CRYPTANALYSIS OF SULMA, AN ULTRA-LIGHTWEIGHT MUTUAL AUTHENTICATION PROTOCOL FOR LOW-COST RFID TAGS

Mahdi Azizi[1] and Nasour Bagheri[2]

[1]Faculty of Communication and Information Technology, IHU University, Tehran, Iran, mmazizi2006@gmail.com

[2]Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran, NBagheri@srttu.edu

ABSTRACT

Recently, Kianersi et al. have proposed an ultra-lightweight mutual authentication protocol for low-cost RFID tags, entitled SULMA. They have claimed that the SULMA protocol is secure against most of the known attacks for an RFID protocol; includes traceability attack, passive attacks and desynchronization. However, in this paper, we analyse the security of SULMA protocol and present several efficient attacks against this protocol. Our attacks include reader impersonation attack and traceability attack. Moreover, we show that the main subcomponent of the protocol called MixBit-function does not satisfy the claimed property.

KEYWORDS

RFID, Mutual authentication protocols, SULMA, security, traceability, Reader Impersonation.

1. INTRODUCTION

Radio frequency Identification (RFID) technology, which is date back to the Second World War, is going to be employed in almost every daily aspects of life [1-4]. In general, an RFID system includes tag, reader and back-end database. Tag is a small chip (normally without power) which have a small memory and antenna to communicate with the reader. This type of tags that is known as passive tags has restricted computational capability and restricted resources. On the other hand, reader is an active device that can communicate with both tag and back-end database. The back-end database can be a power-full computer which can provide the reader with the extra computational capability and storage spaces.

Low-cost RFID can be a good replacement for the most currently extended identification systems, known as barcodes. To mention the main advantages [5] of RFID over barcodes, it should be noted that tag's data can be read automatically, even without line of sight and in the contactless way, at a rate of hundreds of times per second and at a distance of several meters. In addition, tag provides unique identifier for each tagged item, while a barcode only specifies the type of the labelled product.

However, security and privacy could be seen as the main concern in wide spread application of this promising technology.

DOI:10.5121/iju.2011.2402

To address the abovementioned concerns, several approaches have been proposed in the literatures. These proposals can be divided into two general groups. The first group uses blocking, jamming and physical solutions [6,7] while the other group uses cryptographic concepts and privacy preserving protocols. Cryptographic solutions, which is the concern of this paper also, for RFID security issues can be restricted to only lightweight cryptography or let the scheme to even uses the complex cryptographic solutions. Most researchers believe that to replace namely the barcodes by RFID the industry needs simple and low cost RFID tags (below 5 cents per item) which puts a restriction on the number of logical gates [8,9]. For this case, many approaches that are based on the lightweight cryptographic solutions and protocols have been suggested [10-15]. On the other hand, Some RFID researchers, however, believe that it would be possible to use complex cryptographic primitives in future RFID tags. Hence, they suggest the use of advanced encryption standard (AES) [16] or even public key solutions such as elliptic curve cryptography (ECC) [17, 18].

A lightweight protocol has this advantage that it keeps the computational and the price of RFID tags very low. Hence, lightweight protocols have been of interest to both industry and academia and design of secure authentication protocols for low-cost RFID tags has received the attention of a lot of researchers, though many protocols have been published lately [5,6,10-14,19,20]. However, most of them have not satisfied the claimed security goals [21-28].

The Electronic Product Code Class-1 Generation-2 standard specification [29, 30] by EPC Global which also has been ratified by ISO [21] of low-cost RFID tags is an effort on standardization of low cost RFID.

In this work, we perform a security analysis of an ultra-lightweight protocols proposed by Kianersi *et al.* [19], called SULMA. We show that this protocol is vulnerable to some simple security attacks.

Paper Organization: The rest of the paper is organized as follows: In section 2 we introduce SULMA protocol and in section three we analysis its MixBit function. In section four we present our cryptanalysis of protocol. Finally, in section five the closing remarks are given.

