SECURITY FLOWS AND IMPROVEMENT OF A RECENT ULTRA LIGHT-WEIGHT RFID PROTOCOL

Mehrdad Kianersi and Mahmoud Gardeshi

[1]Department of Information Technology and Communication, I.H.University, Tehran, Iran
mehrdad.kianersi@Gmail.com
[2] Department of Information Technology and Communication, I.H.University, Tehran, Iran
mgardeshi2000@yahoo.com

ABSTRACT

In this paper, we analyze the security vulnerabilities of SSL-MAP, an ultra-lightweight RFID mutual authentication protocol recently proposed by Rama N, Suganya R. We present two effective attacks, a de-synchronization attack and a full-disclosure attack, against this protocol. The former permanently disables the authentication capability of a RFID tag by destroying synchronization between the tag and the RFID reader. The latter completely threats a tag by extracting all the secret information that are stored in the tag. The de-synchronization attack can be carried out in three round of interaction in SSL-MAP while the full-disclosure attack is accomplished across several runs of SSL-MAP. We also discuss ways to counter the attacks.

KEYWORDS

RFID, Mutual authentication, Low-cost RFID Tag, SSL-MAP

1. INTRODUCTION

Radio Frequency Identification (RFID) systems offers improved efficiency in inventory control, logistics, and supply chain management. As such, they are of great interest to enterprises intensively reliant on supply chains, particularly large retailers and consumer product manufacturers. The long-term goal of these organizations is to integrate RFID on the retail level. Without proper protection, widespread adoption of retail RFID could raise privacy concerns for everyday consumers.

RFID systems consist of three main components: tags, readers and back-end databases. Tags are radio transponders attached to physical objects. Each tag contains a microchip with a certain amount of computational and storage capabilities and a coupling element. Such devices can be classified according to memory type and power source. Another relevant parameter is tag price, which creates a broad distinction between high-cost and low-cost RFID tags. Radio transceivers, or readers, query these tags for some (potentially unique) identifying information about the objects to which tags are attached. Although readers are often regarded as a simple conduit to a back-end database, for simplicity we treat a reader and a back-end database as a single entity.

2. RELATED WORK

In 2007, Huang chien [12] classified RFID tags according to capability of supporting various cryptographic functions. In his classification, RFID tags divide to two main categories: High-cost and Low-cost. High-cost tags are divided into full fledged that support conventional cryptographic functions like symmetric key encryption, one way hash functions and even public key cryptography and simple tags that can support random number generator and one-way hash functions. In other side, low-cost tags divide into lightweight tags and ultra lightweight tags. Lightweight tags can support random number generators and simple functions such as CRC. But ultra light-weight tags can only compute simple bitwise operations like AND, OR, XOR and rotation, etc. In 2006, Peris et al proposed a new family in mutual authentication called UMAP family. The first member of this family was M2AP [4] and followed by LMAP [5] and EMAP [6]. In this family of protocols, for being very light, the authors used only the simple bitwise operations such as AND, OR, XOR and addition module 2^m. These protocols have been attacked since publication [7, 8, 9, 10, 11], due to low complexity and weak tools that has been used. In 2007, Chien [12] proposed a very interesting ultra lightweight mutual authentication protocol which used a rotation function Rot(x, y) for first time. His protocol was SASI (Strong Authentication and Strong Privacy). But, SASI has broken due to carelessness in computing public messages and bitwise operations [14]. Then, in 2008, Peris et al. [11] proposed another strong ultra light-weight mutual authentication called Gossamer. This protocol was appeared more secure because of using two consecutive rotation functions and also defining a MixBits function that generates new random numbers by using two random numbers. But Gossamer was attacked and broken also, due to some structural and MixBits function weaknesses [12, 13]. Before executing any attack on Gossamer, Rama et al [1] proposed a gossamer based protocol called SSL-MAP. The core of this protocol is taken from Gossamer but there are a few differences in last step of mutual authentication phase (in computing message D and the way of letting ID in this message). In this paper we show that these changes despite of making heavy computations do not enhance security of protocol. After this protocol the researchers proposed various ideas for mutual authentication such as David-Prasad protocol [16], Lee et al protocol [17] but these protocols analyzed by Peris et al [18, 19] very soon.

3. Review of SSL-MAP

The protocol comprises three stages: Tag identification, mutual authentication, and updating as shown in Fig. 1.

