On the Traceability of Tags in SUAP RFID Authentication Protocols

Masoumeh Safkhani[1], Nasour Bagheri[2], Majid Naderi[1]

[1] Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran.
{M_Safkhani,M_Naderi}@iust.ac.ir
[2] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran.
Nbagheri@srttu.ac.ir

Abstract. Widespread adoption of RFID technology in all aspects of our life mainly depends on the fixing the privacy concerns of this technology's customers. Using a tagged object should not lead to existence of the tracing possibility. This concern is a challenging issue that has motivated the researchers to propose several authentication protocols to fix the traceability problem in RFID systems and also provide other security requirements.
In this paper, we analyze the security of three authentication protocols which have recently been proposed by Morshed *et al.* [8]. Our security analysis clearly highlights important security pitfalls in these protocols which leads to their vulnerability against traceability. The complexity of the proposed attacks are only several runs of the protocols while the adversary's advantages to trace the tagged object are maximal.

Keywords: RFID, Ubiquitous, Mutual Authentication, Traceability.

1 Introduction

Radio frequency identification (RFID) is a favorite technology for automated identification in various applications, e.g., libraries, supply chain management, e-passports, human implants and toll payment and we may become as dependent on that in the foreseeable future as we are dependent on e-mail or cellular phones today. The tag, the reader and the back-end server are three basic components of an RFID system:

- Tags are connected to the objects that are supposed to be identified by the reader through radio frequency signals.
- The reader can read or modify tag's information.
- The back-end server aids the reader by an extra storage spaces and further computational capability. In addition, it is much more reliable to keep the valuable data of all tags in back-end server and transfer the necessary data of a particular tag, in case of request, to the reader which prevents the loss of all data in case of reader theft.

Low cost RFID tags are increasingly being deployed in various practical applications nowadays. Security analysis of the way these tags are used in an application is a requirement for the successful adoption of the RFID technology. Depending on the requirements of the particular application, security demands on tags may cover some or all of security aspects such as confidentiality, integrity and availability. However, privacy is a concern which should be satisfied in any application. On the other hand, traceability of a tag is an attack which compromises the tagged object's privacy. Hence, any RFID protocol should resist against tag's traceability.

As a result of increasing deployment of RFID tags, many works on RFID protocols and their security analysis have appeared in the literature [1–4, 6–15, 17, 18] in the past few years. Among them, recently Morshed *et al.* in [8] have proposed three protocols, called SUAP1, SUAP2 and SUAP3 based on an approach which utilizes two very different but widely known approaches to design an RFID protocol, i.e. the "'low-cost authentication protocol (LCAP)" [16] approach and the "one-way hash-based LCAP (OHLCAP)" [5] approach, and claimed that their protocols are more secure than the other schemes. However, in this work we investigate the security of the SUAP protocols and show that these protocols do not satisfy the security notation of traceability. We propose a traceability attack which works for all three variants of SUAP. The proposed attack on these protocols is highly efficient, has the success probability of almost "1" and can be performed on the cost of 16 runs of the learning phase of protocol and only one run of the on-line phase protocol.

Paper Organization: We give a brief description of SUAP protocols(i.e. SUAP 1, SUAP2 and SUAP3) in section 2. Section 3 explains the proposed traceability attack against the three variants of SUAP. Finally, in section 4 we extract some conclusions.

R_i	The i^{th} RFID reader
T_i	The i^{th} RFID tag
ID	Unique identifier of T_i
x	Common secret of tags
GID	Group identifier
$h(.)$	A one-way hash function, $h : \{0,1\}^* \longrightarrow \{0,1\}^l$
h_L	The left half of hash value
h_R	The right half of hash value
t	A temporary variable
Had	Hash address which equals $h(ID)$
N	Number of tags
n	Number of groups
m_i	Number of tags in the i-th group
l	The length of an identifier which is assumed to be 96 bits
r_1 and r_2	l bit random numbers
$\oplus$	XOR operation
$\|\|$	Concatenation operation
$A \rightarrow B$	Sending a message from A to B
$(X)_i$	i^{th}-bit of string X, where the least significant bit(LSB) of X is denoted by $(X)_0$
$\{0\}^x$	A string of zeros of length x-bit
$X\|_{b-a}$	A fraction of string X includes bit b to bit a, where $a > b$.