2. DESCRIPTION OF THE SULMA PROTOCOL

2. 1 NOTATION

- A, B, C and D: Public messages.
- $A \| B$: Concatenation of two strings A and B.
- R: Reader.
- A : Adversary.
- T_i: i^{th} tag.
- $\bar{0}$: Sequence of 96 zeros.
- c: Constant value *(0x3243F6A8885A308D313198A2)*
- *Rot(X,Y)*: The circular shift *X* to left *Y* positions.
- n_1, n_2 : Random numbers of length 96 bits.
- k_1, k_2 : Secret keys shared between the tag and the reader of length 96 bits.

- Mixbit(.): A block cipher based random function which accepts four inputs each of length 96-bit and produces an 96-bit output value.
- $\oplus$: Bitwise exclusive or.
- +: Modular addition.
- *Hw(.):* Hamming weight (number of ones of an string in GF(2))
- $\vec{x}$: Rotating *x* one bit to the right.
- $B \leftarrow A$: Assigning the value of *A* to *B*.

2. 2 SULMA PROTOCOL OVERVIEW

The SULMA protocol [19], which is depicted in Figure 1, consists of three phases: **identification**, **mutual authentication** and **updating**. Each tag keeps three dynamic pairs (IDS^{old}, IDS^{next}), (k_1^{old}, k_1^{next}), (k_2^{old}, k_2^{next}) and a static ID value. Any tag has a unique ID which is static while the IDS and k_1, k_2 are get updated after each successful run of protocol. The details of protocol are as bellow:

Phase 1: the tag identification phase of protocol is as follow:

1. Reader sends a "*Hello*" message to tag.
2. On receiving the message, the tag response with IDS^{next}.
3. Once the reader received the message, it tries to finding and identical entry in its database. If reader finds related entry in its database it authenticates the tag. Otherwise, it requests for IDS^{old} and continue the protocol.

Phase2: the mutual authentication phase of protocol is as follows:

1. When the reader finds the related entry in Phase 1:
   a. Generates two random numbers n_1 and n_2 such that $n_1, n_2 \neq \bar{0}$.
   b. Computes *A* and *B* as follows, where *c* is a constant value same as the constant value of Gossamer protocol [20] and k_1 and k_2 are the secret keys shared between the tag and the reader:

$$A = Rot(Rot(IDS + k_1 + c + n_1, k_2) + k_1, k_1)$$
$$B = Rot(Rot(IDS + k_2 + c + n_2, k_1) + k_2, k_2)$$

   c. Computes n_3 as follows, where we introduce MixBit in details later:

$$n_3 = MixBit(n_1, n_2, k_1, k_2)$$

   d. Computes *C* and *D* as follows:

$$k_1^* = Rot(Rot(n_2 + k_1 + c + n_3, n_2) + k_2 \oplus n_3, k_1)$$
$$k_2^* = Rot(Rot(n_1 + k_2 + c + n_3, k_2) + k_1 + n_3, n_1)$$
$$n_1' = MixBit(n_3, n_2, k_1^*, k_2^*)$$
$$C = Rot(Rot(n_3 + k_1^* + c + n_1', n_3) + k_2^* \oplus n_1', k_2) \oplus n_1'$$
$$D = Rot(Rot(n_2 + k_2^* + ID + n_1', k_1) + k_1^* + n_1', n_3) + n_1'$$

e. Sends $A \| B \| C$ to the tag.

2. When the tag received the message it does as follows:
   a. Extracts n_1 and n_2 from A and B respectively.
   b. Verifies whether $C \stackrel{?}{=} Rot(Rot(n_3 + k_1^* + c + n_1', n_3) + k_2^* \oplus n_1', k_2) \oplus n_1'$ to authenticate the reader.
   c. If tag authenticated the reader, it computes D as follows:

$$D = Rot(Rot(n_2 + k_2^* + ID + n_1', k_1) + k_1^* + n_1', n_3) + n_1'$$

   d. Sends D to the reader.
   e. Updates its memory.
3. Once the reader received D, it verifies the correctness of D to update its database.

