I.J. Computer Network and Information Security, **2015, 1, 31-37**
Published Online December 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.01.05

An Improvement over a Server-less RFID Authentication Protocol

Mohsen Pourpouneh
Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran
Email: m_pourpouneh@mehr.sharif.ir

Rasoul Ramezanian and Fatemeh Salahi
Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran
Department of Mathematical and Computer Sciences, Kharazmi University, Tehran, Iran
Email: {ramezanian@sharif.edu, std_fsalahi@khu.ac.ir}

Abstract—With the increased radio frequency identification (RFID) applications different authentication schemes have been proposed in order to meet the required properties. In this paper we analyze the security of a server-less RFID authentication protocol which is proposed by Deng et al. in 2014. Deng et al. proposed an improvement over Hoque et al. protocol to overcome its vulnerability against data desynchronization attack. However, in this paper we show that their protocol is still vulnerable against data desynchronization attack. Furthermore we present an improved version of this protocol to prevent this attack.

Index Terms—Authentication, Reader, RFID Protocols, Tag.

I. INTRODUCTION

Radio Frequency Identification (RFID) is a wireless technology, which is used to automatic identify remote objects embedded with RFID tags [1]. RFID can be used in a great variety of applications such as supply chain management, transportation, livestock management, animal tracking, human implants, library, and so on [2].

In an RFID system, the cost of the tags is low, which implies that the tags have very limited computational capabilities and storage. General-purpose security protocols cannot be applied directly to the RFID system [3]. The low cost demanded for RFID tags forces them to be very resource limited. Typically, they can only store hundreds of bits, have 5-10K logic gates, and a maximum communication range of a few meters. Within this gate counting, only between 250 and 3000 gates can be devoted to security functions. It is interesting to recall that for a standard implementation of the Advanced Encryption Standard (AES), 20-30K gates are needed [4].

Therefore, these protocols mostly use lightweight primitives known to be implementable on RFID tags. Additionally, power restrictions should be taken into account, since most RFID tags in use are passive. Furthermore, these systems are unable to store passwords securely because they are not tamper-resistant at all[4].

According to the cryptogrophic primitives used in the RFID authentication protocols, they are usuelly classified into four groups, based on the structure of the protocol. The first class contains the protocols that apply ordinary cryptographic functions, such symmetric encryption, cryptographic hash function, or even the public key algorithms. The second class are protocolas that use random number generator and one-way hash functions. The third class refers to those protocols that apply random number generators and cyclic redundancy code(CRC) checksum, which sometimes are called as “Lightwright” protocols. The last class, are the “Ultra Lightweight” protocols. These protocols apply simple bitwise operations such as XOR, AND, OR, etc.

According to the components used in a RFID system, they are divided into two categories. The first category is based on a back-end server [5, 6, 7]. In back-end server based RFID systems, the reader has to communicate with the back-end server containing information of all readers and tags through a secure channel in order to get the required data from a tag, Fig 1.

While the back-end server approach provides security and privacy protections, it is dependent on a reliable connection between an RFID reader and the back-end database [8]. The authentication protocol should be provided, even if the connection between the reader and the back-end server was not established.

Several studies have recently been made on authentication protocol for RFID tag and reader without back-end server [9, 10, 11, 12]. Fig 2 shows the process of the authentication process without the back-end server.

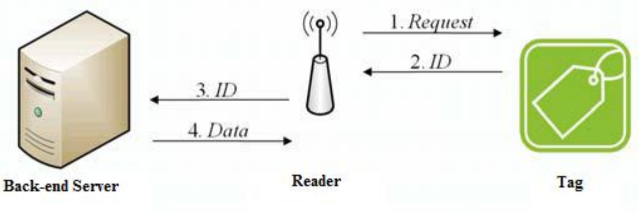

Fig 1: Server-Based RFID protocols.

In server-less protocols, an abstraction of information of a particular reader and all tags are kept in every reader. Obliviously these protocols are more practical, since the assumption of the existence of a secure communicating channel between reader and server is not needed here.