3.1. Tag Identification

The reader sends a “hello” message to the tag. The tag responds with index-pseudonym (IDS). The reader uses this ID as a reference number to search for the shared keys of the tag in its database. If the database has an entry associated an IDS, next phase starts, otherwise the reader requests for older IDS to identify the tag.

3.2. Mutual Authentication

With IDS, the reader acquires private information linked to the tag from the database. Then the reader generates pseudonyms n_1 and n_2, constructs three concatenated public messages $A \| B \| C$ and sends them to the tag. Where c is a 96 bit length constant. The tag in reply sends a public message D or an error message depending on successful reader authentication. So we have two authentications as follow:

1. **Reader Authentication:** From messages A and B, the tag extracts pseudonyms n_1 and n_2 respectively. Then it computes n_3, k_1^*, k_2^*, $\widetilde{n_1}$ and builds a local version of message C as C'. This is compared with the received value C. If both values are same, the reader is authenticated.

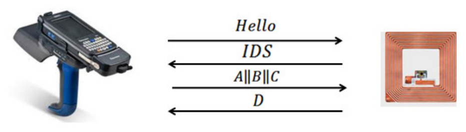
2. **Tag Authentication***:* Finally, the tag sends message *D* to the reader. On receiving *D*, this value is compared with a computed local version. If they are same, the tag is authenticated; otherwise the protocol is abandoned.

3.3. Update Phase

After successfully completing the mutual authentication phase between the reader and the tag, both locally update *IDS* and keys k_1, k_2 as follows:

$$\widetilde{n_2} = \text{MixBits}(\widetilde{n_1}, n_3) \quad (1)$$

$$IDS^{n+1} = ROT((ROT(\widetilde{n_1} + k_1^* + IDS^n + \widetilde{n_2}, \widetilde{n_1}) + k_2^* \oplus \widetilde{n_2}, n_3) \oplus \widetilde{n_2} \quad (2)$$



$$A = ROT((ROT(IDS + k_1 + c + n_1, k_2) + k_1, k_1)$$
$$B = ROT((ROT(IDS + k_2 + c + n_2, k_1) + k_2, k_2)$$
$$n_3 = MixBits(n_1, n_2)$$
$$k_1^* = ROT((ROT(n_2 + k_1 + c + n_3, n_2) + k_2 \oplus n_3, n_1) \oplus n_3$$
$$k_2^* = ROT((ROT(n_1 + k_2 + c + n_3, n_1) + k_1 + n_3, n_2) + n_3$$
$$\widetilde{n_1} = MixBits(n_3, n_2)$$
$$C = ROT((ROT(n_3 + k_1^* + c + \widetilde{n_1}, n_3) + k_2^* \oplus \widetilde{n_1}, n_2) \oplus \widetilde{n_1}$$
$$D = ROT((ROT(n_2 + k_2^* + Y + \widetilde{n_1}, n_2) + k_2^* + \widetilde{n_1}, n_3) + n_1$$

Where c = 0x3243F6A8885A308D313198A2

Fig. 1. SSL-MAP Protocol

$$k_1^{n+1} = ROT((ROT(n_3 + k_2^* + c + \widetilde{n_2}, n_3) + k_1^* + \widetilde{n_2}, \widetilde{n_1}) + \widetilde{n_2} \quad (3)$$

$$k_2^{n+1} = ROT((ROT(IDS^{n+1} + k_2^* + c + k_1^{n+1}, IDS^{n+1}) + k_1^* + k_1^{n+1}, \widetilde{n_2}) + k_1^{n+1} \quad (4)$$

4. Security Analysis

Security analysis of SSL-MAP protocol result that following attacks are possible to carry out.

4.1. De-Synchronization Attack

The tag updates its values irrespective of whether the reader has received message *D* and verified it or not whereas, the reader updates its values only after receiving and verifying message *D*. This causes a difference between the storage of the tag and the reader in case that message *D* does not received by the reader. To avoid this de-synchronization, in Gossamer, the tag is considered to be keeping the older values of *IDS* and keys in memory. So in such case that a de-synchronization occurs the reader can ask for the older *IDS* (not updated) and both can be in synchronization again. However, a de-synchronization attack can still be launched successfully using following procedure [3]:

1. Suppose a tag keep the synchronized value as:

1. $IDS^{n+1} = IDS_x$
2. $k_1^{n+1} = k_{1x}$
3. $k_2^{n+1} = k_{2x}$

This tag now communicates with a reader. The attacker records the corresponding message as A_x, B_x, C_x (being public messages and under the assumption that communication between the reader and the tag is not secure). Now the attacker interrupts message D_x and does not allow it reaches to the reader. The tag does not know whether the reader has verified D or not and updates its value as:

1. $IDS^n = IDS_x$ 2. $k_1^n = k_{1x}$ 3. $k_2^n = k_{2x}$

4. $IDS^{n+1} = IDS_y$ 5. $k_1^{n+1} = k_{1y}$ 6. $k_2^{n+1} = k_{2y}$

2. Next, the attacker allows the tag and the reader to run the protocol without intervening. As IDS_y is not recognized by the reader (did not update its value as D was not received), so it asks the tag for the older values. The tag sends IDS_x which is recognized by the reader and they complete the protocol. After completion, the tag updates its values as:

1. $IDS^n = IDS_x$ 2. $k_1^n = k_{1x}$ 3. $k_2^n = k_{2x}$
4. $IDS^{n+1} = IDS_z$ 5. $k_1^{n+1} = k_{1z}$ 6. $k_2^{n+1} = k_{2z}$

3. Now, the attacker intervenes and sends a "*hello*" message to the tag. The tag responds with IDS_z. The attacker pretends that it cannot identify IDS_z and asks for the older values. The tag responds with IDS_x. Attacker has already copied A_x, B_x, C_x, which are legitimate sub-messages against IDS_x and n_{1x}, n_{2x} generated during step1. Protocol is completed and tag has the following values in memory:

1. $IDS^n = IDS_x$ 2. $k_1^n = k_{1x}$ 3. $k_2^n = k_{2x}$
4. $IDS^{n+1} = IDS_y$ 5. $k_1^{n+1} = k_{1y}$ 6. $k_2^{n+1} = k_{2y}$

Whereas, the reader has the following values in its database:

1. $IDS^{n+1} = IDS_z$ 2. $k_1^{n+1} = k_{1z}$ 3. $k_2^{n+1} = k_{2z}$

As a consequence, the synchronization between them is failed. Since, reader has IDS_z, k_{1z}, k_{2z} in its database, and does not recognize both triple IDS_x, k_{1x}, k_{2x} and IDS_y, k_{1y}, k_{2y}. The tag is unable to establish an association with reader, the next time that they communicate.

4.2. Full Disclosure Attack

Here we establish another attack that leads to disclosure of all secret information on tag. In this attack we need observe several rounds of protocol. This attack works if the following condition is satisfied: n_1, n_2 mod 96 = 0. In this case because $MixBits(0,0) = 0$, n_3, $\widetilde{n_1}$,$\widetilde{n_2}$ all becomes Zero.

$n_1, n_2 \ mod \ 96 = 0$

$MixBits(0 \ mod \ 96 \ , 0 \ mod \ 96) = 0 \ mod \ 96$

$$A = ROT((ROT(IDS^n + k_1 + c, k_2) + k_1, k_1) \quad (5)$$

$$B = ROT((ROT(IDS^n + k_2 + c, k_1) + k_2, k_2) \quad (6)$$

$$k_1^* = k_1 + c + k_2 \quad (7)$$

$$k_2^* = k_2 + c + k_1 \quad \text{Then} \quad (8)$$

$$k_1^* = k_2^* \quad (9)$$

$$C = k_1^* + c + k_2^* \quad \text{Then} \quad (10)$$

$$C - c = k_1^* + k_2^* \quad (11)$$

$$IDS^{n+1} = k_1^* + IDS^n + k_2^* \quad \text{Then} \quad (12)$$

$$IDS^{n+1} - IDS^n = k_1^* + k_2^* \quad (13)$$

By using the equations (11) and (13) we have:

$$C - c = IDS^{n+1} - IDS^n \quad (14)$$

$$D = k_2^* + Y + k_1^* \quad (15)$$

$$D = C - c + Y \quad (16)$$

Now, attacker observes the exchanged public messages, if two consecutive run of protocol satisfy equation **(6)**, then attacker results that n_1, n_2 $mod\ 96 = 0$. Now, he/she finds k_1, k_2, by solving system of equations (5), (11). These values are k_1^n and k_2^n.