Phase 3: In updating phase of protocol, both reader and tag update their entries, as follows:

$$n_2' = MixBit(n_1', k_1^*, n_3, k_2^*)$$
$$IDS^{old} = IDS$$
$$IDS^{next} = Rot(Rot(n_1' + k_1^* + IDS + n_2', n_1') + k_2^* \oplus n_2', k_1) \oplus n_2'$$
$$k_1^{old} = k_1$$
$$k_1^{next} = Rot(Rot(n_3 + k_2^* + c + n_2', n_3) + k_1^* + n_2', k_2) + n_2'$$
$$k_2^{old} = k_2$$
$$k_2^{next} = Rot(Rot(IDS^{next} + k_1^* + c + k_1^{next}, IDS^{next}) + k_2^* + k_1^{next}, n_2')$$

Hence, $IDS^{old}, IDS^{next}, k_1^{old}, k_1^{next}, k_2^{old}, k_2^{next}$ are dynamic and ID is static.

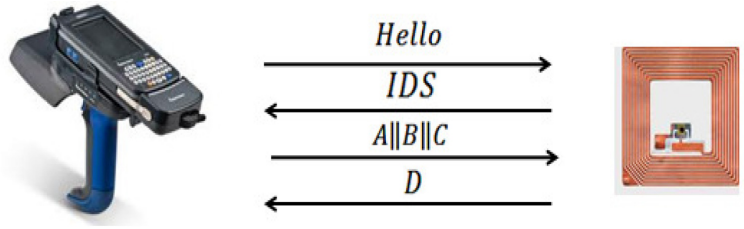

Figure 1:SULMA Protocol [19]

3. ANALYSIS OF MIXBIT FUNCTION

As we have mentioned already, SULMA protocol computation process includes a MixBit function, depicted in Figure 2. The designers have used this function to improve the security of protocol. This function is based on a semi-feistel structure. MixBit function includes two sub-functions denoted by F_1 and F_2 that are as follow:

$$F_1 = Rot(\underbrace{y_1 + k_1 + c}_{x}, \underbrace{k_1 + n_1}_{y = hw(k_1 + n_1)})$$

$$F_2 = Rot(\underbrace{x_2 + k_2 + c}_{x}, \underbrace{k_2 + x_2}_{y = (k_2 + x_2 \bmod 96)})$$

3. 1 On the Randomness of MixBit Function

Designer of the SULMA protocol have stated that the MixBit function is a good random function[19]. To verify their claim, we implemented the MixBit function, and tested 1,048,576 bits of its output to verify whether it passes the standard tests for random functions. We have used the Statistical Test Suite for the random and pseudorandom number generators with 15 tests suggestion by NIST [32]. The results are depicted in Table 1. These results show that MixBit is not a good random function because it failed to pass most of the randomness' tests.

Table 1: The results of Statistical Test of MixBit function

Test Name	Average	χ2prop	Result
Frequency	%0. 00	99. 0000	**Fail**
Frequency within a Block	%0. 00	99. 0000	**Fail**
Runs	%0. 00	99. 0000	**Fail**
Longest Run of Ones in a Block	%0. 00	99. 0000	**Fail**
Binary Matrix Rank	%0. 00	99. 0000	**Fail**
Discrete Fourier Transform	%0. 00	99. 0000	**Fail**
Non Overlapping Template Matching	%0. 00	99. 0000	**Fail**
Overlapping Template Matching	%0. 00	99. 0000	**Fail**
Maurer	%0. 00	99. 0000	**Fail**
Linear Complexity	%0. 00	99. 0000	**Fail**
Serial	%0. 00	99. 0000	**Fail**
Approximate Entropy	%0. 00	99. 0000	**Fail**
Cumulative Sums Forward	%100. 00	0. 0101	**Pass**
Cumulative Sums Backward	%100. 00	0. 0101	**Pass**
Random Excursions	%100. 00	0. 0101	**Pass**
Random Excursions Variant	%100. 00	0. 0101	**Pass**

3. 2 On the Rotational Analysis of MixBit Function

Rotational analysis was first having been proposed to analyse the randomness of cryptographic hash functions [33]. In this analysis, a pair of inputs, known as rotational pair because one message is rotated value of another message, goes through the function and the output pair are analysed in the context of rotational relation. If one of the outputs is the rotated value of the other output with non overwhelming probability, then the given function is vulnerable to the rotational attack. This attack is applicable mainly to those primitives that use rotation, modular addition and *XOR* in their structure, known as ARX primitives. Since MixBit is also an ARX primitive, it worth to investigate whether this function withstand this attack.