Table 1. Notation

2 Protocols Description

2.1 SUAP1

Based on SUAP1 designers' claims, the objective of this protocol is to preserve the ubiquitous property of the protocol to be applicable for an RFID system with small number of tags. In this protocol there is an assumption that a common secret x and the tag's identifier ID are stored in the tag and the back-end

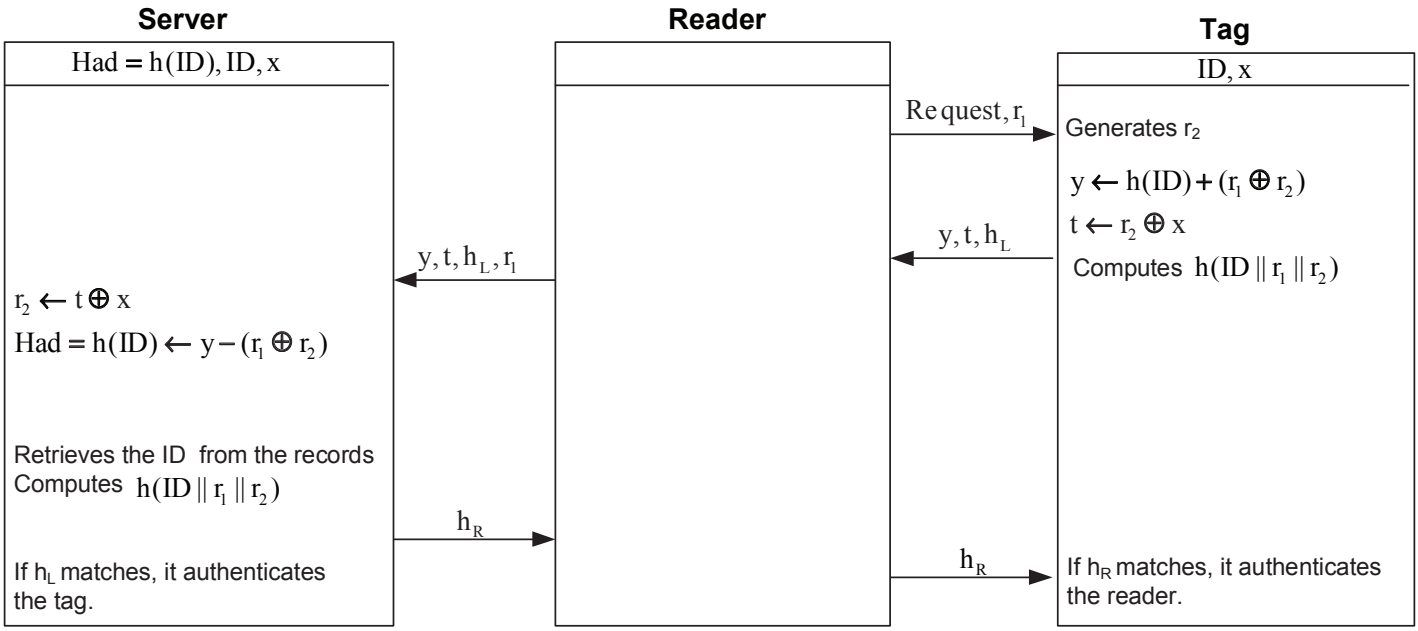

Fig. 1. SUAP1 authentication protocol proposed by Morshed *et al.* [8].

