J Supercomput (2018) 74:65–70
https://doi.org/10.1007/s11227-017-2105-8

On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags

King-Hang Wang[1] · **Chien-Ming Chen**[2] ·
Weicheng Fang[2] · **Tsu-Yang Wu**[3,4]

Abstract Recently, Tewari and Gupta proposed a ultra-lightweight mutual authentication protocol in IoT environments for RFID tags. Their protocol aims to provide secure communication with least cost in both storage and computation. Unfortunately, in this paper, we exploit the vulnerability of this protocol. In this attack, an attacker can obtain the key shared between a back-end database server and a tag. We also explore the possibility in patching the system with some modifications.

Keywords RFID · IoT · Mutual authentication · Cryptanalysis

✉ Chien-Ming Chen
chienming.taiwan@gmail.com

King-Hang Wang
kevinw@cse.ust.hk

Weicheng Fang
626558837@qq.com

Tsu-Yang Wu
wutsuyang@gmail.com

[1] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

[2] Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China

[3] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China

[4] National Demonstration Center for Experimental Electronic Information and Electronic Technology Education, Fujian University of Technology, Fuzhou, China

1 Introduction

Internet of things (IoT) [1] is the network of physical devices which contain embedded technology such as sensors, RFID, and network connectivity to communicate with other devices or external environment. The phrase IoT was firstly proposed by MIT Auto-ID Center in 1999 [10]. In 2005, International Telecommunications Union [11] further pointed out that RFID technology, sensor technology, nanotechnology, and intelligent embedded technology are the fore main technologies to realize IoT. IoT has now become a hot topic and attracted great attention from the computer science literature. Depending on different application requirements, IoT has been rapidly extended, and new technologies have been involved in it.

In the recent years, the science community has put emphasis on the security of IoT since security issues are important to assure reliable interactions among devices [2,4,5,9]. Very recently, Tewari and Gupta [12] proposed a ultra-lightweight mutual authentication protocol for IoT devices using RFID tags in 2016. This protocol is very efficient since it only utilizes bitwise operations. The authors also provided a detailed analysis to demonstrate this protocol is secure against various attacks. However, in this paper, we still find this protocol is still vulnerable to a key disclosure attack. At the end of this paper, we patched their protocol with a simple fix and discussed the security issues of our amendment.

2 Review of Tewari and Gupta's protocol

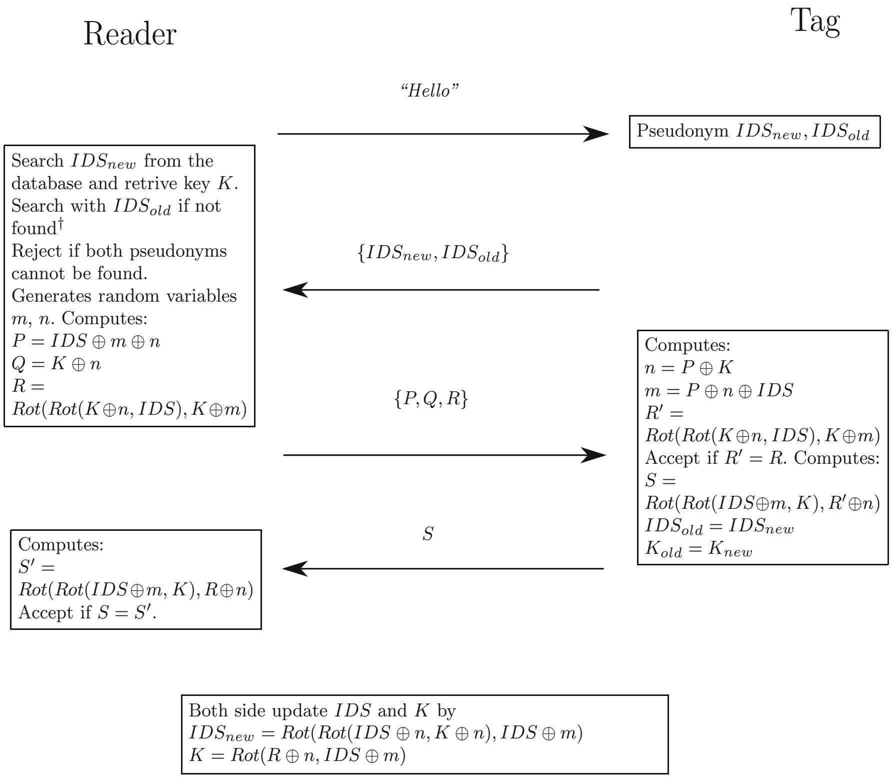
In this section, we briefly review Tewari and Gupta's protocol [12]. This protocol utilizes two kinds of bitwise operation.

