REAL UNDERSTANDING OF LPN-PROBLEM-BASED LIGHTWEIGHT AUTHENTICATION PROTOCOLS

Ya-Fen Chang

Department of Computer Science and Information Engineering
National Taichung Institute of Technology, Taichung 404, Taiwan
e-mail: cyf@cs.ccu.edu.tw

Abstract. A family of lightweight authentication protocols, HB-family, has been proposed for low-computation-load-required applications such as radio frequency identification (RFID). Security of this family is based on the learning parity with noise (LPN) problem which has been proven to be an NP-complete problem. But, we find that security of these LPN-problem-based protocols is doubted. We will demonstrate how to cheat the verifier without solving the secret keys with high probability.

Keywords: RFID, the LPN problem, wireless communications, NP-completeness.

1. Introduction

RFID is a popular technology to be utilized in plenty of applications. An RFID system is composed of tags, a reader, and a back-end application system. Data stored in tags will be transmitted to a reader by wireless technologies, and a reader is connected to the back-end system [5]. In such a way, the back-end application system can further use the received information. There are two types of RFID tags: passive tags and active tags. Passive tags receive energy sent by the reader and transfer it to operation power. Passive tags need no batteries and therefore they possess the following advantages: small size, low cost, and low power consumption. Active tags need batteries plugged, and they can send signals to the reader actively and further than passive ones. With RFID, automatic monitoring facilities can be provided by combining database management systems, computer networks and firewall technologies.

Like most wireless technologies, RFID needs to overcome some security problems. As a result, some cryptographic algorithms and protocols for RFID systems were proposed [11, 13-15]. In these protocols, only the legal reader can get information stored in tags so the reader needs to be authenticated by tags. In 2001, Hopper and Blum proposed a light-weight authentication protocol, HB protocol [7]. Unlike previous protocols, HB protocol requires only dot product operation of binary vectors. The computation load of HB protocol is light, and it suits devices with low computation ability such as passive RFID tags. In 2005, Juels and Weis showed that HB protocol could not resist active attacks and proposed a modified version, HB+ protocol [8]. Later, Katz and Shin [9] and Gilbert et al. [6] successfully mounted attacks on HB and HB+ protocols. In 2006, Bringer et al. [4] and Piramuthu [12] proposed modified HB+ protocol to resist previous mentioned attacks. In 2007, Munilla and Peinado proposed HB-MP′ and HB-MP protocols to improve the computation performance of HB+ protocol and to withstand active attacks [10]. They claimed that HB-MP′ protocol was still vulnerable to man-in-the-middle attacks but HB-MP protocol could defend against active attacks.

With deep insight into HB-family, only dot product operation of binary vectors is needed so the computation load is light. Security of HB-family is based on the computational hardness of the learning parity with noise (LPN) problem [2], which has been proven to be an NP-complete problem [1]. However, we find that security of these LPN-problem-based protocols is doubted. We will demonstrate how to cheat the verifier without solving the secret keys with high probability by mounting active attacks on HB-MP protocol.

The remainder of this paper is organized as follows. Section 2 reviews the LPN problem, HB-MP′ protocol, and HB-MP protocol. Section 3 shows the security of HB-family and further discussions. At last, some conclusions are drawn in Section 4.

2. Reviews of related works

The LPN problem, HB-MP′ protocol, and HB-MP protocol are reviewed in Sections 2.1 to 2.3, respectively.

2.1. The LPN problem

In this section, the concept of learning parity without noise and how to find the secret share are first introduced. Then the condition with noise taken into consideration and the LPN problem are presented. For clarity, the used notations are listed as follows:

i: the length of the shared secret;

x: the shared secret, where x is a binary vector of length i;

g_k: binary vectors of length i, where $k\in[1, n]$;

y, z: binary vectors of length n;

v: random noise, where v is a 1-bit value and v = 1 with probability $p\in[0, 1/2]$;

$\oplus$: XOR operation.

y_k denotes the dot product of $x \cdot g_k$ (mod 2), and shorthand $x \cdot g_k$ for $x \cdot g_k$ (mod 2) is used throughout this paper for simplicity. A linear system with binary matrices A, x and y is illustrated in Figure 1, where A is composed of g_1, g_2, …, g_n. When A and y are available and there is no noise, we can solve x by Gaussian elimination.

$$Ax = \begin{vmatrix} g_{11}\, g_{12}\cdots g_{1i} \\ g_{21}\, g_{22}\cdots g_{2i} \\ \vdots \\ g_{n1}\, g_{n2}\cdots g_{ni} \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_\mathrm{i} \end{vmatrix} = y = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{vmatrix}$$

Figure 1. A linear system with binary matrices A, x and y

When noise is taken into consideration, this problem of learning parity is known as the LPN problem, which is an NP-complete problem [1]. In [3], it takes $2^{O(n/\log n)}$ to get x. For given A, x, y and z, we can reformulate these parameters as follows:

$$y_k = x \cdot g_k, \tag{1}$$

$$z_k = y_k \oplus v. \tag{2}$$

According to Equations (1) and (2), we can define the LPN problem as follows:

The LPN problem: for given g_k, z_k and the probability p, recover x.

