REVIEW PAPER *Special Section on Trust, Security and Privacy for Pervasive Applications*

Lightweight Props on the Weak Security of EPC Class-1 Generation-2 Standard

Pedro PERIS-LOPEZ[†], *Member*, **Tieyan LI**[††], *and* **Julio C. HERNANDEZ-CASTRO**[†††], *Nonmembers*

SUMMARY In 2006 EPCglobal and the International Organization for Standards (ISO) ratified the EPC Class-1 Generation-2 (Gen-2) [1] and the ISO 18000-6C standards [2], respectively. These efforts represented major advancements in the direction of universal standardization for low-cost RFID tags. However, a cause for concern is that security issues do not seem to be properly addressed. In this paper, we propose a new lightweight RFID tag-reader mutual authentication scheme for use under the EPCglobal framework. The scheme is based on previous work by Konidala and Kim [3]. We attempt to mitigate the weaknesses observed in the original scheme and, at the same time, consider other possible adversarial threats as well as constraints on low-cost RFID tags requirements.

key words:
RFID, security, EPC Class-1 Gen-2 standard, authentication, lightweight cryptographic primitives

1. Introduction

In this paper, we focus on designing a secure authentication scheme for use under the EPCglobal framework. A number of previous works, such as [3], [4] and [5], have proposed protocols to enhance the security of the EPC Class-1 Gen-2 standard. Unfortunately, due to some weaknesses later exposed, these protocols fall short of meeting the desired security objectives. It looks like enforcing authentication under the EPC Class-1 Gen-2 standard specifications is an almost impossible task, and any proposed scheme to day that was based on it has been doomed and failed. In fact, the proposed infrastructure seems to be too weak to be able to support any real security. However we contend that, by making some small enhancements and without the need to revamp the entire set of specifications, it is still possible to reach a reasonable trade-off in designing a reasonably secure authentication scheme for low-cost RFID tags. Readers should note that the proposed scheme is not fully secure as tags' passwords conforming to Gen-2 specification are only 32-bits long. However, the proposed scheme can be considered practically secure for the vast majority of intended applications, such as baggage and tire tracking. Basically, it is a trade off between security and tag's price. The usage of standard cryptography could increment security but hardware requirements (circuit area, memory and power consumption) would be much larger, resulting in a higher cost that basically rules out these kind of cryptographically stronger solutions for low-cost transponders.

A tag-reader mutual authentication scheme that uses a specially designed $MixBits$ function is presented in this work. The underlying protocol is similar to that proposed by Konidala and Kim in [3] (we shall refer in the following to their scheme as the tag-reader mutual authentication or TRMA scheme), with its observed weaknesses addressed by introducing the $MixBits$ function. Under some rigorous analysis, we show that $MixBits$ increases the security of the scheme by providing stronger resistance against common attacks. Furthermore, we show that $MixBits$ requires only a small amount of circuit area, memory size, and power consumption and can be feasibly implemented even on low-cost RFID tags.

2. Background

An RFID system is composed of three main components. Readers (transceivers) interrogate tags (transponders) to access the information stored in their memory. Afterwards, they pass this acquired information to a back-end database which employs it as a search index to allocate all the information associated with the target tag. Readers and tags use the radio channel for communication, which is commonly assumed to be insecure. On the other hand, as readers and the back-end database are computationally much more powerful than tags, a secure channel is commonly assumed.

RFID technology may be envisioned as the substitute of barcodes. However, the massive adoption of this technology is being delayed due its associated security threats. A number of research papers addressing RFID security and privacy problems have been published in the existing literature (we refer the reader to [6], [7] for a detailed survey on this topic).

Our interest here is on RFID tag-reader mutual authentication protocols. Among the numerous authentication protocols that have been proposed, a number of them rely on the usage of classic cryptographic prim-

[†] Delft University of Technology (TU-Delft), Information and Communication Theory group (ICT), P.O. Box 5031 2600 GA, Delft, The Netherlands

[††] Institute for Infocomm Research, A*STAR Singapore

[†††] School of Computing Science, Buckingham Building, Lion Terrace, Portsmouth PO1 3HE, United Kingdom

DOI: 10.1587/transinf.E0.D.1

itives, such as symmetric block/stream ciphers, hash functions, etc. In this paper, we focus on designing a lightweight authentication protocol for use under the EPCglobal Framework. Our objective is to feature a small gate count, memory size and power consumption for the implementation of security functions on a low-cost RFID tag. Such a restriction implies that we would have to do without classic cryptographic primitives.