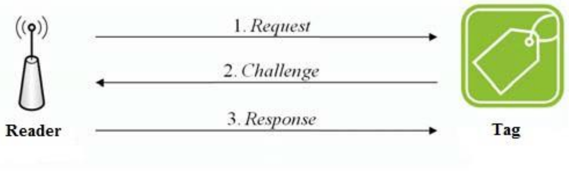

Fig 2: Server-less RFID protocols.

In 2010 Hoque et al. suggested a server less, untraceable authentication, and forward secure protocol for RFID tags [13]. Hoque's authentication protocol safeguards both reader and tag against attacks as often as possible without the intervention of back-end server.

After that, in 2014 Deng et al. showed that the Hoque's authentication protocol is vulnerable to attacks of data desynchronization [14], and proposed an improvement over Hoque's authentication protocol and claimed that it can withstand the attack of data desynchronization.

However, our analysis shows that Deng's protocol is still vulnerable to data desynchronization attack, when the protocol is performed for more than two runs. We propose an improvement over Deng's et al. protocol, which is secure under desynchronization attack.

In the rest of this paper, first in Section *II* a brief review of Deng's authentication protocol is presented. Then after showing the vulnerability of their protocol in Section *III*, we describe an improvement over this protocol in Section *IV*. After that in Section *V*, we discuss the security requirements of our protocol, and in Section *VI*, the performance evolution of our protocol is introduced. Finally, conclusions will be drawn in Section *VII*.

II. Deng's Authentication Protocol

In this section we review Deng's authentication protocol [14].

1) *Notions and assumptions*

In the system, all tags and readers have knowledge of a function M(...) and a pseudorandom number generator P(...). P(...) is a low cost random number generator which applies to the RFID system. M(...) and h(...) are assumed as an one way hash functions. Each RFID reader R has a contact list L and a unique identifier r. L and r are obtained from a Certificate Authority (CA). In addition, each tag T includes a unique secret t and a unique identifier id. Subscripts are utilized to describe a particular T or R and their variables. Other notations are listed below:

- ||: Concatenation operation.
- $\oplus$: Exclusive-or operation.
- $Rand_i$: Random number generated by reader R_i.
- $Rand_j$: Random number generated by tag T_j.
- r_i: Identifier of reader R_i.
- t_j: Secret value of tag T_j.
- n_i: Message generated by reader R_i for authentication.
- n_j: Message generated by tag T_j for authentication.
- $Seed_{Tj}$: $(h(r_i, t_j))$ the secret value shared between reader R_i and tag T_j.
- $Seed_{PTj}$: The previous secret value stored in reader R_i.
- L_i: Downloaded list of tags information from CA by reader R_i where,

$$L_i = \begin{cases} Seed_{T1},\ Seed_{PT1} \\ Seed_{Tn},\ Seed_{PTn} \end{cases}$$

2) *Deng's Protocol description*

Deng et al. protocol, is a server-less RFID protocol and the RFID reader R_i stores contact list L_i and an identifier r_i in its nonvolatile memory. The contact list L_i comprises information about the RFID tags that R_i can access to and each tag contains the current seed $Seed_{Tj}$, and the previous seed $Seed_{PTj}$. The protocol is shown in Fig 3.

In a general authentication process, the current seed of T_j, will be utilized to accomplish the mutual authentication between the reader and the tag. Nevertheless, if the reader fails to look up the current seed for the desynchronization of the shared secret, it may use the previous seed to complete the authentication.

III. Vulnerability of Deng's Protocol

Deng improved Hoque's protocol to troubleshoot primary protocol from desynchronization attack [6]. In this section we claim that, this protocol is secure under desynchronization attack only for one run, but it is vulnerable to this attack when the protocol is performed for more than a single run.