2.2. A review of HB-MP′ protocol

Munilla and Peinado proposed HB-MP′ protocol composed of q rounds [10]. For clarity, the i-th round is illustrated in Figure 2, where only two messages are exchanged between the reader and a tag. The used notations are listed as follows:

x: the secret key shared between the reader and a tag;

k: the length of x;

a, b: random binary vectors of length k;

v: noise, where v is a 1-bit value and v = 1 with probability $p\in[0, 1/2]$;

$\oplus$: XOR operation;

$a \cdot x$: the dot product of vectors a and x, which is the shorthand for $a \cdot x$ (mod 2).

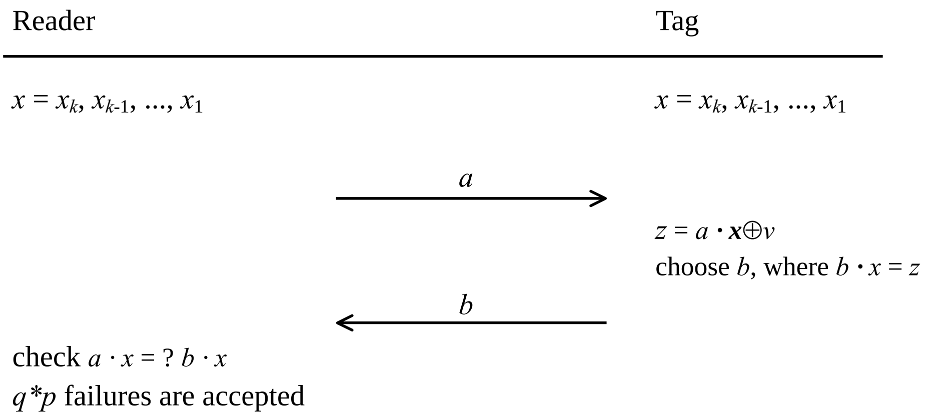
Reader
Tag
$x = x_k, x_{k-1}, ..., x_1$
$x = x_k, x_{k-1}, ..., x_1$
a
$z = a \cdot \mathbf{x} \oplus v$
choose b, where $b \cdot x = z$
b
check $a \cdot x = ?\ b \cdot x$
$q*p$ failures are accepted

Figure 2. The i-th round of HB-MP′ protocol

The details of HB-MP′ protocol are described as follows:

Step 1: The reader chooses one random binary vector a of length k and sends it to the tag.

Step 2: After receiving a, the tag computes $z= a \cdot x\oplus v$ and chooses one k-bit vector b such that $b \cdot x = z$. Then, the tag sends b to the reader.

Step 3: The reader checks if $b \cdot x= a \cdot x$.

After q rounds, the reader accepts the tag if $q*p$ or less rounds to verify b are failed. It can be proven that the problem of finding x by given a and b is at least as difficult as solving the LPN problem [1, 2].

2.3. A review of HB-MP protocol

Although HB-MP′ protocol can resist passive attacks, it is still vulnerable to the same weakness of

HB+ protocol. Thus, Munilla and Peinado proposed HB-MP protocol by modifying HB-MP′ protocol to withstand man-in-the-middle attack [10]. In HB-MP protocol, there are two secret keys shared between the tag and the reader while the length of these shared secret keys does not coincide with that of exchanged messages. The notations used in HB-MP protocol are listed as follows:

x, y: secret keys shared between the reader and the tag;

k: the length of shared secret keys;

m: the length of messages exchanged between the reader and the tag;

xm: an m-bit binary vector, which is the m least significant bits of x;

a, b: random binary vectors of length m;

v: noise, where v is a 1-bit value and v= 1 with probability $p \in [0, 1/2]$;

$\oplus$: XOR operation;

$a \cdot x$: the dot product of vectors a and x, which is the shorthand for $a \cdot x$ (mod 2);

rotate(x, y_k) : a bitwise left rotate operator, which denotes x is left rotated with y_k positions.