Based on the EPC Class-1 Gen-2 standard specifications [1], an RFID tag that is compliant with the standard must implement the following: 1) A 32-bit secret access password that is used to control access to the tag's memory; 2) A 32-bit secret kill password that is used to kill (i.e. permanently disable) the tag; 3) a 16-bit pseudo-random number generator; 4) a 16-bit cyclic redundancy checksum.

Furthermore, a simple cover-codding technique is used to protect Access and Kill password. Specifically, the tag and the reader exchange three messages: 1) The reader sends a request message to the tag; 2) The tag backscatters a random number; 3) The received value is xored with the Access/Kill password and sends the result to the tag. 4) Finally, the tag checks the correctness of the password. The security of the above procedure is extremely weak, so performing a passive attack is very simple. An attacker listening in to the backward (tag-to-reader) and forward (reader-to-tag) channel (a very realistic assumption when using the air channel) can pick up the random number sent by the tag. Then the attacker can decrypt the ciphertext sent by the reader by performing an XOR operation (addition modulo 2) with the previous eavesdropped random number. So the Access/Kill password can be obtained by this quite simple mechanism. A detailed analysis of Gen-2 specification can be found in [8].

3. Related Work

We summarize the most important proposals that attempt to rectify the security flaws presented in Gen-2 standard whilst are as best in conformance to this specification.

In [10], Juels *et al.* examined various ways for RFID tags to perform cryptographic functions while remaining EPC-C1G2 compliant. Their main idea is to take an expansive view of EPC tag memory. Instead of considering memory merely as a storage medium, they use it as an input/output way of interfacing with a cryptographic module within the tag. Read/write commands may therefore involve cryptographic values, such as messages in a challenge-response protocol. Their work clearly shows the need for mutual authentication between readers and tags. However, the assumption that a low-cost tag might support on-board cryptographic modules is unrealistic, at least at present time.

Karthikeyan and Nesterenko [11] proposed an efficient tag identification and reader authentication protocol based on a simple XOR and matrix operations. Two matrices and a key are stored in both the tag (K, M_1, M_2^{-1}) and the back-end database (K, M_1^{-1}, M_2). Once the tag is identified, the reader sends to the tag messages Y, Z. The first is used to authenticate the tag and the second to update the key. However, an attacker can substitute the original Z by a random Z'. Upon receiving Y, Z', the tag will be authenticated and will wrongly update the key. So the legitimate reader and the tag will not be able to authenticate each other any more. Additionally, the protocol is vulnerable to replay attacks, and privacy location is not guaranteed [5].

Duc *et al.* [4] proposed a tag-to-back-end database authentication protocol. The security of Duc *et al.*'s protocol is based on key synchronization between tags and the back-end database. The last message of the protocol consists of an *EndSession* command, which is sent to both tags and readers. Interception of one of these messages will cause a synchronization loss between the tag and the server. The tag and the reader will then be no longer able to authenticate, which is an extremely serious problem. The protocol also presents backward secrecy problems, as compromise of the Electronic Product Code (EPC) allows an attacker to trace back all previous communications.

Chien*et al.* pointed out certain weaknesses in the schemes [11] and [4], and then proposed a new EPC-C1G2 compliant mutual authentication protocol [5]. However, Peris *et al.* [12] showed how none of the objectives are met, as it is vulnerable to attacks including identity impersonation, non-forward security and tracking. Execution of the protocol itself even produces de-synchronization between the tags and the back-end database.

In [3], Konidola and Kim produced an interesting paper which tried to correct some of the security shortcomings of the EPC-C1G2 specification. The authors hold that the proposed tag-reader mutual authentication scheme (TRMA scheme in short) frustrates the access password acquisition using a simple XOR operation, in contrast to what happens in the specification. However, Lim and Li [9] showed how a passive attacker can recover the tag's password by eavesdropping over a single run of the protocol and performing correlation analysis on the captured information. Konidala and Kim then proposed a new version of the TRMA scheme (TRMA$^+$) in which the tag access and kill passwords are used for authentication. The extended TRMA scheme offers greater resistance against Lim and Li's attacks. It is much more difficult for an adversary to recover the access password under the correlation attack, or to forge a successful authentication under the dictionary attack. However, Peris-Lopez *et al.* showed how the access password can be disclosed under the assumption of an active attacker [16]. In addition, Lim *et al.* exposed that the correlation attack