The main goal of this attack is to ruin the synchronization between a legitimate reader and a tag, in order to prevent them from successfully communicating with each other in the latter communications. Moreover, this attack can be carried to prevent the reader from successfully updating the information (identification for instance) of a tag after a successful communication.

```
1.  $R_i \longrightarrow T_j$: request , $rand_i$
2.  Tj : $n_j = P(seed_T \oplus (randi \| randj))$
3.  $R_i \longleftarrow T_j$: $n_j$ , $rand_j$
4.  $R_i : n_i = rand_i$
5.  for all m from 1 to n
6.  Let $n_m = P(seed_m \oplus (rand_i \| rand_j))$
7.  if ($n_m == n_j$) then
8.      Let s = M($seed_m$)
9.      $n_i = P(s)$
10.     $seed_m = M(s)$
11.     $R_i \longrightarrow T_j$: $n_i$
12. else    Let $n_m = P(seed_m \oplus (rand_i \| rand_j))$
13. if ($n_m == n_j$) then
14.     Let s = M($seed_{mp}$)
15.     $n_i = P(s)$
16.     $seed_m = M(s)$
17.     $R_i \longrightarrow T_j$ : $n_i$
18.  $T_j$ : Let k = M($seed_{Tj}$)
19. Let a = P(k)
20. if (a == $n_i$) then
21.     $seed_{Tj} = M(k)$
22. else
23.     Reader is not authorized
                    or is an adversary
```

Fig 3: Deng et al. Protocol

Our proposed desynchronization attack uses two runs. Due to the protocol, before starting the protocol the following values are stored in Reader R_i and Tag T_j database:

R_i: $Seed_{Tj}$, $Seed_{PTj}$
T_j: $Seed_{Tj}$ (which is equal to $Seed_{PTj}$ for the first time).

In which $Seed_{Tj}$ is supposed to store the latest value of shared agreed seed between reader R_i and tag T_j, and $Seed_{PTj}$ is a constant for the first value of seed, obtained from CA since the beginning of construction. (In Deng's protocol the value of $Seed_{PTj}$ is not supposed to change through different runs.)

Assume that Reader and Tag perform a successful run, then the following values are replaced in Reader's and Tag's databases:

R_i: $Seed'_{Tj}$, $Seed_{PTj}$
T_j: $Seed'_{Tj}$

in which $Seed'_{Tj}$ is a fresh value.

In the second run of the protocol reader R_i sends request to tag T_j and T_j replies with the value of $Seed'_{Tj}$. The reader R_i authenticates the tag T_j since the value of $Seed'_{Tj}$ is the same in both participant's databases, then R_i updates the value of $Seed'_{Tj}$ to $Seed''_{Tj}$ and sends the fresh value to T_j.

In line 11 or 17 of the protocol (which the Reader is responding to the Tag) an adversary can change the value of message n_i or prevent T_j receiving n_i. Thus, the reader R_i has replaced the $Seed'_{Tj}$ with $Seed''_{Tj}$ while the tag T_j has the previous value. Therefore, the shared secret between the tag T_j and the reader R_i is not identical, which will threw the RFID system into confusion (Fig 4).

After a successful data desynchronization attack, since adversary makes the reader R_i and the valid tag T_j share the different secrets. The attack destroys the availability of the protocol.

Reader R_i	**Tag T_j**
$Seed_{TJ}$ $Seed_{PTJ}$ *C* *C*	$Seed_{TJ}$ *C*
Beginning: Both side's information are sync.	
$Seed_{TJ}$ $Seed_{PTJ}$ *U_1* *C*	$Seed_{PTJ}$ *U_1*
After one successful run: $Seed_{PTJ}$ is fixed for future use and $Seed_{TJ}$ Is same both sides.	
$Seed_{TJ}$ $Seed_{PTJ}$ *U_2* *C*	$Seed_{PTJ}$ *U_1*
In second Run: Adversary prevents recieveing the update message (U_2) by Tag T_j. Tag and Reader are not sync and can not communicate with eachother any more.	

Fig 4: Desynchronization attack on Deng's protocol.

IV. Improved Protocol

The security gap which led to de-synchronization attack in Deng et al. protocol, is the fixed value, C, in reader's database. The goal of storing this value is to make further communications possible even if in a specific run the updated message is not received by tag. The idea is a common idea to solve these kind of de-synchronization attack (e.g. [15], [16], [17]) but it has not established correctly in [14] and as described in previous section it is still vulnerable to this attack.

In order to settle the data de-synchronization attack issue, the reader ought to update the value of $Seed_{PTj}$ to latest authorized value in every successful run of the protocol. If the value of $Seed_{PTj}$ changes dynamically in every run, even if an adversary interrupt the message n_i or prevent n_i to be received by the tag the values of $Seed_{PTj}$ in reader's database and $Seed_{Tj}$ in tag's database are identical yet and they still can be authorized by each other (Fig 5).