- *Bitwise XOR operation* The bitwise XOR operation $\oplus$ denotes bitwise addition modulo 2.
- *Bitwise rotation operation* The bitwise rotation $Rot(X, Y)$ rotates X left by $wt(Y)$ bits. Note that $wt(Y)$ means the Hamming weight of Y which the number of 1's in Y.

There are three entities involved: a back-end server, a reader, and various tags. Each tag shares a secret key K and a pseudonym IDS with the server. If the authentication proceeds successfully, K and IDS are updated. In case the update is interrupted, the server and the tag also backup them as $\{K_{old}, IDS_{old}\}$ before updating.

The procedures of this protocol are illustrated in Fig. 1.

- *Step* 1. A reader sends "$Hello$" to a tag to initiate a new authentication session.
- *Step* 2. The tag sends $\{IDS, IDS_{old}\}$ to the reader.
- *Step* 3. The reader searches IDS from the back-end server's database and retrieves the key K of this tag. Then, the reader generates two nonces m and n, and sends a challenge $\{P, Q, R\}$ to the tag, where $P = IDS \oplus m \oplus n$, $Q = K \oplus n$, and $R = Rot(Rot(K \oplus n, IDS), K \oplus m)$.
- *Step* 4. The tag derives m and n and calculates R' with m and n. If R' equals to R, the tag sends a response S back, where $S = Rot(Rot(IDS \oplus n), R' \oplus m)$. The tag also backups the key and the pseudonym as $IDS_{old} = IDS$ and $K_{old} = K$,



Pseudonym IDS_{new}, IDS_{old}

Search IDS_{new} from the database and retrive key K. Search with IDS_{old} if not found†
Reject if both pseudonyms cannot be found.
Generates random variables m, n. Computes:
$P = IDS \oplus m \oplus n$
$Q = K \oplus n$
$R = Rot(Rot(K \oplus n, IDS), K \oplus m)$

Computes:
$n = P \oplus K$
$m = P \oplus n \oplus IDS$
$R' = Rot(Rot(K \oplus n, IDS), K \oplus m)$
Accept if $R' = R$. Computes:
$S = Rot(Rot(IDS \oplus m, K), R' \oplus n)$
$IDS_{old} = IDS_{new}$
$K_{old} = K_{new}$

Computes:
$S' = Rot(Rot(IDS \oplus m, K), R \oplus n)$
Accept if $S = S'$.

Both side update IDS and K by
$IDS_{new} = Rot(Rot(IDS \oplus n, K \oplus n), IDS \oplus m)$
$K = Rot(R \oplus n, IDS \oplus m)$

Fig. 1 Illustration of Tewari and Gupta’s protocol. †In case Tag’s IDS_{old} match with the reader’s IDS_{new}, it means that the last updating process at the server side was not success, and the communication will be continued using the tag’s IDS_{old} and K_{old}

and updates them as $IDS = Rot(Rot(IDS_{old} \oplus n, K_{old} \oplus n), IDS_{old} \oplus m)$ and $K = Rot(R \oplus n, IDS_{old} \oplus m)$.

- *Step* 5. The reader verifies the value S. If it holds, the reader backups and updates the secrets in the same way as shown in *Step* 4.

3 Attack

In this section, we demonstrate that Tewari and Gupta’s protocol [12] is vulnerable to a key disclosure attack. In their protocol, S is a rotation of $IDS \oplus n$. It means that there are only 96 possible rotation operations. It gives an adversary $\mathcal{A}$ an opportunity to obtain K.