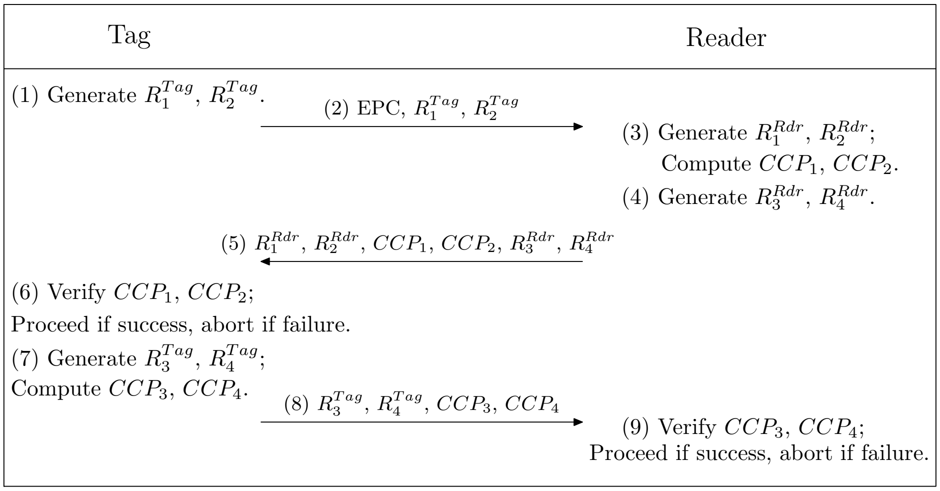
Tag
Reader
(1) Generate R_1^{Tag}, R_2^{Tag}.
(2) EPC, R_1^{Tag}, R_2^{Tag}
(3) Generate R_1^{Rdr}, R_2^{Rdr};
Compute CCP_1, CCP_2.
(4) Generate R_3^{Rdr}, R_4^{Rdr}.
(5) R_1^{Rdr}, R_2^{Rdr}, CCP_1, CCP_2, R_3^{Rdr}, R_4^{Rdr}
(6) Verify CCP_1, CCP_2;
Proceed if success, abort if failure.
(7) Generate R_3^{Tag}, R_4^{Tag};
Compute CCP_3, CCP_4.
(8) R_3^{Tag}, R_4^{Tag}, CCP_3, CCP_4
(9) Verify CCP_3, CCP_4;
Proceed if success, abort if failure.

Fig. 1 The TRMA authentication protocol.

can be used against this scheme in a different way – to recover the kill password after eavesdropping over multiple authentication sessions [17].

Other recent proposals have resulted a vain attempt as shown in [13] and [14].

4. The TRMA Schemes

In this section, we briefly describe the original TRMA scheme, as well as the extended TRMA scheme.

4.1 The Original TRMA Scheme

In [3], Konidala and Kim presented a lightweight tag-reader mutual authentication (TRMA) scheme that uses some of the features in a EPC Class-1 Gen-2 tag, as well as a specially designed pad generation function $PadGen$.

The $PadGen$ function is used to produce a cover-coding pad to mask the tag's access password before transmission. The function is performed on the tag's 32-bit access password PWD, which is broken up into 2 parts – PWD_M (comprising the 16 most significant bits) and PWD_L (comprising the 16 least significant bits). $PadGen$ takes two 16-bit random numbers R_i^{Tag} (generated by the tag) and R_i^{Rdr} (generated by the reader) as its inputs. Using each of the four hexadecimal digits in R_i^{Tag} (or R_i^{Rdr}) to indicate a bit address within PWD_M or PWD_L, $PadGen$ then selects those bits from PWD_M and PWD_L to form the 16-bit output pad (see the original paper for details [3]).

Under the scheme, each cover-coding pad PAD_i (for $i = 1, 2, 3, 4$) can be expressed as

$$PAD_i = PadGen(PWD,\ R_i^{Tag},\ R_i^{Rdr}) \tag{1}$$

and the authentication responses (otherwise known as the cover-coded passwords in [3] and [15]) can be expressed as

$$CCP_{\{1,3\}} = PWD_M \oplus PAD_{\{1,3\}} \tag{2}$$

$$CCP_{\{2,4\}} = PWD_L \oplus PAD_{\{2,4\}} \tag{3}$$

Fig. 1 depicts a single run of the authentication protocol.