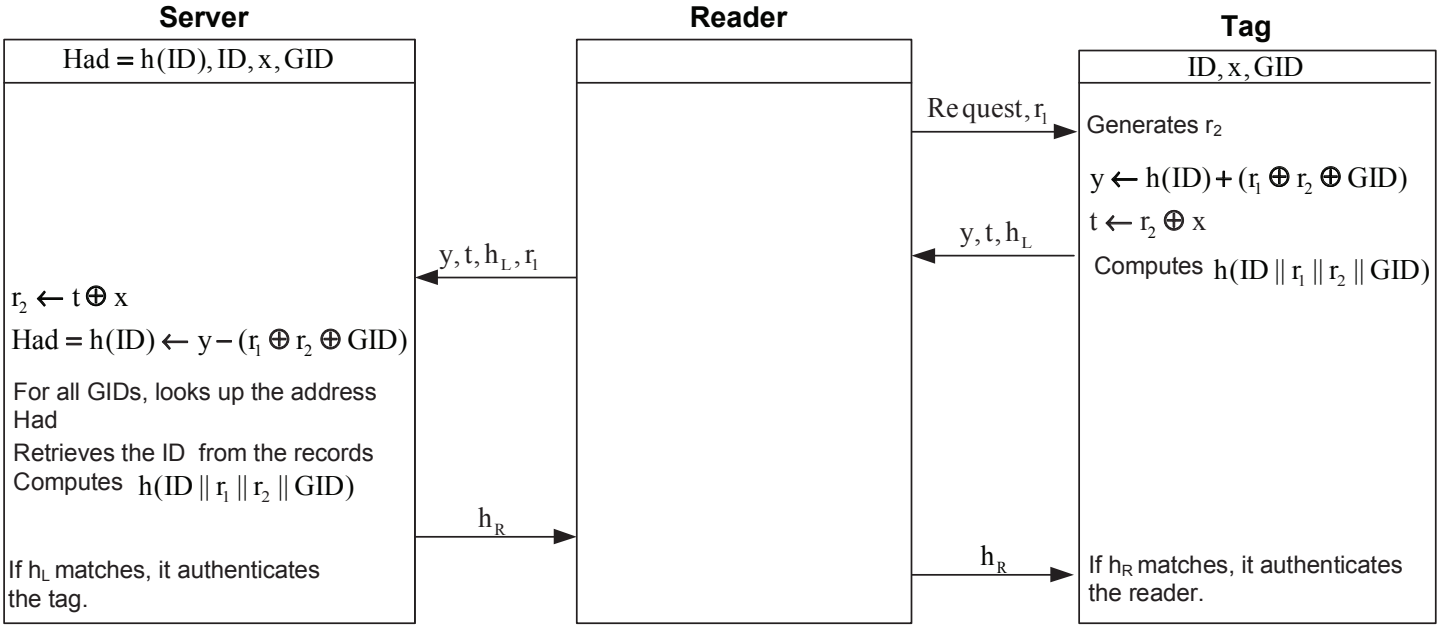

Fig. 2. SUAP2 authentication protocol proposed by Morshed *et al.* [8].