The basic idea of the rotational attack is the lemma 1 in [8] which states:

$$pr\left(Rot\left(x+y,r\right)=Rot\left(x,r\right)+Rot\left(y,r\right)\right))=\frac{1}{4}\left(1+2^{r-n}+2^{-r}+2^{-n}\right)$$

Where $|x|=|y|=n$. On the other hand, *XOR* and rotation passes the rotational conditions with the probability of "1". Hence, to determine the total probability of the rotational attack in an ARX function it may be enough to count the number of modular addition in a path started from the input of given function to its output.

Following the above lemma, for two different inputs with one bit circular rotation to the right, *r=1*, we have

$$pr\left(Rot\left(x+y,r\right)=Rot\left(x,r\right)+Rot\left(y,r\right)\right))=\frac{1}{4}\left(1+2^{r-n}+2^{-r}+2^{-n}\right)=0.375=2^{-1.41503}$$

Given that in MixBit function for each of F_1 and F2 includes three modular additions, MixBit function includes six modular additions in total. Hence, for a given rotational input pair the output of MixBit is satisfying the rotational property with the following probability:

$$(2^{-1.41503})^6=2^{-8.5}$$

Hence, given 362 rotational input pairs to MixBit function of SULMA, we expect to receive a rotational pair at output with a high probability. For the more details on rotational attack we suggest interested readers to read [33].

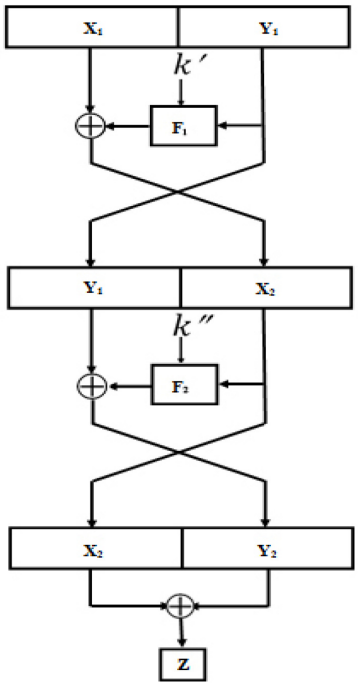

Figure 2:MixBit Function [19]

4. CRYPTANALYSIS OF SULMA PROTOCOL

The authors of SULMA [19] have claimed that the proposed protocol is secure against traceability attack, passive attacks, de-synchronization attack and have forward security. However, in this section we analysis the security of this protocol and demonstrate several weakness on this protocol.

4. 1 READER IMPERSONATION ATTACK

Reader impersonation attack is a forgery attack that leads to identifying a spoofed reader by a legitimate tag as a legal reader. In this section we show how an adversary can deceive the tag to authenticate it as a legitimate reader. In our reader impersonation attack, the adversary, which is an active adversary, can follow the steps described below:

Phase 1 (Learning)**:** in this phase of attack the adversary eavesdrop one session of protocol between the legitimate reader and the tag and stores the transmitted messages includes:

(1) $R \rightarrow T$: *Hello*

(2) $T \rightarrow R$: *IDS*

(3) $R \rightarrow T$: $A \| B \| C$

(4) $T \rightarrow R$: D

Phase 2 (Impersonation)**:** in this phase the adversary (A) supplants the reader and does as follow:

1. A sends a "*Hello*" message to tag.
2. On receiving the message, the tag response with IDS^{next}.
3. Once A received the message, it requests for IDS^{old} and continue the protocol.

1. When A received IDS^{old} it reply with the eavesdropped $A \| B \| C$ from the learning phase:
2. When the tag received the message it does as follows:
    a. Extracts n_1 and n_2 from A and B respectively.
    b. Verifies weather $C \stackrel{?}{=} Rot(Rot(n_3 + k_1^* + c + n_1', n_3) + k_2^* \oplus n_1', k_2) \oplus n_1'$.
    c. If tag authenticated the reader, which it will , it computes D as follows:

$$D = Rot(Rot(n_2 + k_2^* + ID + n_1', k_1) + k_1^* + n_1', n_3) + n_1'$$

    d. Sends D to the reader (here A).
    e. Updates its memory.