To launch a key disclosure attack, an adversary $\mathcal{A}$ acts in a passive mode where only eavesdropping is allowed and follows the steps described in Algorithm 1. In the beginning of the algorithm, $\mathcal{A}$ inputs whatever sent by the reader or the tag during an actual protocol run. The algorithm will output a list of possible K, alone with corresponding nonces m and n. After that, $\mathcal{A}$ can verify each possible K by checking the updated pseudonym IDS_{new} in the later session. Eventually, $\mathcal{A}$ can obtain the correct K.

Algorithm 1 Passive Attack

```
procedure PASSIVEATTACK($\{IDS, P, Q, R, S\}$)
    for $i = 0$ to 95 do
        $w = \{1\}^i\{0\}^{96-i}$                                  ▷ The Hamming weight of $w$ is $i$
        Set $T = Rot(S, w)$
        Set $m' = IDS \oplus T$
        Set $n' = P \oplus IDS \oplus m'$
        Set $K' = Q \oplus n'$
        Set $IDS'_{new} = Rot(Rot(IDS \oplus n', K \oplus n'), IDS \oplus m')$
        if $S = Rot(Rot(IDS \oplus m', K'), R \oplus n')$ then
            Append $\{K', IDS'_{new}\}$ to result list
    return result list
```

For better illustration, we demonstrate the attack with some real values. Assume that a tag stores the following information representing in hexadecimal format.

$$\begin{aligned} IDS &= \mathtt{4BED\ 09C8\ 2DAD\ F140\ 1009\ BCBC} \\ IDS_{old} &= \mathtt{E110\ 9321\ 1143\ 0909\ B98C\ CC04} \\ K &= \mathtt{1237\ 7A7A\ BCAF\ F002\ 0239\ 6F25} \end{aligned}$$

The protocol runs by server generating m and n and computes P, Q, and R.

$$\begin{aligned} m &= \mathtt{CCE2\ 0101\ 942E\ DDA9\ 8232\ 1D1D} \\ n &= \mathtt{4421\ 31E0\ A148\ 7740\ 70B1\ 1E88} \\ P &= \mathtt{C32E\ 3929\ 18CB\ 5BA9\ E28A\ BF29} \\ Q &= \mathtt{5616\ 4B9A\ 1DE7\ 8742\ 7288\ 71AD} \\ R &= \mathtt{5859\ 2E68\ 779E\ 1D09\ CA21\ C6B5}. \end{aligned}$$

Then the tag response S according to the protocol as:

$$S = \mathtt{1C3C\ 2326\ E60C\ B3A6\ 48EE\ 8686}.$$

Given the on-the-air information IDS, P, Q, R, S, the passive attack runs and outputs two sets of candidate keys:

$$\begin{aligned} K^1 &= \mathtt{1237\ 7A7A\ BCAF\ F002\ 0239\ 6F25} \\ IDS^1_{new} &= \mathtt{E61C\ 1446\ 72C3\ 0030\ 5C51\ 1A07} \\ K^2 &= \mathtt{E5C8\ FE28\ 9D1E\ 1272\ B3B8\ D49C} \\ IDS^2_{new} &= \mathtt{19DE\ 3D56\ AA32\ 3868\ 9C8C\ C6FC} \end{aligned}$$

By listening to the second read of the tag (or simply sending a “Hello” message to the tag) and comparing the reply IDS_{new} and the key in the list, we declare the first set of key are the secret exploited in this attack.

In such an attack, K is revealed by an adversary. It further results in failure to provide confidentiality, mutual authentication, and untraceable privacy. More specifically, with K, an adversary $\mathcal{A}$ can also compute the pseudonym to track the tag’s privacy. Besides, having full control over the communication between the reader and the tag, $\mathcal{A}$ may attack in an active mode to impersonate as either side.

4 Further discussion

The causes of the vulnerability are due to two verification equations R and S. The equation R can be transformed to $R = Rot(Rot(Q, IDS), IDS \oplus P \oplus Q)$. Obviously, all parameters in this equation are public values. It means that R cannot verify the key at all. Besides, the equation S is computed using two consecutive bitwise rotation operations, which can be regarded as one if the intermediate result is ignored. By enumerating all the possible Hamming weights, we may obtain the inverse of such operations.

