Security Analysis of Lightweight Authentication Protocol from WISTP 2013

Wang Shao-Hui[1,2,3], Xiao Fu[1,2], Chen Dan-wei[1,2], Wang Ru-chuan[1,2]

[1)](College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

[2)](Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China)

[3)](Network and Data Security Key Laboratory of Sichuan Province, Chengdu 610054, China)

Abstract One of the key problems in Radio Frequency Identification (RFID) is security and privacy. Many RFID authentication protocols have been proposed to preserve security and privacy of the system. Nevertheless, most of these protocols are analyzed and it is shown that they can not provide security against some RFID attacks. In WISTP 2013, a new lightweight authentication protocol using AES S-box and some special function is presented. The new protocol has a good implementation in resource constrained tags. In this paper, we give the security analysis on this new authentication protocol. After impersonating the valid reader to query the tag and collecting the responses, we can deduce all the secrets shared between the reader and tag through analyzing the messages. The attack utilizes the structure of the invertible function and the property of the special function introduced in the new protocol.

Key words RFID; Lightweight Authentication; Privacy; Active Attack

1 Introduction

Radio Frequency Identification (RFID) systems are used for automated identification of objects and people. Applications that use RFID technology include warehouse management, logistics, railroad car tracking, product identification, library books check-in/check-out, asset tracking, passport and credit cards, etc. Privacy and security is one of the key problems in RFID, because the communication between the reader and the tag are more vulnerable to malicious adversaries. The possible security threats to RFID systems include denial of service (DoS), man in the middle (MIM), counterfeiting, spoofing, eavesdropping, traffic analysis, traceability, de-synchronization etc.

An effective and flexible way to assure privacy and security is to adopt authentication protocols. The low cost deployment demand for RFID tags forces the lack of resources for performing true cryptographic operations to provide security. It is worthwhile to study ultra-lightweight authentication protocols which require tags to involve only simple bitwise operations such as bitwise XOR, bitwise OR, bitwise AND and rotation.

However, providing lightweight security in RFID systems is not a trivial task. Several ultra-lightweight protocols have already been proposed. However, they all have certain flaws and vulnerabilities. Vajda and Buttyan [1] have proposed a set of extremely lightweight challenge response authentication algorithms. These can be used for authenticating the tags, but they may be easily attacked by a powerful adversary. Juels [2] proposed a solution based on the use of pseudonyms, without using any hash function. But after a set of authentication sessions, the list of pseudonyms will need to be reused or updated through an out-of-band channel, which limits the practicality of this scheme.

A family of ultralightweight mutual authentication protocols have been proposed, but later it

was reported that these protocols are vulnerable to variable attacks, such as passive attack, desynchronization attack and full-disclosure attack[3-13]. One of the reasons for the vulnerabilities of these protocols is the imbalance of some triangular operations like bitwise OR and bitwise AND, and the operations used can not provide good cryptographic property.

Recently, Dusart and Traore proposed a lightweight authentication protocol for Low-Cost RFID Tags(named D-T protocol) in the conference of WISTP 2013[14]. The protocol utilizes the security qualities of the AES S-Boxes[15] to build a function, and the authors claim the new protocol can provide good Strict Avalanche property, and can resist many passive or active attacks. In this paper, we examine the security of D-T protocol. An active attack on D-T protocol is proposed, in which we first collect some authentication messages through impersonating valid reader to query the tag; then using the property of the f function introduced in D-T protocol, we can obtain all the secret key bytes. The number we need to query the tag is less than 2048, so we conclude D-T can not provide privacy guarantee as claimed.

The rest of the paper is organized as follows: We describe the D-T protocol in section 2, and the detail security analysis of D-T is presented in section 3, and show how to extract all the secrets shared by the reader and the tag; Conclusion is given in section 4.

2 Description of D-T Protocol

D-T protocol utilizes the challenge-response authentication model, which uses a non-invertible function h. In D-T protocol, to verify the identity of the tag, the reader *R* and the tag *T* shares secret key K. D-T protocol is proceeded as follows:

Step 1. The reader *R* generates and sends the challenge $C = (C_0, C_1, ..., C_{15})$ to the tag *T*, where C_i are bytes randomly chosen.

Step 2. The tag computes and sends $t_1 = ID \oplus h_K(C)$, $t_2 = h_{ID}(C)$, where ID is the identity of the tag.

Step 3. The reader polls all the secrets K to compute $h_K(C)$, and retrieves the identity of the tag. Finally authentication of the tag can be verified by checking $h_{ID}(C)$.