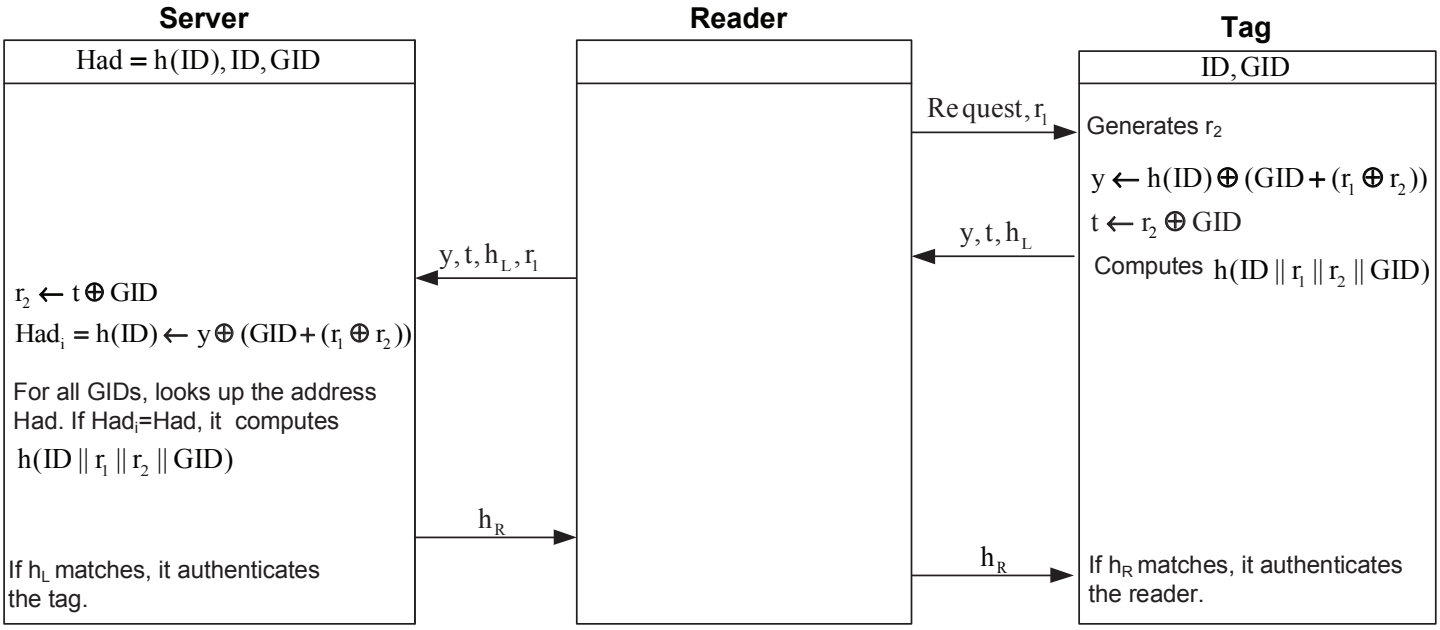

Fig. 3. SUAP3 authentication protocol proposed by Morshed *et al.* [8].

database keeps the tag's identifier ID, common secret number x and hash address $Had = h(ID)$ for each tag. SUAP1, which is depicted in Fig. 1, proceeds as below:

1. The reader generates a random number r_1 and sends it to the tag.
2. Once receipt the message, the tag generates another random number r_2. If r_1 or r_2 equals 0, the protocol aborts. Otherwise, the tag does as follows:
   - computes $y = h(ID) + (r_1 \oplus r_2)$, $t = r_2 \oplus x$ and $h(ID\|r_1\|r_2)$.
   - sends y, t and the left half of the computed hash value, i.e. h_L, to the reader.
3. The reader then sends y, t, h_L and r_1 to the back-end database.
4. On receipt the message, the back-end database does as follows:
   - retrieves r_2 as $t \oplus x$.
   - retrieves Had, i.e. $h(ID)$, as $y - (r_1 \oplus r_2)$ where Had is the address of the record containing the ID.
   - retrieves ID from the record.
   - computes $h(ID\|r_1\|r_2)$.
   - compares the left half of the computed value of $h(ID\|r_1\|r_2)$ by the received value of h_L. If they are the same, it authenticates the tag and sends h_R to the reader where h_R is the right half of $h(ID\|r_1\|r_2)$.
5. The reader forwards h_R to the tag.
6. Upon receiving the message, the tag compares the received h_R with the computed value by itself. In the case of equality, the tag authenticates the reader.

Morshed *et al.* have stated that SUAP1 is mainly suitable for an RFID system with small number of tags. However, it is an important concern to have only a single secret x for all the tags in a large organization and this protocol should be avoided in such applications.

2.2 SUAP2

To overcome the problem of SUAP1, Morshed *et al.* have proposed SUAP2 which is suitable for large number of tags. In this protocol they have assumed that the back-end data base divides tags to n groups and the tag's identifier ID, secret number of a group x and one extra variable which presents a group identifier GID is stored in the tag side and the database side. The back-end database also keeps $Had = h(ID)$ as a address of the record containing the tag's ID. The steps of the SUAP2, as depicted in Fig. 2, are as follows:

1. The reader generates a random number r_1 and sends it to the tag.
2. Once receipt the message, the tag generates another random number r_2. If r_1 or r_2 equals 0, the protocol aborts. Otherwise, the tag does as follows:
   - computes $y = h(ID) + (r_1 \oplus r_2 \oplus GID)$, $t = r_2 \oplus x$ and $h(ID\|r_1\|r_2\|GID)$.
   - sends y, t and the left half of the computed hash value, i.e. h_L, to the reader.
3. The reader then sends y, t, h_L and r_1 to the back-end database.
4. On receipt the message, the back-end database does as follows:
   - retrieves r_2 as $t \oplus x$.
   - retrieves Had, i.e. $h(ID)$, as $y - (r_1 \oplus r_2 \oplus GID)$ where Had is the address of the record containing ID.
   - looks up the address Had.
   - retrieves ID from the record.
   - computes $h(ID\|r_1\|r_2\|GID)$.
   - compares the left half of the computed value of $h(ID\|r_1\|r_2\|GID)$ by the received value of h_L. If they are the same, it authenticates the tag and sends h_R to the reader where h_R is the right half of $h(ID\|r_1\|r_2\|GID)$.
5. The reader forwards h_R to the tag.
6. Upon receiving the message, the tag compares the received h_R with the computed value by itself. In the case of equality, the tag authenticates the reader.

2.3 SUAP3

For enhancing the SUAP2 efficiency, Morshed *et al.* have proposed SUAP3 in which the only difference compared to SUAP2 is that SUAP3 does not use the secret x for the tag and the database. SUAP3, which is depicted in Fig. 3, proceeds as below:

1. The reader generates a random number r_1 and sends it to the tag.
2. Once receipt the message, the tag generates another random number r_2. If r_1 or r_2 equals 0, the protocol aborts. Otherwise, the tag does as follows:
   - computes $y = h(ID) \oplus (GID + (r_1 \oplus r_2))$, $t = GID \oplus r_2$ and $h(ID\|r_1\|r_2\|GID)$.
   - sends y,t and the left half of hash value, i.e. h_L, to the reader.
3. The reader then sends y, t,h_L and r_1 to the back-end database.
4. On receipt the message, the back-end database does as follows:
   - retrieves r_2 as $t \oplus GID$.
   - retrieves Had_i as $y \oplus (GID + (r_1 \oplus r_2))$, where $Had_i = h(ID)$ is the address of the record containing the ID.
   - looks up the address Had_i.
   - retrieves ID from the record if $Had_i = Had$ for any ID.
   - computes $h(ID\|r_1\|r_2\|GID)$.
   - compares the left half of the computed value of $h(ID\|r_1\|r_2\|GID)$ by the received value of h_L. If they are the same, it authenticates the tag and sends h_R to the reader, where h_R is the right half of $h(ID\|r_1\|r_2\|GID)$.
5. The reader forwards h_R to the tag.
6. Upon receiving the message, the tag compares the received h_R with the computed value by itself. In the case of equality, the tag authenticates the reader.

3 Traceability Attack

Morshed *et al.* have claimed that the using of two random numbers in their protocols make the transferred messages unpredictable so that it is impossible to perform tracing attack by a malicious reader. However, in this section we present an efficient traceability attack against all versions of SUAP. The proposed attack is based on the following observations:

1. Assume that $A = a_{l-1}\| \dots \|a_1\|a_0$, $r = r_{l-1}\| \dots \|r_1\|r_0$ and $r' = r'_{l-1}\| \dots \|r'_1\|r'_0$ are strings each of l-bit where (e.g.) r_i denotes the i^{th} bit of r. For $Y = A + r$ and $Z = A + r'$, if $r_i = r'_i$ for $0 \leq j \leq i$ then $Y_i = Z_i$ for $0 \leq j \leq i$ and vice versa.

Hence, e.g. in SUAP1 where $t = r_2 \oplus x$, if $t_i = t'_i$ for $0 \leq j \leq i$, where $t' = r'_2 \oplus x$, then we can conclude that $r_{2\,i} = r'_{2\,i}$ for $0 \leq j \leq i$ and vice versa.

Given the above observation, to trace the target tag T_i in SUAP1, SUAP2 or SUAP3, the adversary $\mathcal{A}$ does as follows:

Phase 1(Learning) : the adversary $\mathcal{A}$ creates a table Tab includes N rows, chooses $r_1 = 1\|\{0\}^{l-1}$, where $\{0\}^{l-1}$ denotes a string of zeros of length $(l-1)$-bit, and supplants N sessions with T_i as follows, for $1 \leq j \leq N$:

1. $\mathcal{A}$ sends r_1 to the tag.
2. Once receipt the message, the tag generates a random number r_2^j and if $r_2^j \neq 0$ it does as follows:
   - computes y^j, t^j and h^j,
   - sends y^j,t^j and the left half of the computed hash value, i.e. h_L^j, to the reader which is supplanted by $\mathcal{A}$.
3. $\mathcal{A}$ stores y^j and t^j in the j^{th} row of Tab.