4.2 The Extended TRMA Scheme

In [15], Konidala, Kim and Kim presented an extended version of TRMA, which uses both the 32-bit access password and the 32-bit kill password. The TRMA$^+$ scheme uses the same message exchange but consumes two rounds of $PadGen$ (instead of a single round in the original TRMA scheme), one nested within the other, to compute each cover-coding pad. The inner round performs $PadGen$ over the access password, while the outer round performs $PadGen$ over the kill password. Instead of (1), the resulting pad would then be expressed as

$$PAD_i = PadGen(KWD,\ PadGen(PWD, R_i^{Tag}, R_i^{Rdr}),\ R_i^{Tag}) \tag{4}$$

where KWD denotes the kill password.

5. M^3 Authentication Protocol (M^{3}AP)

In this section, we introduce a new lightweight authentication scheme, known as M^{3}AP, to strengthen the security of the EPC Class-1 Gen-2 standard. We design M^{3}AP by extending Konidala and Kim's scheme [3], and make use of a $MixBits$ function to mitigate the security weaknesses found in the original scheme.

5.1 Objectives

With M^{3}AP, we emphasize that the main objective is to design a simple, cost-effective, lightweight and practical authentication protocol that provides mutual authentication between an RFID tag and an RFID reader under the EPCglobal framework. Privacy is not a main objective like in Gen-2 specification. Hence, as in the authentication protocol specified under the Gen-2 standard and the previously proposed TRMA protocols, we do not make provisions to enforce privacy by protecting the unique EPC but instead, allow the EPC of RFID tags to be transmitted in clear. Inevitably, this poses a problem to application environments whereby the privacy of tags and/or tag users is essential. In such cases, it would be pertinent to include measures for privacy protection. While our current scheme does not enforce privacy, we contend that it would be possible to extend the scheme to provide the necessary protection although this would require additional considerations.

5.2 The Protocol

In this section, we present an improved version of Konidala and Kim's TRMA scheme that seeks to mitigate its security weaknesses. The proposed protocol was designed by taking into account tag restrictions (computational, storage and circuitry) and with minimal modifications to the general framework of the Gen-2 specification.

Assumptions: Each tag has two passwords (i.e. access -PWD- and kill -KWD- password), and is able to generate 16-bit random values and to compute checksum values. According to the set of operations supported on-chip, transponders make use of bitwise XOR operation, *PadGen* function defined in [3], and the specific-defined *MixBits* function.

We assume that the tag is singulated using a probabilistic (e.g. Aloha-based protocol) or deterministic (e.g. binary tree-walking protocol) collision avoidance protocol. At the end of each singulation, a tag is selected to communicate with the reader.

The protocol is described as follows:

(1) $Tag \rightarrow Reader : EPC,\ R_1^{Tag},\ R_2^{Tag},\ R_3^{Tag},\ R_4^{Tag}$

The tag backscatters its EPC number. Then, the reader sends the command *Req_RN* to the tag over four times. Each time, the tag backscatters a new random number (R_i^{Tag} for $i = \{1, 2, 3, 4\}$) and stores it into its memory. These are used as random challenges to the reader. Upon receiving the EPC, the reader uses it to perform an index search to retrieve the access password PWD associated with the tag from the backend database. Once PWD is obtained, the reader will then go on to compute the authentication responses.

(2) $Reader \rightarrow Tag : CCP_1,\ CCP_2,\ R_1^{Rdr},\ R_2^{Rdr},\ R_3^{Rdr},\ R_4^{Rdr}$

The reader transmits its computed responses, as well as a set of random numbers as authentication challenges to the tag. To obtain the responses CCP_1 and CCP_2, it first computes an intermediate 32-bit vector PWD' from PWD and the received random challenges $\{R_i^{Tag}\}_{i=1}^{4}$ by using our proposed $MixBits$ function:

$$\begin{aligned} PWD' &= MixBits(PWD \oplus (R_1^{Tag} \mid\mid R_2^{Tag}), \\ &\quad R_3^{Tag} \mid\mid R_4^{Tag}) \end{aligned} \tag{5}$$

The reader then computes the authentication responses CCP_1 and CCP_2 as follows:

$$\begin{aligned} CCP_1 &= PWD_M \oplus PadGen(PWD', \\ &\quad R_1^{Tag} \oplus PWD'_L,\ R_1^{Tag} \oplus PWD'_M) \\ CCP_2 &= PWD_L \oplus PadGen(PWD', \\ &\quad R_2^{Tag} \oplus PWD'_L,\ R_2^{Tag} \oplus PWD'_M) \end{aligned}$$

Instead of applying *PadGen* to the static access password, we apply *PadGen* on a vector computed from the access password. This vector changes as the random challenges vary. Furthermore, both CCP_1 and CCP_2 depend only on the random challenges generated by the tag. In the original and extended TRMA schemes, CCP_1 and CCP_2 would partially depend on pseudo-random numbers generated by the reader, which presents an avenue for a malicious reader to exploit and reduces the reliability of the responses computed. After computing the responses, the reader then generates four new random numbers (R_i^{Rdr} for $i = \{1, 2, 3, 4\}$) and presents them as challenges to the tag. The reader also stores the random challenges, which will be used to verify the tag responses.