To provide mutual authentication, D-T protocol can be adapted with a slightly modification: In Step 2, a challenge C' is sent with the tag's response, and the reader should respond with the computation of $h_{K \oplus C'}(C \oplus ID)$.

Description of function h. Function h is composed of f Function and S function, and S function is chosen as the AES[15] SubBytes function**.** Let $M = \{M_0, M_1, ..., M_{15}\}$ be a 16-byte vector, then $S(M) = (\text{SubBytes}(M_0), \text{SubBytes}(M_1), ..., \text{SubBytes}(M_{15}))$.

f function takes two input bytes, and produces one byte output $f: F_{256} \times F_{256} \to F_{256}$:

$$f(x, y) = [[x \oplus ((255 - y) << 1)] + 16 \cdot [((255 - x) \oplus (y >> 1)) \bmod 16]] \bmod 256$$

where $\oplus$ is the bitwise Exclusive Or, + represents the classical integer addition, $n >> 1$ divides *n* by 2, $n << 1$ multiplies *n* by 2.

Let $i \in \{0,1,...,15\}$ be a vector index and $j \in \{1,2,3,4\}$ a round index, and $M = \{M_0, M_1, ..., M_{15}\}$ is a vector of 16 bytes. The following functions is defined:

$$F_i^j(M) = f(M_i, M_{(i+2^{j-1}) \bmod 16}) \quad \text{and} \quad F^j(M) = (F_0^j(M), F_1^j(M), ..., F_{15}^j(M))$$

To compute $h_K(C)$, First the challenge C is Xored to the key K, and we can obtain $D = C \oplus K = (C_0 \oplus K_0, ..., C_{15} \oplus K_{15})$. The first state M^0 is initialized by $M^0 = S(D)$. Then, the following values are calculated:

$$M^1 = S(F^1(M^0)) \oplus K, \quad M^2 = S(F^2(M^1)) \oplus K,$$
$$M^3 = S(F^3(M^2)) \oplus K, \quad M^4 = S(F^4(M^3)) \oplus K.$$

Finally, the function $h_K(C)$ is denoted as $h_K(C) = M^4 = (M_0^4, M_1^4, ..., M_{15}^4)$.

3 Security Analysis of D-T Protocol

Suppose the secret key is $K = (K_0, K_1, ..., K_{15})$, in this section we give a security analysis of the D-T protocol and show how to deduce the secret key from $h_K(C)$.

3.1 Analysis of the D-T Protocol

We first give some observations about the f function and S function.

Observation 1. As to f function, given y, $f(x_0, y) \neq f(x_1, y)$ for any different x_0 and x_1; while given x and y, there exists a unique y' satisfying $f(x, y) = f(x, y')$.

Observation 2. As to S function, SubBytes is a bijective function mapping F_{256} to F_{256}.

From the description of D-T protocol, we know M_0^4 is calculated as figure 1 shows. Here M_{2k}^1 is obtained from M_{2k}^0 and M_{2k+1}^0 ($k = 0,1,...,7$):

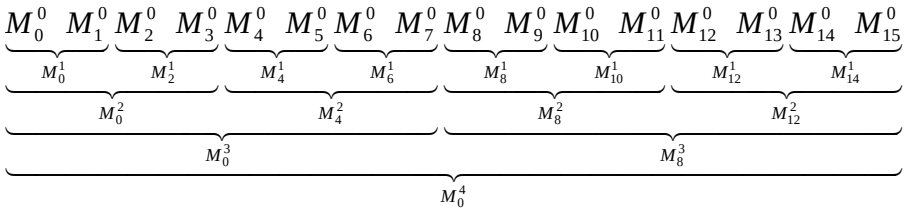

Fig 1. calculation of M_0^4

From the above observation, we can obtain the following conclusion:

Conclusion 1. Given the challenge $C=(C_0,C_1,...,C_{15})$, we denote $h_K(C)$ as $(M_{C,0}^4,M_{C,1}^4,...,M_{C,15}^4)$. Then we poll the second byte C_1 over the field F_{256}, and the other bytes unchanged, we can get another unique challenge $C^*=(C_0,C_1^*,...,C_{15})$ satisfying $M_{C^*,0}^4=M_{C,0}^4$.