Therefore, in improved protocol reader sends the request and generated nonce, and tag responses with its seed and nonces (both reader's nonce and its own generated nonce). Reader after receiving this message, extracts tag's seed and checks if it is equal to latest updated value from last run of the protocol.

Reader R_i	Tag T_j
$Seed_{TJ}$ Seed $_{PTJ}$ C C	$Seed_{TJ}$ C
Beginning: Both side's information are sync.	
$Seed_{TJ}$ Seed $_{PTJ}$ U_1 C	Seed $_{PTJ}$ U_1
After one successful run: Seed $_{PTJ}$ is fixed for future use and $Seed_{TJ}$ Is same both sides.	
$Seed_{TJ}$ Seed $_{PTJ}$ U_2 U_1	Seed $_{PTJ}$ U_1
...	...
$Seed_{TJ}$ Seed $_{PTJ}$ U_{i+1} U_i	Seed $_{PTJ}$ U_i
...	...
In every Run: Even if adversary prevent recieveing the update message (U_{i+1}) by Tag T_j, Tag and Reader can authenticate eachother by U_i.	

Fig 5: Resistance against desynchronization attack of improved protocol.

In case of equality reader authenticates the tag, and otherwise it check whether it is equal to the latest value which is accepted in last successful run of the protocol.

The rest of the protocol is the same as the original one except the update step. In improved protocol the value of $Seed_{PTJ}$ will update to last accepted $Seed_{TJ}$. Our improved protocol is shown in Fig 6.

```
$R_i \longrightarrow T_j$: request , $rand_i$
Tj : $n_j = P(seed_T \oplus (randi \| randj))$
R $\longleftarrow T_j$: $n_j$ , $rand_j$
$R_i : n_i = rand_i$
for all m from 1 to n
Let $n_m = P(seed_m \oplus (rand_i \| rand_j))$
if $(n_m == n_j)$ then
    $seed_{PTj} = seed_{Tj}$ \\Updating the old seed
    Let $s = M(seed_{Tj})$
    $n_i = P(s)$
    $seed_m = M(s)$ \\Updating the new seed
    $R_i \longrightarrow T_j$: $n_i$
else for all m from 1 to n
Let $n_m = P(seed_{PTj} \oplus (rand_i \| rand_j))$
if $(n_m == n_j)$ then
    Let $s = M(seed_{PTj})$
    $n_i = P(s)$
    $seed_m = M(s)$ \\Updating the new seed
    $R_i \longrightarrow T_j : n_i$
$T_j$ : Let $k = M(seed_{Tj})$
Let $a = P(k)$
if$(a == n_i)$ then
    $seed_{Tj} = M(k)$
else
    Reader is not authorized
                    or is an adversary
```

Fig 6: Improved protocol

V. Security Analysis

In this section, we analyze our protocol against different types of attacks. For each attack, we first give a brief description of the attack, and the common assumptions about the adversary. It is followed by an explanation of how the protocol defends against the attack. We denote the adversary as **A**, and a legitimate reader and tag as R_i and T_j respectively. A fake tag j impersonating the real tag j is depicted as Ť.

Basic attack This attack occurs when **A** can access the information of tag. Under this attack, we generally assume that **A** has a list of targeted RFID tags. Our protocol is not vulnerable under this attack because consider for example, the tag T_j attached to a valuable container in a warehouse. **A** then queries every tag in the warehouse to decide the most valuable one to steal. In our protocol, each time any reader queries T_j, T_j generates a new response $P(seed_T \oplus (rand_i \|rand_j))$ for authentication. Thus ***A*** cannot identify which RFID tag is on his list. This protects the privacy of the tag.

Tracking Under this attack, ***A*** tries to distinguish T_j from other RFID tags over time. For example, T_j could be attached to a passport. By repeatedly querying with a value that yields a consistent reply, ***A*** will be able to track the movements of T_j over time.