$$C - c = k_1^* + k_2^* = 2k_1 + 2k_2 \quad (17)$$

$$A = ROT((ROT(IDS^n + k_1 + c, k_2) + k_1, k_1) \quad (18)$$

Now the attacker has the values of IDS^n, k_1^n, k_2^n, IDS^{n+1}, k_1^{n+1}, k_2^{n+1}. He/she continues the attack as follows. In next session, the tag sends *IDS* for reader and receives $A\|B\|C$ from it. Now, the attacker Using this messages and secret values that he/she gained, computes n_1, n_2 and lets them in *D* for calculating *Y*. It's apparent that computing $\widetilde{n_1}$, n_3, k_1^*, k_2^* values is easy. Now attacker constructs a system of 12 equation 12 unknown using values n_1, *Y* and calculates *ID*.

$$y_1 = (n_{11} * ID_1) + (n_{12} * ID_2) + (n_{13} * ID_3) + (n_{14} * ID_4) + (n_{15} * ID_5) + (n_{16} * ID_6) \quad (19)$$

$$y_2 = (n_{11} * ID_2) + (n_{12} * ID_3) + (n_{13} * ID_4) + (n_{14} * ID_5) + (n_{15} * ID_6) + (n_{16} * ID_7) \quad (20)$$

$$y_3 = (n_{11} * ID_3) + (n_{12} * ID_4) + (n_{13} * ID_5) + (n_{14} * ID_6) + (n_{15} * ID_7) + (n_{16} * ID_8) \quad (21)$$

$$y_4 = (n_{11} * ID_4) + (n_{12} * ID_5) + (n_{13} * ID_6) + (n_{14} * ID_7) + (n_{15} * ID_8) + (n_{16} * ID_9) \quad (22)$$

$$y_5 = (n_{11} * ID_5) + (n_{12} * ID_6) + (n_{13} * ID_7) + (n_{14} * ID_8) + (n_{15} * ID_9) + (n_{16} * ID_{10}) \quad (23)$$

$$y_6 = (n_{11} * ID_6) + (n_{12} * ID_7) + (n_{13} * ID_8) + (n_{14} * ID_9) + (n_{15} * ID_{10}) + (n_{16} * ID_{11}) \quad (24)$$

$$y_7 = (n_{11} * ID_7) + (n_{12} * ID_8) + (n_{13} * ID_9) + (n_{14} * ID_{10}) + (n_{15} * ID_{11}) + (n_{16} * ID_{12}) \quad (25)$$

$$y_8 = (n_{11} * ID_8) + (n_{12} * ID_9) + (n_{13} * ID_{10}) + (n_{14} * ID_{11}) + (n_{15} * ID_{12}) + (n_{16} * ID_1) \quad (26)$$

$y_9 = (n_{11} * ID_9) + (n_{12} * ID_{10}) + (n_{13} * ID_{11}) + (n_{14} * ID_{12}) + (n_{15} * ID_1) + (n_{16} * ID_2)$ (27)

$y_{10} = (n_{11} * ID_{10}) + (n_{12} * ID_{11}) + (n_{13} * ID_{12}) + (n_{14} * ID_1) + (n_{15} * ID_2) + (n_{16} * ID_3)$ (28)

$y_{11} = (n_{11} * ID_{11}) + (n_{12} * ID_{12}) + (n_{13} * ID_1) + (n_{14} * ID_2) + (n_{15} * ID_3) + (n_{16} * ID_4)$ (29)

$y_{12} = (n_{11} * ID_{12}) + (n_{12} * ID_1) + (n_{13} * ID_2) + (n_{14} * ID_3) + (n_{15} * ID_4) + (n_{16} * ID_5)$ (30)

5. Proposed Solutions

In this section we propose efficient countermeasures for existence attacks:

5.1. A Countermeasure for De-synchronization Attack

To address this vulnerability, we propose a simple solution. In our countermeasure, both old and new keys and *IDS* will be stored in the reader side as the tag side. In this case reader and tag have at least one common triple (*IDS*, k_1, k_2) to authenticate each other. We launch the same attack as discussed above on this extended protocol at follows. If we suppose that the initial values of tag and reader are:

Tag : IDS_x , k_{1x}, k_{2x}
Reader : IDS_x , k_{1x}, k_{2x}

***Step 1**- The attacker interrupts message D, so the tag updates its values but the reader doesn't:*

Tag : $IDS^n = IDS_x$, $k_1^n = k_{1x}$, $k_2^n = k_{2x}$
$IDS^{n+1} = IDS_y$, $k_1^{n+1} = k_{1y}$, $k_2^{n+1} = k_{2y}$
Reader : $IDS^n = IDS_x$, $k_1^n = k_{1x}$, $k_2^n = k_{2x}$