Hence, following the above mentioned attack the tag authenticates the adversary as a legitimate reader. The success probability of attack is “1” while the attack’s complexity is only two runs of protocol.

4. 2: TRACEABILITY ATTACK

The authors of SULMA [19] have claimed that SULMA protocol is secure against traceability attack. More precisely, they have stated that since after successful mutual authentication, tag updates its *IDS* and updating procedure contains several random numbers hence *IDS* of tag have

random nature and the adversary cannot identify or trace the tag using it. Although they confirmed that it is possible to trace the tag between two successful mutual authentications, because if the adversary request *IDS* from the tag the tag answers with same *IDS,* but they have stressed that "*after updating IDS value successfully the malicious reader cannot trace it*". However, we present an attack which can trace the tag even after one successful run of protocol and therefore updating the *IDS* values. To trace the tag, adversary A can follow the bellow steps:

Phase 1 (Learning)**:** in this phase, given the target tag *T* that should be traced, the adversary (A) supplants the reader *R* and does as follow:

1. A sends a "*Hello*" message to *T*.
2. On receiving the message, *T* response with its IDS^{next}.
3. Once A stored IDS^{next}, it terminates the protocol.

Phase 2 (Challenge)**:** in this phase, a tag *T'* is given to A and it should decide whether it is the target tag *T*. In this phase of attack the adversary (A) supplants *R* and does as follow:

1. A sends a "*Hello*" message to *T'*.
2. On receiving the message, *T'* response with its IDS'^{next}.
3. Once A received the message, it requests for IDS'^{old}.
4. Once *T'* received the message, it replys with its IDS'^{old}.
5. Once A stored IDS'^{old} , it terminates the protocol

Phase 2 (Decision)**:** based on the received values in the learning and challenge phases of attack, the adversary makes its decision public as follows:

1. A verifies whether IDS'^{next} *or* IDS'^{old} , received in the challenge phase, match the stored *IDS* in the learning phase. If either of them matches *IDS* then A concludes that *T'* is the target tag *T*; otherwise, A concludes that *T'* is not the target tag *T*.

It must be noted that this attack works properly up to only one successful run of protocol. However, it is enough to contradict the designer claim on untraceability of tags. It must be noted if between transient from learning phase to challenge phase of attack, the target tag *T* update its internal state once, then following Table 2, it will assign the stored *IDS* to its IDS^{old}. Otherwise, the stored *IDS* remain as IDS^{next} of *T*. However, it is possible a different tag update its IDS^{old} or IDS^{next} to a value similar to the stored *IDS*, each of these cases happen with the probability of 2^{-n}. In that case, the adversary can output a false decision. On the other hand, in general any random adversary has "50%" chance of output the correct decision. Hence, our given adversary advantage to trace *T* up to one successful update of protocol is as follows, which is not negligible:

$$A^{Adv} = 1 - 2^{-n+1} - 0.5 = 0.5 - 2^{-n+1}$$

The complexity of the given attack is two runs of protocol.

Table 2: The records of *IDS* in the tag and the reader after each run of protocol and its connection to the records of previous and next runs.

No. run	Tag	Reader
1	$T_1 = \{IDS^{old}, IDS^{new_1}\}$	$R_1 = \{IDS^{old}, IDS^{new_1}\}$
2	$T_2 = \{IDS^{old} = IDS^{new_1}, IDS^{new_2}\}$	$R_2 = \{IDS^{old} = IDS^{new_1}, IDS^{new_2}\}$
⋮	⋮	⋮
n	$T_n = \{IDS^{old} = IDS^{new_{n-1}}, IDS^{new_n}\}$	$R_n = \{IDS^{old} = IDS^{new_{n-1}}, IDS^{new_n}\}$

.

5. CONCLUSIONS

In this paper, we have analysed the security of a recently proposed ultra-lightweight mutual authentication protocol for low-cost RFID tags entitled as SULMA[19]. We have presented two efficient attacks against this protocol. The first attack was a reader impersonation attack with the success probability of "1" and the complexity of two runs of protocol. The second attack was a traceability attack which can trace a target, tag even after one successful updating of protocol. The later attack contradicts the claim of designers on the untraceability of SULMA protocol after the successful updating of the internal values.