Under our protocol, ***A*** can reuse the same n_A and $rand_A$ for every query, but cannot predict the random $rand_j$

generated each time by T_j. In the protocol, we return the entire P(seed$_{Tj}$$\oplus$ (rand$_i$ || rand$_j$)).Since rand$_j$ is a random number chosen by the tag for each query, ***A*** learns nothing from repeated queries.

Cloning Under this attack, ***A*** will usually first query T_j and obtain a response. He then places the response on a fake RFID tag, $\check{T}_j$. By creating fake RFID tags that contain the responses of real RFID tags, A attempts to pass off his counterfeits as legitimate. A succeeds if R_i believes that $\check{T}_j$ is T_j. Under our protocol, T_j will return a different response based on the random n_i and rand$_i$ provided by R_i each time. Since ***A*** cannot predict the random rand$_i$ generated each time by R_i, the response that A obtains from T_j will not be the same. Thus ***A*** cannot create a $\check{T}_j$ that can fool R_i.

Eavesdropping In this attack, ***A*** captures all the messages transferring between readers and tags in RFID system, which means that ***A*** is able to observe all interactions between R_i and T_j. By this ability ***A*** tries to use the data to launch any of the three attacks mentioned above.

In our protocol, every transaction between R_i and T_j begin by both parties generating a different n_i and n_j. An **A** eavesdropping on the communication observes a different query and a different response each time, even if R_i is querying the same tag T_j. Thus, our protocol prevents ***A*** from using eavesdropping to launch a basic privacy attack, tracking attack or cloning attack.

Physical attack We consider two different flavors of physical attack. First, we consider ***A*** compromising R_i. The adversary will know the contents of L_i, as well as rand$_i$. He will therefore be able to impersonate R_i and obtain data from tags T_1,...,T_n . The goal is to prevent ***A*** from using the knowledge to create counterfeit tags. Let T_j be in L_i, and ***A*** wishes to create a counterfeit tag $\check{T}_j$ that can fool another authenticated RFID reader R_x. ***A*** knows $M(r_i ,t_j)$ and id$_j$ from L_i . To create $\check{T}_j$ to fool R_x, ***A*** has to be able to derive $M(r_x,t_j)$. This is because each M(.,.) value in the access list is different for every RFID reader. R_i will have $M(r_i,t_j)$,and R_x will have $M(r_x,t_j)$. Thus ***A*** cannot substitute his $M(r_i,t_j)$ and id$_j$ into $\check{T}_j$. Since M(.,.) is irreversible, ***A*** cannot derive t_j from $M(r_i,t_j)$.

Next, we consider ***A*** compromising tag T_j. The adversary will now be able to create a fake $\check{T}_j$ that can fool the honest R_i. We want to prevent ***A*** from creating another tag that can fool R_i. We let this other tag be T_x, and assume that T_x is inside L_i. Since ***A*** has compromised T_j, we assume that ***A*** knows any information that R_i passes to T_j. To create T_x to fool R_i, ***A*** has to be able to generate the correct $M(r_i,t_x)$. However, each RFID tag has a unique secret t. Thus ***A*** knowing t_j cannot derive t_x. Therefore, ***A*** cannot create a fake T_x to fool R_i [8].

Denial of service (DoS) attack In DoS attacks ***A*** tries to find a way to fail target tag from receiving services. To launch a DoS attack, ***A*** sends a large number of requests to the backend server to overwhelm the server. This results in a legitimate R_i being unable to access the database to obtain information about the tag. Under our solutions, a reader only needs to contact the server once to obtain an access list L_i .The reader is then able to interact with RFID tags without further interaction with the server. A DoS attack under our schemes will not affect readers that have already been authenticated.

Desynchronization attack In desynchronization attack, which is one kind of DoS attacks, the shared secret values among the tag and the reader are made inconsistent by an ***A***. Then, the tag and reader cannot recognize each other in future and tag becomes disabled. For protocols that require some synchronization between central database and tag, a common defense against DoS attacks is to require Tj to change its value only after receiving some confirmation generated by the database[18, 19]. Under our protocol the value of Seed$_{PTj}$ for reader R_i changes dynamically in every run, even if an adversary changes the message n_i or prevent n_i to be received by the tag the values of Seed$_{PTj}$ in reader's database and Seed$_{Tj}$ in tag's database are identical and they still can be authorized by each other. Thus our protocol is guaranteed under this attack.