Step 2- The tag and the reader run protocol completely:

Tag : $IDS^n = IDS_x$, $k_1^n = k_{1x}$, $k_2^n = k_{2x}$
$IDS^{n+1} = IDS_z$, $k_1^{n+1} = k_{1z}$, $k_2^{n+1} = k_{2z}$
Reader : $IDS^n = IDS_x$, $k_1^n = k_{1x}$, $k_2^n = k_{2x}$
$IDS^{n+1} = IDS_z$, $k_1^{n+1} = k_{1z}$, $k_2^{n+1} = k_{2z}$

Step 3- The attacker and the tag negotiate together:

Tag : $IDS^n = IDS_x$, $k_1^n = k_{1x}$, $k_2^n = k_{2x}$
$IDS^{n+1} = IDS_y$, $k_1^{n+1} = k_{1y}$, $k_2^{n+1} = k_{2y}$
Reader : $IDS^n = IDS_x$, $k_1^n = k_{1x}$, $k_2^n = k_{2x}$
$IDS^{n+1} = IDS_z$, $k_1^{n+1} = k_{1z}$, $k_2^{n+1} = k_{2z}$

In this case even though the attacker has successfully completed all the steps, the tags and database are still synchronized since valid communication can take place using the old values of the keys.

5.2. A Countermeasure for Full disclosure attack

MixBits function in SSL-MAP guarantees if both of its two inputs are zeros mod 96, its output will be zero mod 96. Authors proposed a MixBits function in [15] -as shown in figure 2- that guarantees when two inputs of function are zeros mod 96; its output will not be zero mod 96. Then this modification will enhance the security of the protocol. The extra countermeasures are modifying the structure of some messages or internal states as follows:

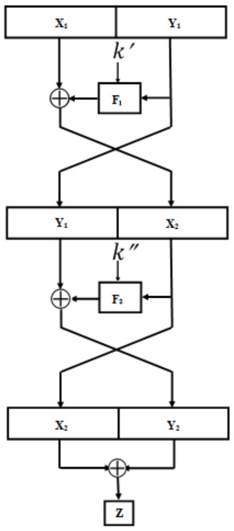


Figure 2. The MixBits Function

In which, F_1 and F_2 are defined as follows:
$F_1 = Rot(Y_1 + k' + c, k' + Y_1)$
$F_2 = Rot(X_1 + k'' + c, k'' + X_1)$
In which, Rot(x, y) in F_1, performs a circular shift on the value x, wh(y) positions to left and Rot(x, y) in F_2, performs a circular shift on value x, (y mod 96) positions to left.

Another solution that can be suggested is that one of the functions rotates the value of “x” y mod 96 positions and another function rotates wh(y) position to compute public messages and internal values with two consecutive rotation functions (Rot(x,y)). By considering this suggests in protocol we see that the aforementioned attack cannot be successful, absolutely. The cause of this claim is that by this way at least one of rotation functions operates on the value of x and causes a nonlinear effect on computing the public messages and internal values.

6. Conclusion

SSL-MAP is an ultra-lightweight Gossamer-based mutual authentication protocol proposed by Rama N and Suganya R. In this paper we showed vulnerability of this protocol by launching two attacks such as de-synchronization and full-disclosure. In de-synchronization attack we destroyed the synchronization between the tag and the reader using an active attack, in three rounds of protocol and in full-disclosure attack, we gained all of secret information shared between tag and reader by eavesdropping several rounds of protocol and analyzing the acquired data. Then, we proposed profit countermeasures and showed that those are efficient against aforementioned attacks.

ACKNOWLEDGEMENTS

I would like to thanks from engineers A. Bahrami and H. Daneshpaye for their kindly encouragements.

REFERENCES

[1] Rama, N. & Suganya R, (2010)."SSL-MAP: More Secure Gossamer Based Mutual Authentication Protocol for Passive RFID Tags". *International Journal on Computer Science and Engineering,* Vol. 02, pp. 363—367.

[2] Peris-Lopez, P. & Hernandez-Castro & J.C., Tapiador & Juan M. E.,Ribagorda, A. , (2009)" Advances in Ultra lightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol". *In Journal of Information Science and Engineering,* Vol. 25 No. 1, pp. 33—57.

