On the Traceability of Tags in SUAP RFID Authentication Protocols

Masoumeh Safkhani*, Nasour Bagheri†, Pedro Peris-Lopez‡ and Aikaterini Mitrokotsa§
*Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran
Email: M_Safkhani@iust.ac.ir
†Department of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran
Email: Nbagheri@srttu.ac.ir
‡Computer Security Lab (COSEC), Carlos III University of Madrid, Spain
Email: pperis@inf.uc3m.es
§EPFL, Lausanne, Switzerland,
Email:katerina.mitrokotsa@epfl.ch

***Abstract*—RFID technology is one of the most pervasive computing technologies with important advantages and a wide range of applications. Nevertheless, the widespread adoption of RFID technology mainly depends on fixing the security and privacy concerns of this technology. Using a tagged object should not lead to the traceability of this object. This concern is a challenging issue that has motivated the proposal of several authentication protocols that attempted to fix the traceability problem.**

In this paper, we analyze the security of three authentication protocols that have been recently proposed by Morshed *et al.* [2]. Our security analysis clearly highlights important security pitfalls in these protocols that lead to their vulnerability against traceability. The proposed attacks require only several runs of the protocols while the adversary's advantages to trace the tagged object are maximal.

I. Introduction

Radio frequency identification (RFID) is a prominent technology for automated identification with various applications, e.g., supply chain management, e-passports, human implants and toll payment. RFID systems consist of RFID tags, RFID readers and a back-end database.

- The *RFID tags* are connected or embedded to the objects that are supposed to be identified by the RFID reader.
- The *RFID reader* reads though radio frequency signals the RFID tags and may also able to modify the tags' information.
- The *back-end database* provides extra storage space where additional information about the tagged objects may be stored. Obviously, it is much more reliable to keep the valuable data of all tags in back-end database and transfer only the necessary data of a particular tag, in case of request, to the reader.

Low cost RFID tags are increasingly being deployed in various applications. Nevertheless, high security and privacy concerns are raised depending on the application. Traceability of RFID tags is an important issue that should be avoided in order not to violate the privacy rights of the parties carrying or using the tagged objects.

As a result of the increased deployment of RFID tags, a broad range of RFID authentication protocols have been proposed. Among them, recently Morshed *et al.* in [2] have proposed three protocols, called SUAP1, SUAP2 and SUAP3 based on an approach which uses two very different but widely known approaches to design an RFID protocol, i.e. the "low-cost authentication protocol (LCAP)" [3] approach and the "one-way hash-based LCAP (OHLCAP)" [1] approach, and claimed that their protocols are more secure than other existing schemes. However, in this work we investigate the security of the SUAP protocols and show that these protocols are vulnerable to traceability. We describe a traceability attack which can be deployed against all three variants of the SUAP protocol. The proposed attack on these protocols is highly efficient, has success probability almost equal to 1 and can be performed on the cost of 16 runs of the *learning phase* of the protocol and only one run of the *on-line phase* of the protocol.

Paper Organization: In section II we give a brief description of the SUAP protocols (i.e. SUAP 1, SUAP 2 and SUAP 3). In Section III we explain the proposed traceability attack against the three variants of the SUAP protocols. Finally, section IV concludes the paper.

II. Protocols Description

A. SUAP1

Based on SUAP1 designers' claims, this protocol can be used in an RFID system where a small number of RFID tags is employed. In this protocol, a common secret x and the tag's identifier ID are stored in the tag and the back-end database keeps the tag's identifier ID, the common secret number x and the hash address $Had = h(ID)$ for each tag. The protocol SUAP1 (depicted in Fig. 1) can be summarised as follows:

1) The reader generates a random number r_1 and sends it to the tag.
2) After receiving r_1, 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:

978-1-4673-0328-6/12/$31.00 ©2012 IEEE

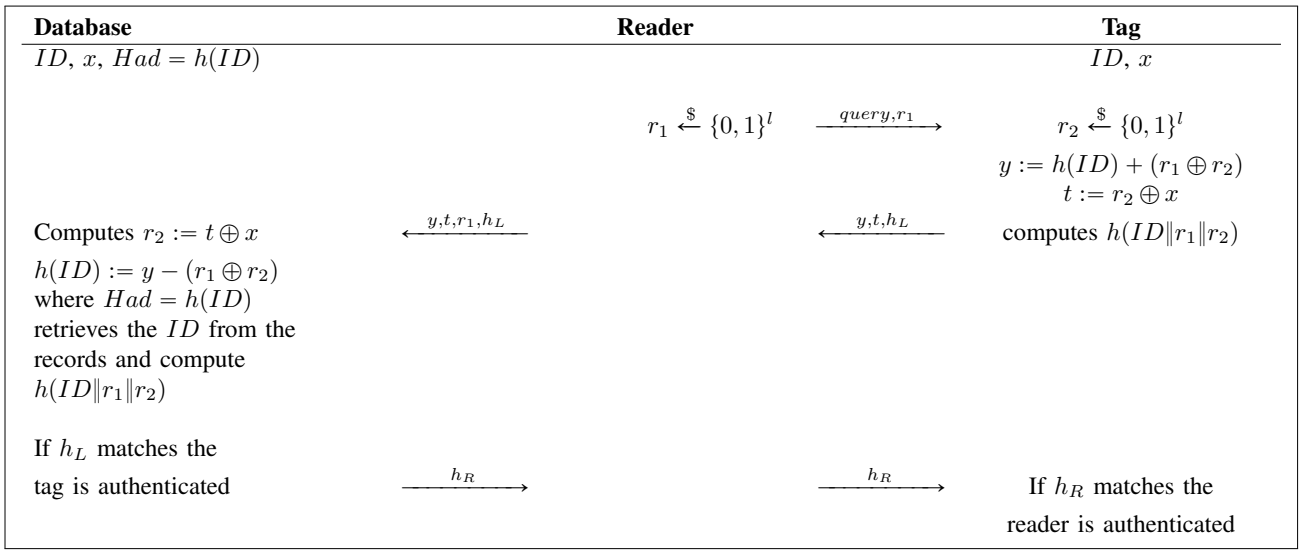

Fig. 1. The SUAP1 authentication protocol proposed by Morshed *et al.* [2].