VI. Performance Evolution

RFID technology requires that the RFID protocols not only to be secure, but to be practical and efficient. In this section, we evaluate the performance of our protocol by estimating its communication cost, the number of needed operations and its needed space.

The overall communication costs of our authentication protocol are similar with the overall communication costs of Deng's authentication protocol. Considering communication cost, assuming that both reader and tag id$_s$ have the same length, the authentication protocol requires 2|n| + 2|id$_j$ | + m bits where |n| is the length of the random numbers rand$_i$ and rand$_j$.

Next we study computational cost of protocol. Similarly to Deng's protocol the improved authentication protocol contains two hash functions, M(.) and h(,), but the reader get L_i from CA, and h(,) is computed by CA. Therefore the cost of the protocol may be determined based on the computation of M(.) function. In our protocol it can be seen that M(.) is computed twice. The cost for our protocols is higher than alternative protocols [20, 21, 22], which require the tag to perform only one hash function.

Now we study the required space of the protocol. An RFID tag in improved scheme just stores its one seed for every RFID reader in the system. Certainly a tag still requires other memory space for communication and computation. However, computation in the improved protocol only contains hash operation and only one value utilized as authentication needs to be stored, so the required memory space is very limited [6].

VII. Conclusion

RFID authentication systems are rapidly developing in different areas. But, designing a secure authentication protocol for low-cost RFID tags is still a challenging problem. In this paper the cryptanalysis of a recent lightweight RFID authentication protocol is proposed, and we showed that the improved RFID authentication protocol proposed by Deng is still vulnerable to attack of data desynchronization.

Our Improved protocol is resistant against data desynchronization attack, and it satisfies the required security requirements, such as privacy protection, tracking attack resistance, cloning and physical attack resistance. Our improved protocol is lightweight which makes it suitable for the low-cost RFID environment.

REFERENCES

[1] L. Wang, X. Yi, C. Lv and Y. Guo, \Security Improvement in Authentication Protocol for Gen-2 Based RFID System", Journal of Convergence Information Technology, AICIT, vol.6, no.1, pp.157-169, (2011).

[2] Ting, P. H. (2013). An Ecient and Guaranteed Cold-Chain Logistics for Temperature-Sensitive Foods: Applications of RFID and Sensor Networks. International Journal of Information Engineering and Electronic Business (IJIEEB), 5(6), 1.

[3] Chai Q (2012) Design and analysis of security schemes for low-cost RFID systems [D]. Ph.D. dissertation of Waterloo University, Water-loo.

[4] Peris-Lopez P, Hernandez-Castro J C, Estevez-Tapiador J M, Ribagorda A, M2AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags, (2006), Springer-Verlag Berlin Heidelberg.

[5] Wu, Z. Y., Chen, L., & Wu, J. C. (2013). A reliable RFID mutual authentication scheme for healthcare environments. Journal of medical systems, 37(2), 1-9.

[6] Lars Kulseng, Zhen Yu, Yawen Wei, Yong Guan, Lightweight mutual authentication and ownership transfer for RFID systems, in: Proceed-ings of IEEE INFOCOM 2010, pp. 15, 2010.

[7] Karda, S., elik, S., Arslan, A., & Levi, A. (2013). An ecient and private RFID authentication protocol supporting ownership transfer. In Lightweight Cryptography for Security and Privacy (pp. 130-141). Springer Berlin Heidelberg.

[8] Tan, Chiu C., Bo Sheng, and Qun Li. \Secure and serverless RFID au-thentication and search protocols." Wireless Communications, IEEE Transactions on 7.4 (2008): 1400-1407.

[9] Ahamed, S. I., Rahman, F., Hoque, E., Kawsar, F., & Nakajima, T. (2008, April). S3PR: Secure server-less search protocols for RFID. In Information Security and Assurance, 2008. ISA 2008. International Conference on (pp. 187-192). IEEE.

[10] Lee, C. F., Chien, H. Y., & Laih, C. S. (2012). Server-less RFID authentication and searching protocol with enhanced security. International Journal of Communication Systems, 25(3), 376-385.