HB-MP protocol is also composed of q rounds. For simplicity, the i-th round is illustrated in Figure 3, and the details are showed as follows:

Step 1: The reader first chooses one random binary vector a of length m and sends it to the tag.

Step 2: After receiving a, the tag computes x= rotate(x, y_i), where y_i is the i-th bit of y and computes $z= a \cdot xm \oplus v$. Then, the tag selects an m-bit binary vector b such that $b \cdot xm = z$ and sends b to the reader.

Step 3: The reader computes x = rotate(x, y_i), where y_i is the i-th bit of y, and checks if $a \cdot xm= b \cdot xm$.

Note that the computation result of x = rotate(x, y_i) is only for the i-th round and will not be stored to replace the original x. After q rounds, the reader accepts the tag if $q*p$ or less rounds to verify b are failed.

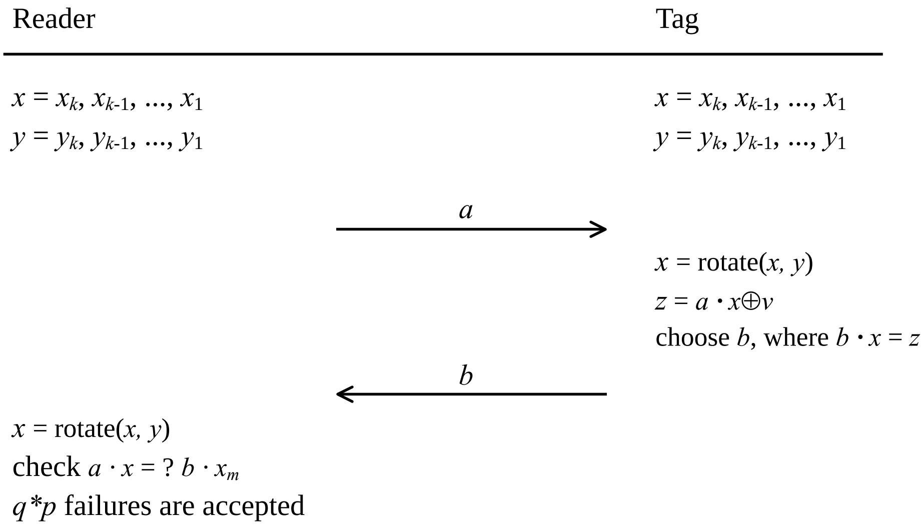
Reader
Tag
$x = x_k, x_{k-1}, ..., x_1$
$y = y_k, y_{k-1}, ..., y_1$
$x = x_k, x_{k-1}, ..., x_1$
$y = y_k, y_{k-1}, ..., y_1$
a
x = rotate(x, y)
$z = a \cdot x \oplus v$
choose b, where $b \cdot x = z$
b
x = rotate(x, y)
check $a \cdot x$ = ? $b \cdot x_m$
$q*p$ failures are accepted

Figure 3. The i-th round of HB-MP protocol

3. Security of HB-family and further discussions

In HB-family, it has been proven that HB, HB+ and HB-MP′ protocols suffer from active attacks or man-in-the-middle attacks. Only HB-MP protocol is claimed to be secure. Unfortunately, we find that HB-MP protocol is still vulnerable to active attacks even if two secret keys are used. In the following, we are going to demonstrate how an attacker cheats the reader with a high probability without knowing what two secret keys are.

In HB-MP protocol, two secret keys x and y are shared between the reader and the tag. In the i-th round, a tag and the reader need to compute x=rotate(x, y_i), where y_i denotes the i-th bit of secret key y. Because y is a binary vector, y_i must be 1 or 0. If y_i=1, it denotes that $xm=x_{m-1}x_{m-2}\ldots x_1x_k$ while $xm=x_mx_{m-1}\ldots x_1$ if y_i=0. Obviously, only $x_mx_{m-1}\ldots x_1x_k$ are involved for authentication though x is of length k. If an attacker tends to cheat the reader, he only needs to get partial information of x by figuring out where two consecutive zeros appear. Attack procedure is as follows:

Step 1: The attacker impersonates a RFID tag and guesses the j-th and (j-1)-th bits are zero, where j-1 = k if j =1.

Step 2: After getting binary vector a sent from the RFID reader, the attacker executes binary-vector-modification algorithm to get b and sends b to the reader. Note that Step 2 will be executed q times.