Proof. From the observation 1 and observation 2, we know there must exist C_1^* satisfying $M_{C^*,0}^1=M_{C,0}^1$. Combined with the calculation of M_0^4, we know $M_{C^*,0}^4=M_{C,0}^4$. In addition, we know the value C_1^* is unique, because if $M_{C^*,0}^1\neq M_{C,0}^1$, we also know $M_{C^*,0}^2\neq M_{C,0}^2$ and $M_{C^*,0}^3\neq M_{C,0}^3$ because of observation 1, thus $M_{C^*,0}^4\neq M_{C,0}^4$.

Using the conclusion 1, we present the following algorithm to deduce the second key byte K_1 from $h_K(C)$.

Algorithm 1 Recovery of the second key byte K_1

Step 1. A random challenge $C=(C_0,C_1,...,C_{15})$ is chosen, and we obtain $(M_{C,0}^4,M_{C,1}^4,...,M_{C,15}^4)$

Step 2. Try all the other challenge $C'=(C_0,C_1',...,C_{15})$, where the second value C_1' is polled from 0 to 255, and the other values are the same as C, this step will stop if we find C_1^* satisfying $M_{C^*,0}^4\neq M_{C,0}^4$. Now we get (C_0,C_1) and (C_0,C_1^*), which have $M_{C^*,0}^1=M_{C,0}^1$.

Step 3. we try all the possible K_0 and K_1, and output the pairs satisfying:

$$f(\text{SubBytes}(K_0\oplus C_0),\text{SubBytes}(K_1\oplus C_1))=f(\text{SubBytes}(K_0\oplus C_0),\text{SubBytes}(K_1\oplus C_1^*))$$

Step 4. Usually, there are many possible outputs of the Step 3, we can change the first and second bytes of the challenge C in Step 1, then proceed Step 2 and Step 3 to filter out the wrong possible values.

3.2 Experiment Results and complexity analysis

We give an experiment as appendix shows, here $K_0 = 0xa7,\ K_1 = 0x92$; we first take $C_0 = 0x37$ and $C_1 = 0xb7,$ and we can get another $C_1^* = 102$. The number of the possible pairs (K_0, K_1) is 512, and possible values of K_1 are 67 and 146; Then we change the first and second byte of challenge as $C_0 = 0xa6$ and $C_1 = 0x79$, and $C_1^* = 118$, and possible values of K_1 are 157 and 146. So we can deduce the value of K_1 is 146.

In our attack, we can not obtain the value of the first secret byte, and the number of the possible value of the second byte is always 2. So after trying the algorithm 1 two times, we can obtain the value of K_1. Using the same method, we can deduce all the bytes of the secret K.

When attacking the RFID D-T authentication protocol using algorithm 1, the attack complexity is coming from the number to query the tag. As the experiment shows, to obtain K_1, we query the tag about $256*2 = 2^9$ times, so we almost have to query the tag $2^9 *16 = 2^{13}$ times to obtain all the secret. To reduce the query number, we can get the remaining 32 bits in brute force way after obtaining 12 bytes secrets, and the query number is about $2^9 *12 = 6144$.

In fact, we can reduce the query number as the way of birthday attack. We query the tag challenge $C = (C_0, C_1, ..., C_{15})$, where C_1 is randomly chosen and the other values are constant. From the birthday attack, we know after query 130 challenges, we must have at least 2 pairs of $\{C^1 = (C_0, C_1^1, ..., C_{15}), C^2 = (C_0, C_1^2, ..., C_{15})\}$ and $\{C^3 = (C_0, C_1^1, ..., C_{15}), C^4 = (C_0, C_1^2, ..., C_{15})\}$ satisfying what we need, then the total number we need is about $2^7 *16 = 2^{11}$ (130*12=1560 if brute force attack considered).

4 Conclusion

In this paper, we give the active attack on the new proposed lightweight authentication protocol at WISTP 2013. We first impersonate the valid reader to query the tag, and send some special challenges and collect the corresponding responses. Utilizing the structure of the h

function and the property of the f function, we can deduce all the secrets shared between the reader and the tag. The attack complexity is to query the tag about 2^{11} times, and in the real attack, the number is much shorter than this value. So we conclude the new protocol is not secure enough to use in reality. How to reduce the analysis complexity and analyze the protocol theoretically will be considered in the future work.

Acknowledgments

This work is supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions(PAPD), National Natural Science Funds (Grant No.60903181) and Nanjing University of Post and Telecommunication Funds (Grant No. NY211064).