[11] Kim, Z., Kim, J., Kim, K., Choi, I., & Shon, T. (2011, May). Un-traceable and server-less RFID authentication and search protocols. In Parallel and Distributed Processing with Applications Workshops (IS-PAW), 2011 Ninth IEEE International Symposium on (pp. 278-283). IEEE.

[12] Kim, S., Lee, K., Kim, S., & Won, D. (2009). Security analysis on anonymous mutual authentication protocol for RFID tag without back-end database and its improvement. World Acad Sci Eng Technol, 460-464.

[13] Hoque ME, Rahman F, Ahamed SI et al (2010) Enhancing privacy and security of RFID system with server-less authentication and search protocols in pervasive environments. Wirel Pers Commun 55:6579.

[14] Deng, M., Yang, W., & Zhu, W. (2014). Weakness in a Server-less Authentication Protocol for Radio Frequency Identification. In Mechatronics and Automatic Control Systems (pp. 1055-1061). Springer International Publishing.

[15] Rostampour S., Eslamnezhad Namin M., Hosseinzadeh M., (2014), A Novel Mutual RFID Authentication Protocol with Low Complexity and High Security, I.J. Modern Education and Computer Science.

[16] Habibi M H, Gardeshi M, R. Alaghband M, (2011), Practical Attacks on a RFID Authentication Protocol Conforming to EPC C-1 G-2 Standard, International Journal of UbiComp (IJU), Vol.2, No.1.

[17] Tieyan Li, Guilin Wang, (2007), Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols, New Approaches for Security, Privacy and Trust in Complex Environments, pages 109-120.

[18] T. Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm, Athens, Greece, September 2005. IEEE.

[19] S.-M. Lee, Y. J. Hwang, D. H. Lee, and J. I. L. Lim. Efficient authentication for low-cost RFID systems. In O. Gervasi, M. Gavrilova, V. Kumar, A. Lagana`a, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, International Conference on Computational Science and its Applications - ICCSA 2005, Proceedings, Part I, volume 3480 of Lecture Notes in Computer Science, pages 619–627, Singapore, May 2005. Springer-Verlag.

[20] Weis, S. A., Sarma, S. E., Rivest, R. L., & Engels, D. W. (2004). Security and privacy aspects of low-cost radio frequency identification systems. In Security in pervasive computing (pp. 201-212). Springer Berlin Heidelberg.

[21] Molnar, D., & Wagner, D. (2004, October). Privacy and security in library RFID: issues, practices, and architectures. In Proceedings of the 11th ACM conference on Computer and communications security (pp. 210-219). ACM.

[22] Tsudik, G. (2006, March). YA-TRAP: Yet another trivial RFID authentication protocol. In Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on (pp. 4-pp). IEEE.

Authors' Profiles

Mohsen Pourpouneh was born in 1989 in Isfahan. He got his B.Sc. (2011) and M.Sc. (2013) in Computer Science, from Shahid Beheshti University and Tehran University, respectively. He started his career as a Ph.D. student at Sharif University of Technology, Tehran, Iran in 2013. His research interest includes Formal method, Specifying and Verifying Security Protocols, Computational Number Theory, Electronic Voting, Multi-Agent Systems.

Rasoul Ramezanian was born in Mashhad in 1979. He got his B.S. and M.S. in Mathematics. In 2008, he graduated from a Ph.D. program of Mathematical Science Department of Sharif University of Technology, Tehran, Iran. He is an assistant professor at the same department.His research interests include Formal method, Specifying and Verifying Security Protocols, Multi-Agent Systems, and Process Algebra.

Fatemeh Salahi was born in Tehran in 1988. She is a Computer Science student in Kharazmi University and attending in Sharif University of Technology. Her research interests include Specifying and Verifying Security Protocols, Data Mining, Data Stream Clustering.

How to cite this paper: Mohsen Pourpouneh, Rasoul Ramezanian, Fatemeh Salahi,"An Improvement over a Server-less RFID Authentication Protocol", IJCNIS, vol.7, no.1, pp.31-37, 2015. DOI: 10.5815/ijcnis.2015.01.05