In addition, we analysed the randomness of a sub function of SULMA called MixBit, which is expected to be a random function. Our analysis on randomness of this function demonstrated that this function cannot pass most of the NIST's randomness tests and it is also easy to find a rotational pair for this function.

Hence, we conclude that the SULMA protocol was not successful to reach its target security level and we suggest the interested researchers to design a secure mutual authentication protocol for RFID systems, which is a crucial need at this point.

ACKNOWLEDGEMENTS

Nasour Bagheri is supported by a grant numbered 1565-90/1/28 from Shahid Rajaee Teacher Training University.

REFERENCES

[1]. Transport for London, " Oyster card, http://www. oystercard. co. uk.

[2]. "Michelin Embeds RFID Tags in Tires", RFID Journal, http://www. rfidjournal. com/article/articleview/ 269 /1/1/.

[3]. J. -H. , Hoepman &E. Hubbers & B., Jacobs & M.& Oostdijk & R. W. , Scherer, (2006) "Crossing borders: Security and privacy issues of the European e-passport", NAME (IWSEC 2006). LNCS, Springer-Heidelberg, vol. 4266 , pp 152–167

[4]. E. -C. Australia, (2008) "Access control, sensor control, and trans-ponders", at: http://www. rfid. com. au/rfid uhf. htm.

[5]. P. Peris-Lopez & J. C. Hernandez-Castro & J. E. Tapiador & J. C. A. van der Lubbe , (201) " Cryptanalysis of an EPC class-1 generation-2 standard compliant authentication protocol", Eng. Appl. of AI, 24(6):1061–1069.

[6]. E. Vahedi & V. Shah-Mansouri & V. Wong & I. F. Blake, (2010) " A probabilistic approach for detecting blocking attack in RFID systems", in Proc. of IEEE ICC, Cape Town, South Africa.

[7]. A, Juels & R. Rivest & M. Szydlo, (2003) " The blocker tag: Selective blocking of RFID tags for consumer privacy", in Proc. of Computer and Communications Security Conf., Washington, DC.

[8]. A, Juels , (2006) " RFID security and privacy: A research survey", IEEE J. on Selected Areas in Communication, vol. 24, no. 2, pp. 381394.

[9]. G. Avoine & P. Oechslin, (2005) " RFID traceability: A multi-layer problem", in Proc. Of Financial Cryptography and Data Security, Roseau, Dominica.

[10]. D. Henrici & P. Muller, (2004) " Hash-based enhancement of location privacy for radiofrequency identification devices using varying identifiers", in Proc. of IEEE PERCOM04, Orlando, FL.

[11]. T. L. Lim& T. Li& &T. Gu, (2008) " Secure RFID identification and authentication with triggered hash chain variants", in Proc. of IEEE ICPADS08, Melbourne, Australia.

[12]. C. C. Tan & B. Sheng & Q. Li, (2008) " Secure and serverless RFID authentication and search protocols", IEEE Trans. on Wireless Communications, vol. 7, no. 4, pp. 1400-1407.

[13]. H. M. Sun & W. C. Ting, (2009) " A Gen2-based RFID authentication protocol for security and privacy", IEEE Trans. on Mobile Computing, vol. 8, no. 8, pp. 1052-1062.

[14]. S. Piramuthu , (2008) " Lightweight cryptographic authentication in passive RFID-tagged systems", IEEE Trans. on Systems, Man, and Cybernetics, vol. 38, no. 3, pp. 360376.

[15]. J. Wu & D. R. Stinson, (2009) " How to improve security and reduce hardware demands of the WIPR RFID protocol", in Proc. of IEEE Intl Conf. on RFID, Orlando, FL.

[16]. M. Feldhofer & S. Dominikus & J.Wolkerstorfer, (2004) " Strong authentication for RFID systems using the AES algorithm", in Proc. of Intl Workshop on Cryptographic Hardware and Embedded Systems, Cambridge, MA.