Phase 2 (Execution) : Given T'_i the adversary $\mathcal{A}$ creates a table Tab' includes N rows, chooses $r_1 = 1\|\{0\}^{l-1}$, where $\{0\}^{l-1}$ denotes a string of zeros of length $(l-1)$-bit, and supplants N' sessions with T_i as follows, for $1 \leq f \leq N'$:

1. $\mathcal{A}$ sends r_1 to the tag.
2. Once receipt the message, the tag generates a random number r_2^f and if $r_2^f \neq 0$ it does as follows:
   - computes y^f, t^f and h^f,
   - sends y^f,t^f and the left half of the computed hash value, i.e. h_L^f, to the reader which is supplanted by $\mathcal{A}$.
3. $\mathcal{A}$ stores y^f and t^f in the f^{th} row of Tab'.

Phase 3 (Decision): To decide whether T'_i is the target tag T_i, if $\exists(((y^j, t^j) \in Tab) \ \wedge ((y^f, t^f) \in Tab')) \ \big| (t^j|_{0-k} = t^f|_{0-k}) \wedge (y^j|_{0-k} \neq y^f|_{0-k})$ then $T_i \neq T'_i$, , for $0 \leq j \leq N$ and $0 \leq f \leq N'$; otherwise, $T_i = T'_i$.

The total complexity of the given attack is N sessions in the learning phase of attack plus N' sessions in the execution phase of attack. The adversary's advantage Adv_A to make the correct decision in the third phase of attack can be determined as follows:

$$Adv_A = \left| Pr[A^{T_i = T'_i} \Rightarrow 1] - Pr[A^{T_i \neq T'_i} \Rightarrow 1] \right|$$

To determine Adv_A one can state that:

1. For any entry t^j in Tab', for $1 \leq j \leq N'$, we denote the number of entries in Tab such that $(t^j|_{0-(k-1)} = t^f|_{0-(k-1)}) \wedge (t_k^j \neq t_k^f)$ by M_k^j, for $1 \leq k \leq l$.
2. The expected value of M_k^j is $\frac{N}{2^{k+1}}$.

3. Following the given observation, if $t^j|_{0-(k-1)} = t^f|_{0-(k-1)}$ then the adversary can conclude that $(r_2^j)|_{0-(k-1)} = (r_2^f)|_{0-(k-1)}$ and it expects to receive $y^j|_{0-(k-1)} = y^f|_{0-(k-1)}$. This condition is satisfied for $T_i = T_i'$ with the probability of "1" and for $T_i \neq T_i'$ with the probability of "2^{-k}".
4. Hence, the probability of the wrong alarm Pr_{wrong} is determined as follows:

$$Pr_{wrong} \leq \left(\prod_{k=1}^{l}(2^{-k})^{M_k^j}\right)^{N'} = \left(\prod_{k=1}^{l}(2^{-k})^{\frac{N}{2^{k+1}}}\right)^{N'}$$

5. The adversary's advantage to trace the target tag successfully is as follows:

$$Adv_A = \left|Pr[A^{T_i=T_i'} \Rightarrow 1] - Pr[A^{T_i\neq T_i'} \Rightarrow 1]\right| \geq 1 - \left(\prod_{k=1}^{l}(2^{-k})^{\frac{N}{2^{k+1}}}\right)^{N'}$$

Following the given procedure the adversary's advantage to distinguish the given tag from the target tag is non-negligible. As an example, for $N = 16$ (which can be considered as the off-line phase of the attack) and $N' = 1$ (which can be considered as the on-line phase of the attack) and doing some numerical calculation we have $Adv_A \geq 1 - 2^{-14}$. Hence, even for the on-line complexity of only one run of the protocol, the success probability of the given attack is almost "1". For $N = N' = 16$ we have $Adv_A \geq 1 - (2^{-14})^{16} = 1 - 2^{-224}$ which is almost "1". An interesting point of this attack is that it works for SUAP1, SUAP2 and SUAP3 and even the adversary does not require to know which protocol the target tag uses.