[3] Gamal, E.A., Shaaban, E., Hashem, M. (2010)" Lightweight Mutual Authentication Protocol for Low-cost RFID Tags". *International Journal of Network Security & Its Applications (IJNSA)*, Volume 2, Number 2.

[4] Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, Juan M. E.,Ribagorda. (2006)" M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags". *In Proc. of UIC'06, 4159 of LNCS, Springer-Verlag,* pp. 912—923.

[5] Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, Juan M. E.,Ribagorda. (2006)" LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags". *Hand. of Workshop on RFID and Lightweight Crypto.*

[6] Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, Juan M. E.,Ribagorda. A. (2006)" EMAP: An efficient mutual authentication protocol for low-cost RFID tags," *In Proc. of IS'06, volume 4277 of LNCS,* pp. 352--361.

[7] Li, T., Deng, R. (2007)"Vulnerability analysis of EMAP an efficient RFID mutual authentication protocol". *In Proc. of AReS'07.*

[8] Li, T., Wang, G. (2007)" Security analysis of two ultra-lightweight RFID authentication protocols". *In Proc. of IFIP-SEC'07.*

[9] Hung-Yu, C., Chen-Wei, H. ,(2007)" Security of ultra-lightweight RFID authentication protocols and its improvements". *SIGOPS Oper.* Syst. Rev., 41(4), pp. 83–86.

[10] B´ar´asz, M., Boros, B., Ligeti, P., L´oja, K., Nagy, D. (2007)"Breaking LMAP". *In Proc. Of RFIDSec'07.*

[11] B´ar´asz, M., Boros, B., Ligeti, P., L´oja, K., Nagy, D. (2007)"Passive Attack Against theM2AP Mutual Authentication Protocol for RFID Tags". *In Proc. of First Interna-tional EURASIP Workshop on RFID Technology.*

[12] Chien, H.-Y. (2007)"SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity". *IEEE Transactions on Dependable and Secure Computing* 4(4), pp. 337—340.

[13] Tagra, D., Rahman, M., Sampalli, S. (2010)"Technique for Preventing DoS Attacks on RFID Systems". *International conference on Software, Telecommunication and computer networks (softCOM).*

[14] Hernandez-Castro, J.C., Tapiador, Juan M. E., Peris-Lopez, P., Ribagorda. A. (2008), "Cryptanalysis of the SASI Ultra lightweight RFID Authentication Protocol". *IEEE Transactions on Dependable and Secure Computing.*

[15] Kianersi, Mehrdad & Gardeshi, Mahmoud & Arjmand, Mohammad(2011):"SULMA: Secure Ultra Lightweight Mutual Authentication Protocol For Low-Cost RFID Tags", *International Journal of Ubicomp (IJU)*, Vol.2, No.2, pp 17-24.

[16] David, M. & Prasad, N. R. (2010)," Providing Strong Security and High Privacy in Low-Cost RFID Networks," springerlink,.

[17] Lee, Y.-C. & Hsieh, Y.-C. & You, P.-S. & Chen, T.-C(2009)." A New Ultralightweight RFID Protocol with Mutual Authentication", In Proc. of WASE'09, Volume 2 of ICIE, pages 58-61.

[18] P-Lopez & Julio Cesar Hernandez-Castro & Juan M. E. Tapiador & Arturo Ribagorda, (2010) "Security Flaws in a Recent Ultralightweight RFID Protocol," rfid sec asia.

[19] J. Hernandez-Castro1& P.Peris-Lopez & R. C. W. Phan & and Juan M. E. Tapiador," Cryptanalysis of the David-Prasad RFID Ultralightweight Authentication Protocol," RFIDSec'10, Istanbul, June, 2010.

Authors

Mehrdad Kianersi received the Bachelor's degree in Telecommunication Engineering from Islamic Azad University of Najaf abad, Iran, in 2008 and Master's degree in Telecommunication in the field of Cryptography from IHU, Tehran, Iran (2011). Currently, he is research assistant (RA) at the research centre of cryptography, IHU, Tehran, Iran. His research interests includes: Lightweight cryptography, RFID security and authentication protocols.

Mahmoud Gardeshi received his Erudition Degree in applied mathematics from Amir Kabir University, Islamic Republic of Iran in 2000. Currently, he is a researcher at the I. H. University. His research interest includes: cryptography and information security.