[17]. Y. K. Lee & K. Sakiyama & L. Batina & I. Verbauwhede, (2008) " Elliptic curve-based security processor for RFID", IEEE Trans. on Computers, vol. 57, no. 11, pp. 1514-1527.

[18]. L. Batina & J. Guajardo & T. Kerins & N. Mentens & P. Tuyls & I. Verbauwhede, (2007) " Public key cryptography for RFID tags, in Proc. of 5th IEEE Intl Conf. on Pervasive Computing and Communications Workshop, White Plains, NY.

[19]. M. Kianersi& M. Gardeshi & M. Arjmand, (2001) "" SULMA: A secure ultra light-weight mutual authentication protocol for lowcost RFID tags" International Journal of UbiComp (IJU), Vol. 2, No. 2, pp. 17-24.

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

[21]. M. Safkhani & N. Bagheri & M. Naderi, (2011) " Vulnerabilities in a new RFID access control protocol". In 6th International Conference on Internet Technology and Secured Transactions (ICITST 2011), Abu Dhabi, UAE.

[22]. M. Safkhani & N. Bagheri & M. Naderi & hamid Behnam, (2011) " On the Security of Wei et al.'s RFID Mutual Authentication Protocol". In DPM'11.

[23]. M. Safkhani & N. Bagheri & M. Naderi & Y. Luo & Q. Chai, (2011) " Tag Impersonation Attack on Two RFID Mutual Authentication Protocols". In ARES, pages 581–584.

[24]. M. Safkhani & N. Bagheri & M. Naderi & S. Sandhya, (2011) " Security analysis of LMAP++, an RFID authentication protocol". In 6th International Conference on Internet Technology and Secured Transactions (ICITST 2011), Abu Dhabi, UAE.

[25]. M. Safkhani & M. Naderi & (2010) " Cryptanalysis and Improvement of a Lightweight Mutual Authentication Protocol for RFID system". In 7th International ISC Con- ference on Information Security and Cryptology 2010(ISCISC'10), pages 57–59.

[26]. M. Safkhani & M. Naderi & N. Bagheri, (2010) " Cryptanalysis of AFMAP". IEICE Electronics Express, 7(17):1240–1245.

[27]. M. Safkhani & M. Naderi & N. Bagheri & S. K. Sanadhya, (2011) "Cryptanalysis of Some Protocols for RFID Systems". Cryptology ePrint Archive, Report 2011/061. http://eprint.iacr.org/.

[28]. M. Safkhani & M. Naderi & H. F.Rashvand, (2010) " Cryptanalysis of the Fast Lightweight Mutual Authentication Protocol (FLMAP)". International Journal of Computer & Communication Technology (IJCCT), 2(2,3,4):182–186.

[29]. (2008) "Class-1 generation 2 UHF air interface protocol standard version 1.2.0", Gen2, 2008. http://www.epcglobalinc.org/standards/.

[30]. (2010) " "EPC Tag data standard version 1.4.2008". http://www.epcglobalinc.org/standards/. Yearly report on algorithms and keysizes, Technical Report D.SPA.13Rev.1.0,ICT-2007-216676,. In Gen2. ECRYPT.

[31]. (2005),"Information technology radio frequency identification for item management. Part 6: parameters for air interface communications at 860 MHz to 960MHz ".http://www.iso.org.

[32]. A. Rukhin & J. Soto & J. Nechvatal & M. Smid & E. Barker & S. Leigh & M. Levenson & M. Vangel & D. Banks & A. Heckert & J. Dray & S. Vo,(2000) "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" , NIST SP-800-22.

[33]. D. Khovratovich & I. Nikoli ,(2010)" Rotational Cryptanalysis of ARX", FSE 2010, pp. 333-346.

Authors :

Mahdi Azizi received his M.S. degree in Communications, Cryptology &Information Security in 2005. Currently, he is a Ph.D. candidate at the Department of Information and Communication Technology I.H University, Tehran, Iran. His research interests include RFID security, authentication protocols and Cryptanalysis.

Nasour Bagheri is a lecturer at Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of 20 articles in information security and cryptology. Homepage of the author is available at: http://n-bagheri.srttu.ir