4 Conclusions

In this paper we have shown that recently proposed RFID authentication protocols by Morshed *et al.* fails to provide adequate security against traceability attacks. In this paper we presented an attack which can trace the tag whenever it uses either of the suggested protocols by Morshed *et al.*, i.e. SUAP1, SUAP2 and SUAP3.

References

1. Y.-Y. Chen, M.-L. Tsai, and J.-K. Jan. The design of rfid access control protocol using the strategy of indefinite-index and challenge -response. *computer communication*, 34(3):250–256, 2011.
2. H.-Y. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, December 2007.
3. J.-S. Cho, S.-S. Yeo, and S. K. Kim. Securing against brute-force attack: A hash-based rfid mutual authentication protocol using a secret value. *Computer Communications*, 34(3):391–397, 2011.
4. D. N. Duc and K. Kim. Defending rfid authentication protocols against dos attacks. *Computer Communications*, 34(3):384–390, 2011.
5. EY Choi and SM Lee and DH Lee. Efficient RFID authentication protocol for ubiquitous computing environment. In *Embed. Ubiquit. Comput.*, volume 3832, pages 945–954, 2005.
6. H.Y.Chien. Secure access control schemes for rfid systems with anonymity. In *Proceedings of the 7th International Conference on Mobile Data Management (MDM 2006)*, page 96, 20O6.
7. Y. Luo, Q. Chai, G. Gong, and X. Lai. A lightweight stream cipher WG-7 for RFID encryption and authentication. In *GLOBECOM*, pages 1–6. IEEE, 2010.
8. M. M. Morshed, A. Atkins, and H. Yu. Secure ubiquitous authentication protocols for rfid systems. *EURASIP Journal on Wireless Communications and Networking 2012, 2012:93 doi:10.1186/1687-1499-2012-93*, pages 1–35, 2012.

9. P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. LMAP:A Real Lightweight Mutual Authentication Protocol for Low cost RFID tags. In *RFIDSec*, 2006.
10. P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol. In *WISA*, pages 56–68, 2008.
11. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags. In *OTM Federated Conferences and Workshop: IS Workshop – IS'06*, volume 4277 of *Lecture Notes in Computer Science*, pages 352–361, Montpellier, France, November 2006. Springer-Verlag.
12. M. Safkhani, N. Bagheri, and M. Naderi. Vulnerabilities in a new RFID access control protocol. In *6th International Conference on Internet Technology and Secured Transactions (ICITST 2011)*, Abu Dhabi, UAE, Dec. 2011.
13. M. Safkhani, N. Bagheri, M. Naderi, Y. Luo, and Q. Chai. Tag Impersonation Attack on Two RFID Mutual Authentication Protocols. In *FARES*, 2011.
14. M. Safkhani, N. Bagheri, M. Naderi, and S. Sandhya. Security analysis of LMAP++, an RFID authentication protocol. In *6th International Conference on Internet Technology and Secured Transactions (ICITST 2011)*, Abu Dhabi, UAE, Dec. 2011.
15. M. Safkhani, N. Bagheri, S. Sandhya, and M. Naderi. On the security of mutual authentication protocols for rfid systems : the case of wei *et al.*'s protocol. In *DPM 2011, LNC*, volume 7122 of *LNCS*, 2011.
16. SM Lee and YJ Hwang and DH Lee and JI Lim. DEfficient authentication for low-cost RFID systems. In *in ICCSA05, LNCS*, volume 3480, pages 619–629, (Springer-Verlag, 2005.
17. S.Weis, S.Sarma, R.Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *1st International Conference on Wierless Communications, Networking and mobile computing 2007(WiCom 2007)*, pages 2078–2080, 2007.
18. T.-C. Yeh, C.-H. Wu, and Y.-M. Tseng. Improvement of the rfid authentication scheme based on quadratic residues. *Computer Communications*, 34(3):337–341, 2011.