References

[1] Vajda I, Buttyan, L (2003) Lightweight authentication protocols for low-cost RFID tags. In: the Second Workshop on Security in Ubiquitous Computing. Seattle, Washington.

[2] Juels, A (2005) Minimalist Cryptography for Low-Cost RFID Tags (Extended Abstract). In: Fourth Conference on Security in Communication Network. Amalfi.

[3] Peris-Lopez P, Hernandez-Castro JC, Tapiador JME, Ribagorda A (2006) LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags. In: Workshop on RFID Security 2006. Graz, Austria.

[4] Peris-Lopez P, Hernandez-Castro JC, Tapiador JME, Ribagorda A (2006) M2AP: a minimalist mutual-authentication protocol for lowcost RFID tags. In: 2006 International Conference on Ubiquitous Intelligence and Computing. Wuhan, China.

[5] Chien HY (2007) SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. IEEE Transactions on Dependable and Secure Computing 4(4): 337–340.

[6] Sadighian A, Jalili, R (2009) AFMAP: Anonymous forward-secure mutual authentication protocols for RFID systems. In: Third IEEE International Conference on Emerging Security Information, Systems and Technologies. Athens, Glyfada.

[7] Sadighian A, Jalili, R (2008) FLMAP: A fast lightweight mutual authentication protocol for RFID systems. In: 16th IEEE International Conference on Networks. New Delhi, India.

[8] Peris-Lopez P, Hernandez-Castro JC, Tapiador JME, Ribagorda A (2008) Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In: 9th International Workshop on Information Security Applications. Jeju Island, Korea.

[9] Li T, Wang G (2007) Security analysis of two ultra-lightweight RFID authentication protocols. In: 22nd International Information Security Conference. Sandton, South Africa.

[10] Safkhani M, Naderi M, Bagher N (2010) Cryptanalysis of AFMAP. IEICE Electronics Express 7(17): 1240–1245.

[11] Bárász M, Boros B, Ligeti P, Lója K, Nagy D (2007) Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags. In: First International EURASIP Workshop on RFID Technology. Vienna, Austria.

[12] Phan RCW (2009) Cryptanalysis of a new ultralightweight RFID authenticaion protocol—SASI. IEEE Transactions on Dependable and Secure Computing 6(4): 316–320.

[13] Avoine G, Carpent X, Martin B (2010) Strong authentication and strong integrity (SASI) is not that strong. In: Workshop on RFID Security - RFIDSec'10. Istanbul.

[14] Dusart P, Traore s . Lightweight Authentication Protocolfor Low-Cost RFID Tags. L. Cavallaro and D. Gollmann (Eds.): WISTP 2013, LNCS 7886, pp. 129–144, 2013.

[15] Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002).

Appendix.

```
#include<stdio.h>
void main()
{
```

```
int x,y,z1,z2,z11,z12,z,z0,x1,y1,k0,k1,i,j,k;
k0=0xa7;
k1=0x92;
x=0x37;
y=0xb7;
int number=0;

int sbox[256] = {
//0     1     2     3     4     5     6   7     8     9     A     B   C   D     E     F
0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76, //0
0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0, //1
0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15, //2
0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75, //3
0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84, //4
0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf, //5
0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8, //6
0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2, //7
0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73, //8
0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb, //9
0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79, //A
0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08, //B
0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, //C
0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e, //D
0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, //E
0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 }; //F

x=sbox[x^k0];
y=sbox[y^k1];

z1=x^((255-y)<<1);
z2=16*(((255-x)^(y>>1))%16);
z=(z1+z2)%256;
printf("%d,",z);

for(i=0;i<256;i++)
{
    y=sbox[i^k1];
    z11=x^((255-y)<<1);
   z12=16*(((255-x)^(y>>1))%16);
   z0=(z11+z12)%256;
    if(z0==z)
    {
        printf("%d,",i);
    }
```

```
    }
    printf("\n");

    for(j=0;j<256;j++)
    {
        for(k=0;k<256;k++)
        {
            x=sbox[0x37^j];
            y=sbox[102^k];
            y1=sbox[183^k];

            z1=x^((255-y)<<1);
            z2=16*(((255-x)^(y>>1))%16);
            z=(z1+z2)%256;

            z11=x^((255-y1)<<1);
            z12=16*(((255-x)^(y1>>1))%16);
            z0=(z11+z12)%256;

            if(z0==z)
            {
                printf("%d,%d      ",j,k);
                number++;
            }
        }
    }
    printf("\n%d,",number);
}
```