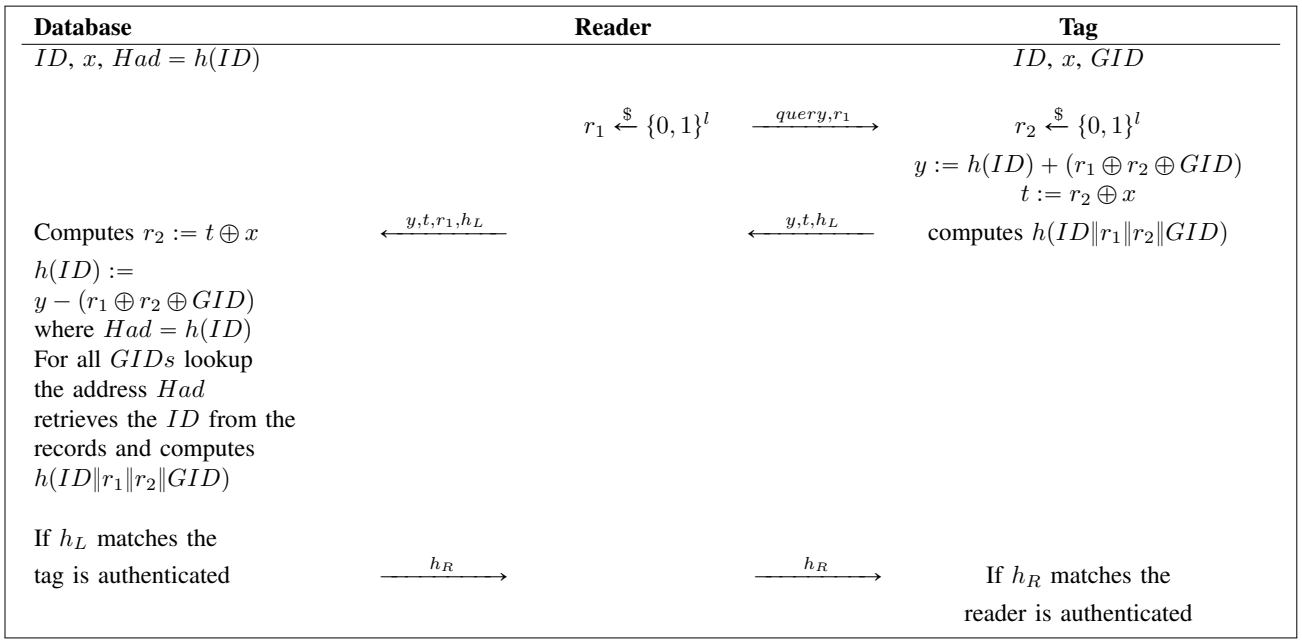

Fig. 2. The SUAP2 authentication protocol proposed by Morshed *et al.* [2].

- it computes $\begin{cases} y = h(ID) + (r_1 \oplus r_2) \\ t = r_2 \oplus x \\ h(ID \| r_1 \| r_2) \end{cases}$
- it 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) After receiving the values y, t, h_L and r_1, the back-end database:
   - 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 h_R, the tag compares the received value with the computed value. If they match, the tag authen-

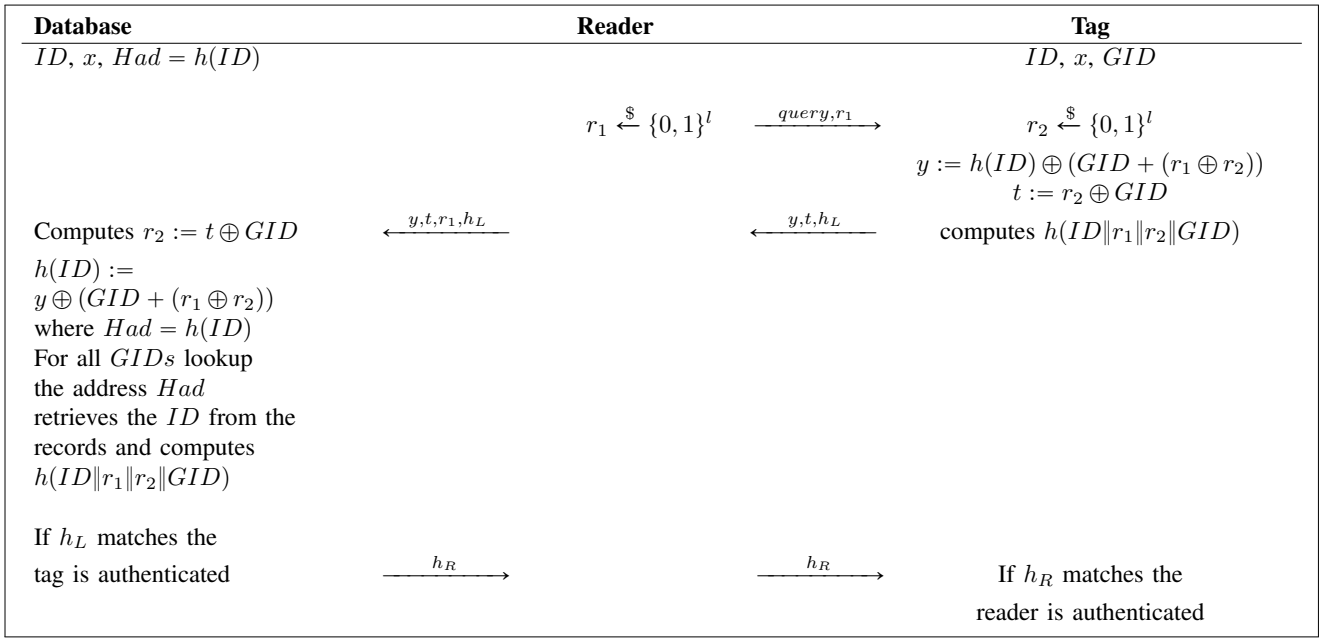

Fig. 3. The SUAP3 authentication protocol proposed by Morshed *et al.* [2].