A simple improvement is to modify these two equations to

$$R = Rot(Rot(K, IDS \oplus n) \oplus m, K)$$

and

$$S = Rot(Rot(K, IDS \oplus m) \oplus n, K \oplus R).$$

This amendment solves the immediate problem mentioned above. Obviously, the storage requirement and communication cost of this amendment are equal to Tewari and Gupta’s protocol [12] which performs remarkably well in comparison with various well-known protocols [3,6–8,13].

However, it is likely this cannot prevent attackers to launch a more sophisticate attack if they can collect more rounds of messages or do a few more rounds of interaction with the tag. The problem remains open where how to create a secure authentication protocol in an ultra-lightweight setting. Therefore, we advise the amended protocol should be accessed with a maximum number of read (say 10 times) and remains hibernate until a factory reset.

We shall also note that if a binary string is uniformly distributed over $\{1\}^{96}$, the hamming weight is actually not uniform from 0 to 96. The probability of a random string having a hamming weight of x is a binomial distribution,

$$\Pr(\texttt{hamming} = x) = \binom{n}{x}\left(\frac{1}{2}\right)^n. \tag{1}$$

Therefore, it is not a good idea to use hamming weight in Rot to perform random shifting. Instead, the rotation is better consuming any seven bits (for example, the lowest seven bits) of the key.

5 Conclusions

In this paper, we show that the ultra-lightweight authentication protocol proposed by Tewari and Gupta is insecure against disclosure attacks. Although the protocol uses only bitwise operations to achieve high performance in terms of storage and computation cost for IoT devices, it fails to provide the fundamental security requirements for an authentication protocol. Our attack stems from lightweight operations in the protocol. To surmount, we also put forward possible improvement. In general, there is a trade-off between computation cost and security requirements. To design a secure ultra-lightweight protocol, it is also necessary to simulate the protocol against different kinds of attacks.

Acknowledgements Funding was provided by National Natural Science Foundation of China (CN) (Grant No. 61402135) and Shenzhen Technical Project (Grant No. JCYJ20150513151706574).

References

1. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805
2. Chen D, Chang G, Sun D, Li J, Jia J, Wang X (2011) TRM-IoT: a trust management model based on fuzzy reputation for internet of things. Comput Sci Inf Syst 8(4):1207–1228
3. Chien HY (2007) Sasi: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Trans Dependable Secure Comput 4(4):337–340
4. He D, Zeadally S (2015) An analysis of RFID authentication schemes for internet of things in healthcare environment using elliptic curve cryptography. IEEE Internet Things J 2(1):72–83
5. Nguyen KT, Laurent M, Oualha N (2015) Survey on secure communication protocols for the internet of things. Ad Hoc Netw 32:17–31
6. Peris-Lopez P, Hernandez-Castro J, Estevez-Tapiador J, Ribagorda A (2006) Emap: an efficient mutual-authentication protocol for low-cost RFID tags. In: Meersman R, Tari Z, Herrero P (eds) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. Lecture notes in computer science, vol 4277. Springer, Berlin, p 352–361
7. Peris-Lopez P, Hernandez-Castro JC, Estévez-Tapiador JM, Ribagorda A (2006) Lmap: a real lightweight mutual authentication protocol for low-cost RFID tags. In: Proceedings of the 2nd Workshop on RFID Security
8. Peris-Lopez P, Hernandez-Castro JC, Estevez-Tapiador JM, Ribagorda A (2006) M2ap: a minimalist mutual-authentication protocol for low-cost RFID tags. In: International Conference on Ubiquitous Intelligence and Computing. Springer, Berlin, pp 912–923
9. Roman R, Alcaraz C, Lopez J, Sklavos N (2011) Key management systems for sensor networks in the context of the internet of things. Comput Electr Eng 37(2):147–159
10. Sundmaeker H, Guillemin P, Friess P, Woelfflé S (2010) Vision and challenges for realising the internet of things. Cluster of European Research Projects on the Internet of Things, European Commision
11. The internet of things (2005) itu international reports. Tech. rep, International Telecommunications Union
12. Tewari A, Gupta BB (2016) Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. J Supercomput. doi:10.1007/s11227-016-1849-x
13. Tian Y, Chen G, Li J (2012) A new ultralightweight RFID authentication protocol with permutation. IEEE Commun Lett 16(5):702–705