Step 3: If the attacker is authenticated successfully, x_jx_{j-1}=00 occurs with high probability. The attacker regards x_jx_{j-1}=00.

From now on, the attacker may simply adopt binary-vector-modification algorithm to modify binary vector *a* to cheat the RFID reader.

Binary-vector-modification algorithm

```
Input: binary vector $a= a_ma_{m-1}\ldots a_1$ of length $m$ and position $(j, j-1)$, where $j-1 = k$ if $j = 1$
Output: binary vector $b$ of length $m$

Step 1:  Modify $a_ja_{j-1}$ to be $a_j'a_{j-1}'$.
         Case 1: If $a_ja_{j-1}$=00, $a_j'a_{j-1}'$=10.
         Case 2: If $a_ja_{j-1}$=01, $a_j'a_{j-1}'$=11.
         Case 3: If $a_ja_{j-1}$=10, $a_j'a_{j-1}'$=00.
         Case 4: If $a_ja_{j-1}$=11, $a_j'a_{j-1}'$=01.
Step 2:  $b = a$.
```

For clarity, we are going to demonstrate the spirit of binary-vector-modification algorithm. In binary-vector-modification algorithm, a_ja_{j-1} are modified to be $a_j'a_{j-1}'$ according to a_ja_{j-1}. One of the following two cases will hold.

Case 1: If y_i=0, dot product operation over a_ja_{j-1} will be executed with x_jx_{j-1}, respectively. That is, the difference between the dot product of *xm* and binary vector *a* and that with modified binary vector, *b*, will be x_j.

Case 2: If y_i=1, dot product operation over a_ja_{j-1} will be executed with $x_{j-1}x_{j-2}$., respectively. That is, the difference between the dot product of *xm* and binary vector *a* and that with modified binary vector, *b*, will be x_{j-1}.

After *q* rounds, if the attacker is authenticated successfully, x_jx_{j-1}=00 occurs with high probability. The attacker may regard x_jx_{j-1}=00. Later, the attacker may cheat the RFID reader by simply using binary-vector-modification algorithm to modify binary vector *a*.

Every bit of *x* may be 0 or 1 with probability 1/2. One interesting question occurs: whether there are no two consecutive zeros appearing? For this interesting question, we need to further analyze HB-MP protocol. Note that it is determined that two consecutive zeros appear if the last and the first bits are zero. The used symbols are listed as follows:

S_j: the set of all binary vectors of length *j* which contain no two consecutive zeros;

$S_{j,01}$: the set of all binary vectors of length *j* which contain no two consecutive zeros while MSB (most significant bit, MSB) is 0 and LSB (least significant bit, LSB) is 1;

$S_{j,10}$: the set of all binary vectors of length *j* which contain no two consecutive zeros while MSB is 1 and LSB is 0;

$S_{j,11}$: the set of all binary vectors of length *j* which contain no two consecutive zeros while MSB is 1 and LSB is 1;

We have $S_j = S_{j,01} \cup S_{j,10} \cup S_{j,11}$.

$S_{j,01} = \{A1\} \cup \{B01\}$, where $A \in S_{j-1,01}$ and $B \in S_{j-2,01}$.

$S_{j,10} = \{A0\}$, where $A \in S_{j-1,11} = \{B10\} \cup \{C10\}$, where $B \in S_{j-2,10}$ and $C \in S_{j-2,11}$.

$S_{j,11} = \{A1\} \cup \{B1\}$, where $A \in S_{j-1,11}$ and $B \in S_{j-1,10}$.

That is,

$|S_{j,01}| = |S_{j-1,01}| + |S_{j-2,01}|$.

$|S_{j,10}| = |S_{j-2,10}| + |S_{j-2,11}|$.

$|S_{j,11}| = |S_{j-1,11}| + |S_{j-1,10}|$.

$|S_j| = |S_{j,01}| + |S_{j,10}| + |S_{j,11}| = |S_{j-1,01}| + |S_{j-2,01}| + |S_{j-2,10}| + |S_{j-2,11}| + |S_{j-1,11}| + |S_{j-1,10}| = |S_{j-1}| + |S_{j-2}|$.

Let $F(n) = |S_j|$, we have $F(n)=F(n-1)+F(n-2)$, where $F(2)=3$ and $F(3)=4$. Munilla and Peinado suggest that the length of *x* may be 64-bit. According to the above equation, the number of 64-bit binary vectors containing no two consecutive zeros is $2.372515*10^{13}$, and the probability will be $1.286143*10^{-6}$. As a result, the probability to find two consecutive zeros is high. According to the above analyses, HB-MP protocol is still vulnerable to active attacks even two secret keys *x* and *y* are shared between the reader and the tag.