ID	Unique identifier of the RFID tag
x	Common secret key of the RFID tags
GID	Group identifier
$h(.)$	A one-way hash function, $h : \{0,1\}^* \longrightarrow \{0,1\}^l$
h_L	The left half of a hash value
h_R	The right half of a 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	Random numbers with length l bits
$\oplus$	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-bits
$X\|_{b \sim a}$	A fraction of string X includes bit b to bit a, where $a > b$.
X^i	The value of string X at the i^{th} run of protocol.

TABLE I
NOTATION

ticates the reader.

Morshed *et al.* have stated that SUAP1 is mainly suitable for an RFID system with a small number of tags. We should make clear that 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.

B. SUAP2

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

1) The reader generates a random number r_1 and sends it to the tag.
2) After receiving r_1, the tag generates another random number r_2. If r_1 or r_2 equals 0, the protocol aborts. Otherwise, the tag:
   - 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) After receiving these values, the back-end database:
   - 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) After receiving h_R, the tag compares the received h_R with the computed value. If they match, the tag authenticates the reader.

C. 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 key x shared between the tag and the database. SUAP3, which is depicted in Fig. 3, is summarized below:

1) The reader generates a random number r_1 and sends it to the tag.
2) After receiving r_1, the tag generates another random number r_2. If r_1 or r_2 equals 0, the protocol aborts. Otherwise, the tag:
   - 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 the hash value $h(ID \| r_1 \| r_2 \| GID)$, denoted by h_L, to the reader.
3) The reader then sends y, t,h_L and r_1 to the back-end database.
4) After receiving these values, the back-end database:
   - 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 h_R, the tag compares the received h_R with the computed value. If these values are equal, the tag authenticates the reader.

III. Traceability Attack

Morshed *et al.* have claimed that the use of two random numbers make the transferred messages unpredictable and thus, the protocol is not vulnerable to a tracing attack. However, in this section we present an efficient traceability attack against all versions of the SUAP protocols. The proposed attack is based on the following observations:

1) Assume that:
   $A = (A)_{l-1} \| \dots \| (A)_1 \| (A)_0$,
   $V = (V)_{l-1} \| \dots \| (V)_1 \| (V)_0$ and
   $W = (W)_{l-1} \| \dots \| (W)_1 \| (W)_0$
   are strings each of l-bits where $(V)_i$ denotes the i^{th} bit of V.
   a) Assume that $\mathcal{X} = A + V$ and $\mathcal{Y} = A + W$: if $V|_{0\sim i} = W|_{0\sim i}$ then $\mathcal{X}|_{0\sim i} = \mathcal{Y}|_{0\sim i}$, for any value of A.
   b) Assume that $\mathcal{X} = A \oplus V$ and $\mathcal{Y} = A \oplus W$: if $V|_{0\sim i} = W|_{0\sim i}$ then $\mathcal{X}|_{0\sim i} = \mathcal{Y}|_{0\sim i}$ and vice versa, for any value of A.

Hence, e.g. in SUAP1 where $t = r_2 \oplus x$ and $t' = r_2' \oplus x$, if $t|_{0\sim i} = t'|_{0\sim i}$, then we can conclude that $r_2|_{0\sim i} = r_2'|_{0\sim i}$ and vice versa.

Given the above observation, to trace the target tag T_i in the protocols SUAP1, SUAP2 or SUAP3, the adversary $\mathcal{A}$ performs the following steps:

Phase 1 (Learning): The adversary $\mathcal{A}$ creates a table Tab with N rows and chooses $r_1 = 1 \| \{0\}^{l-1}$, where $\{0\}^{l-1}$ denotes a string of zeros of length $(l-1)$-bits, and runs N sessions with that tag T_i as follows, for $1 \leqslant j \leqslant N$:

1) $\mathcal{A}$ sends $r_1 = 1 \| \{0\}^{l-1}$ to the tag.
2) After receiving r_1, the tag generates a random number r_2^j. If $r_2^j \neq 0$ then the tag:
   - 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 impersonated 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' with N' rows, chooses $r_1 = 1 \| \{0\}^{l-1}$ and runs N' sessions with the tag T_i' as follows, for $1 \leqslant f \leqslant N'$:

1) $\mathcal{A}$ sends $r_1 = 1 \| \{0\}^{l-1}$ to T_i'.
2) After receiving r_1, the tag T_i' generates a random number r_2^f. If $r_2^f \neq 0$ then the tag:
   - 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 impersonated 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, the adversary $\mathcal{A}$ checks:

- if there exists a pair $(((y^j, t^j) \in Tab), ((y^f, t^f) \in Tab'))$ such that $(t^j|_{0\sim k-1} = t^f|_{0\sim k-1})$ but $(y^j|_{0\sim k} \neq y^f|_{0\sim k})$ then $T_i \neq T_i'$, for $0 \leqslant j \leqslant N$, $0 \leqslant f \leqslant N'$ and $0 \leqslant k \leqslant l-1$; otherwise, $T_i = T_i'$.

The total complexity of the given attack is N sessions, required for the learning phase, plus N' sessions, required for the execution phase. The adversary's advantage Adv_A to make the correct decision in the third phase of the attack is defined 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 do as follows:

1) For any entry t^j in Tab', for $1 \leqslant j \leqslant N'$ and for any $1 \leqslant k \leqslant l$, we denote the number of entries in Tab such that $(t^j|_{0\sim(k-1)} = t^f|_{0\sim(k-1)})$ but $((t^j)_k \neq (t^f)_k)$, for $1 \leqslant f \leqslant N$, by M_k^j.
2) The expected value of M_k^j is $\frac{N}{2^{k+1}}$.
3) Following the given observation, if $t^j|_{0\sim(k)} = t^f|_{0\sim(k)}$ and $T_i = T_i'$ then the adversary can conclude that $(r_2^j)|_{0\sim(k-1)} = (r_2^f)|_{0\sim(k-1)}$. On the other hand, for any

version of the SUAP protocols and for the selected value as r_1, one can state that $y^j = \mathcal{X} + r_2$, where $\mathcal{X}$ depends on the used version of the SUAP protocol but is static for a given tag in a given version of the protocol. So, if $r_2^j|_{0\sim(k)} = r_2^f|_{0\sim(k)}$ the adversary verifies whether $y^j|_{0\sim(k-1)} = y^f|_{0\sim(k-1)}$; which is satisfied for $T_i = T_i'$ with the probability equal to 1 and for $T_i \neq T_i'$ with the probability equal to 2^{-k}.

4) Hence, recall that $M_k^j = \frac{N}{2^{k+1}}$, the probability that $T_i \neq T_i'$ but the adversary outputs "1"(wrong alarm), Pr_{wrong}, is determined as follows:

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

5) The adversary's advantage to trace the target tag successfully is given by:

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

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 $Adv_A \geqslant 1-2^{-14}$. Hence, even for the on-line complexity of only one run of the protocol, the success probability of the given attack is almost equal to 1. For $N = N' = 16$ we have $Adv_A \geqslant 1-(2^{-14})^{16} = 1-2^{-224}$ which is almost equal to 1. An interesting point of this attack is that it works for all three protocols SUAP1, SUAP2 and SUAP3 and even the adversary does not require to know which protocol the target tag is using.

IV. Conclusions

In this paper we have shown that the recently proposed RFID authentication protocols by Morshed *et al.* fail to provide adequate security against traceability attacks. In this paper we presented an attack which can trace an RFID tag whenever it uses any of the protocols proposed by Morshed *et al.*, i.e. SUAP1, SUAP2 and SUAP3. We also show that the success probability of the attack is very high. In general terms SUAP family of protocols overuses the XOR operation, and this weak property should be avoided in cryptography.

V. Acknowledgements

This work was partially supported by the Marie Curie IEF project "PPIDR: Privacy-Preserving Intrusion Detection and Response in Wireless Communications", grant number: 252323.

References

[1] E. Y. Choi and S. M. Lee and D. H. Lee. Efficient RFID authentication protocol for ubiquitous computing environment. In *Embed. Ubiquit. Comput.*, volume 3832 of *Lecture Notes in Computer Science*, pages 945–954. Springer-Verlag, 2005.

[2] 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.

[3] S. M. Lee and Y. J. Hwang, D. H. Lee and J. I. Lim. DEfficient authentication for low-cost RFID systems. In *ICCSA05*, volume 3480 of *Lecture Notes in Computer Science*, pages 619–629, Springer-Verlag, 2005.