=1)$	$r[128-n]$		$r[2]$	$r[1]$	$r[0] \oplus 1 = 0$		$r[129-n]$
$t \oplus t_1(r[0]=1)$	$r[127-n] \oplus r[128-n]$		$r[1] \oplus r[2]$	$r[0] \oplus r[1]$	$r[127] \oplus 0$		$r[128-n] \oplus r[129-n]$

Table 2: Disclosing the secret values

//case $r[0] = 0$
for $n = 0$ to 127
{
$r[1] = t[n+1] \oplus t_1[n+1] \oplus 1$
for $i = 0$ to 126
$r[i+2] = t[n+2+i] \oplus t_1[n+2+i] \oplus r[i+1]$
If $n = weight(r)$ and $r[0] = 0$ Then
{$ReaderID = s \oplus r$
$TagID = t \oplus Rot(r, n)$
If $t_2 = TagID \oplus Rot(ReaderID \oplus s_2, weight(ReaderID \oplus s_2))$
Then return $(ReaderID, TagID)$}
}
//case $r[0] = 1$
for $n = 1$ to 128
{
$r[127] = t[n-1] \oplus t_1[n-1]$
for $i = 0$ to 126
$r[126-i] = t[n-2+i] \oplus t_1[n-2+i] \oplus r[127-i]$
If $n = weight(r)$ and $r[0] = 1$ Then
{$ReaderID = s \oplus r$
$TagID = t \oplus Rot(r, n)$
If $t_2 = TagID \oplus Rot(ReaderID \oplus s_2, weight(ReaderID \oplus s_2))$
Then return $(ReaderID, TagID)$}
}

showed that the LAK protocol cannot resist replay attacks, and therefore an adversary can impersonate the tag. Next, we presented a full-disclosure attack on the CWH protocol. By sending malicious queries to the tag and collecting the response messages emitted by the tag, this attack allows an adversary to extract the secret information from the tag. The calculation that involves only simple bitwise operations implies only minor changes will happen in the response if the attacker delicately change few bits of the challenges, which makes the attacker obvious clues to infer the secret information. How to design a secure ultra-lightweight RFID authentication protocol is now an open problem.

Acknowledgments

This work is supported by the Jiangsu Provincial Natural Science Foundation of China (BK2007035), the open research fund of National Mobile Communications Research Laboratory,Southeast University (W200817) and the Science and Technology Foundation of CUMT (0D080309).

References

[1] T. J. Cao, E. Bertino, and H. Lei, "Security analysis of the SASI protocol," *IEEE Transactions on Dependable and Secure Computingy*, Accepted, May 20, 2008.

[2] Y. C. Chen, W. L. WANG, and M. S. Hwang, "Low-cost RFID authentication protocol for anti-counterfeiting and privacy protection," *Asian Journal of Health and Information Sciences*, vol. 1, no. 2, pp. 189-203, 2006.

[3] H. Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 337-340, 2007.

[4] H. Y. Chien and C. W. Huang, "Security of ultra-lightweight RFID authentication protocols and its improvements," *ACM Operating System Review*, vol. 41, no. 2, pp. 83-86, July 2007.

[5] H. Y. Chien and C. W. Huang, "A lightweight RFID protocol using substring," *EUC 2007*, LNCS 4808, pp. 422-431, 2007.

[6] S. Lee, T. Asano, and K. Kim, "RFID mutual authentication scheme based on synchronized secret information," *Proceedings of the SCIS*, http://caislab.icu.ac.kr/Paper/paper_files/2006/SCIS_Lee.pdf, Dec. 11, 2008.

[7] T. Li and R. H. Deng, "Vulnerability analysis of EMAP-An efficient RFID mutual authentication protocol," *Proceedings in Second Int'l Conf. Availability, Reliability, and Security*, pp. 238-245, 2007.

[8] T. Li and G. Wang, "Security analysis of two ultra-lightweight RFID authentication protocols," *IFIP international Federation for Information Processing*, vol. 232, Springer, pp. 109-120, May 2007.

[9] Y. Z. Li, Y. B. Cho, N. K, Um, and S. H. Lee, "Security and privacy on authentication protocol for low-cost RFID," *IEEE International Conference on Computational Intelligence and Security*, vol. 2, pp. 1101-1104, 2006.

[10] D. Lin, H. G. Elmongui, E. Bertino, and B. C. Ooi, "Data management in RFID applications," *International Conference on Database and Expert Systems Applications*, LNCS 4653, pp. 434-444, 2007.

[11] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags," *Proceedings Second Workshop RFID Security*, pp. 137-148, July 2006.

[12] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "EMAP: An Efficient mutual authentication protocol for low-cost RFID tags," *Proceedings OTM Federated Conference and Workshop: IS Workshop*, pp. 352-361, Nov. 2006.

[13] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags," *Proceedings International Conference on Ubiquitous Intelligence and Computing (UIC'06)*, pp. 912-923, 2006.

Tianjie Cao received the BS and MS degree in mathematics from Nankai University, Tianjin, China and the PhD degree in computer software and theory from State Key Laboratory of Information Security of Institute of Software, Chinese Academy of Sciences, Beijing, China. He is a professor of computer science in the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. From 2007 to 2008, he has been a visiting scholar at the Department of Computer Sciences and CERIAS, Purdue University. His research interests are in security protocols and network security.

Peng Shen is currently working toward the Master degree in the School of Computer Science and Technology, China University of Mining and Technology.