Why these LPN-problem-based authentication protocols are insecure even though the LPN problem is an NP-complete problem? It is because the dot product of two binary vectors will be either 0 or 1. All the attacker has to do is finding the position of zero in HB, HB+ and HB-MP′ protocols and the position of two consecutive zeros in HB-MP protocol.

4. Conclusions

Though the LPN problem has been proven to be an NP-complete problem, these LPN-problem-based authentication protocols are insecure. It is because the computation results for authentication will be either 0 or 1. All the attacker has to do is finding the position of one zero or two consecutive zeros. In our opinion, the concept of HB-MP protocol may be adopted in such a way that the RFID reader and a tag still share two secrets *x* and *y*. But, *x* for the *i*-th round should not be computed by *x*=rotate(*x*, y_i). We may use different operations to have *x* vary in different rounds -- $P_i = \sum_{j=1}^{i} y_j$ and *x*=rotate(*x*, P_i) in the *i*-th round for example. By simple modification, it is hard for the attacker to find the position of zeros, and the security of the LPN problem can be ensured.

Acknowledgement

This work was supported in part by National Science Council under the grant NSC 98-2221-E-025-007-.

References

[1] **E.R. Berlekamp, R.J. McEliece, H.C.A. van Tillborg.** On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, *Vol*. 24, 1978, 384-386.

[2] **A. Blum, M.L. Furst, M.J. Kearns, R.J. Lipton.** Cryptographic primitives based on hard learning problems. *Advances in Cryptology – CRYPTO*'93, *Lecture Notes in Computer Science*, *Vol*. 773, Springer, 1994, 278-291.

[3] **A. Blum, A. Kalai, H. Wasserman.** Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, *Vol*. 50, *No*. 4, *July* 2003, 506-519.

[4] **J. Bringer, H. Chabanne, E. Dottax.** HB++: a lightweight authentication protocol secure against some attacks. *Proceedings of IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing- SecPerU*'06, 2006, 28-33.

[5] **K. Finkenzeller.** RFID Handbook. *Second edition, Wiley & Sons, UK*, 2002.

[6] **H. Gilbert, M. Robshaw, H. Silbert.** An active attack against HB+- a provable secure lightweight authentication protocol. *Cryptology ePrint Archive, Report* 2005/237, 2005, *http://eprint.iacr.org*.

[7] **N.J. Hopper, M. Blum.** Secure human identification protocols. *Advances in Cryptology - ASYACRYPT'* 2001, *Lecture Notes in Computer Science*, *Vol*. 2248, *Springer*, 2001, 52-66.

[8] **A. Juels, S. Weis.** Authenticating pervasive devices with human protocols. *Advances in Cryptology – Crypto*2005, *Lecture Notes in Computer Science*, *Vol*. 3621, *Springer*, 2005, 293-308.

[9] **J. Katz, J.S. Shin.** Parallel and concurrent security of the HB and HB+ protocols. *Cryptology ePrint archive, Report* 2005/461, 2005, *http://eprint.iacr.org*.

[10] **J. Munilla, A. Peinado.** HB-MP: a further step in the HB-family of lightweight authentication protocols. *Computer Networks*, *Vol*. 51, 2007, 2262-2267.

[11] **M. Ohkubo, K. Suzuki, S. Kinoshita.** Efficient hash-chain based RFID privacy protection scheme. *Proceedings of Ubiquitous Computing*, *September* 2004.

[12] **S. Piramuthu.** HB and related lightweight authentication protocols for secure RFID tag/reader authentication. *Proceedings of CollECTeR Europe Conference, Basel*, *Switzerland, June* 2006.

[13] **S.E. Sarma, S.A. Weis, D.W. Engels.** RFID systems and security and privacy implications. *Workshop on Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, *Vol*. 2523, *Springer*, 2002, 454-469.

[14] **I. Vajda, L. Buttyan.** Lightweight authentication protocols for low-cost RFID tags. *Proceedings of Ubiquitous Computin*g, 2003.

[15] **S.A. Weis, S.E. Sarma, R.L. Rivest, D.W. Engels.** Security and privacy aspects of low-cost radio frequency identification systems. *Security in Pervasive Computing, Lecture Notes in Computer Science*, *Vol*. 2802, *Springer*, 2004, 201-212.

Received February 2010.

DOI: 10.5755/j01.itc.39.3.12304