(3) $Tag :$ Verify CCP_1 and CCP_2 .

The tag receives CCP_1 and CCP_2. The access password and the random numbers used in the computation of CCP_1 and CCP_2 are already stored in its memory. Therefore, it has the necessary information to compute CCP'_1 and CCP'_2 in the same way that the reader computed CCP_1 and CCP_2. The tag then compares these values with the values sent by the reader:

1. If $CCP_1 = CCP'_1$ and $CCP_2 = CCP'_2$, then verification is successful. The tag considers the reader to be an authorized entity.
2. Otherwise, verification fails. The tag ends its communication with the reader and returns to arbitrate state.

(4) $Tag \rightarrow Reader : CCP_3,\ CCP_4$

To authenticate itself, the tag needs to reply to the reader with CCP_3 and CCP_4, which are computed by taking steps similar to those taken by the reader. The tag first computes an intermediate 32-bit vector from its access password:

$$PWD'' = MixBits(PWD' \oplus (R_1^{Rdr} \mid\mid R_2^{Rdr}),$$

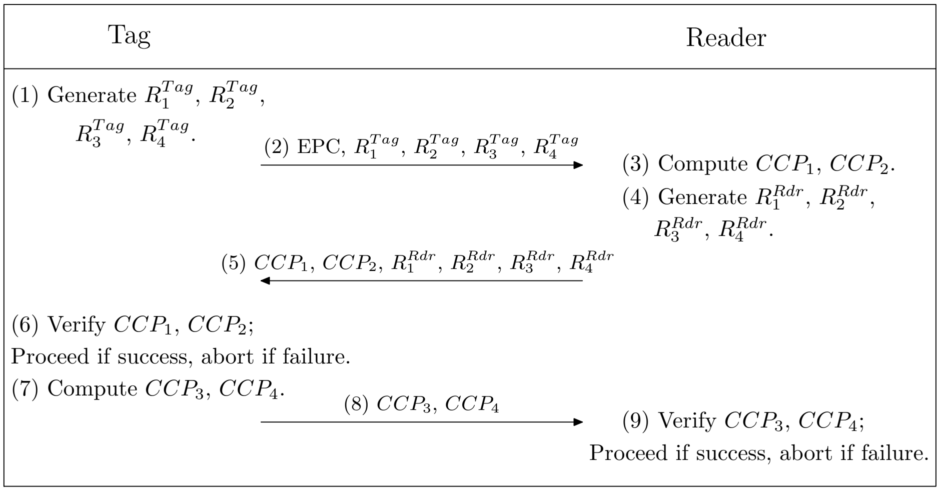
Tag
Reader
(1) Generate R_1^{Tag}, R_2^{Tag}, R_3^{Tag}, R_4^{Tag}.
(2) EPC, R_1^{Tag}, R_2^{Tag}, R_3^{Tag}, R_4^{Tag}
(3) Compute CCP_1, CCP_2.
(4) Generate R_1^{Rdr}, R_2^{Rdr}, R_3^{Rdr}, R_4^{Rdr}.
(5) CCP_1, CCP_2, R_1^{Rdr}, R_2^{Rdr}, R_3^{Rdr}, R_4^{Rdr}
(6) Verify CCP_1, CCP_2;
Proceed if success, abort if failure.
(7) Compute CCP_3, CCP_4.
(8) CCP_3, CCP_4
(9) Verify CCP_3, CCP_4;
Proceed if success, abort if failure.

Fig. 2 The M^{3}AP authentication protocol.

$$R_3^{Rdr} \mid\mid R_4^{Rdr}) \qquad (6)$$

Thereafter, the tag computes CCP_3 and CCP_4 as follows (note that $PadGen$ is now computed over the new intermediate vector PWD''):

$$\begin{aligned} CCP_3 &= PWD_M \oplus PadGen(PWD'', \\ &\quad R_1^{Rdr} \oplus PWD''_L,\ R_1^{Rdr} \oplus PWD''_M) \\ CCP_4 &= PWD_L \oplus PadGen(PWD'', \\ &\quad R_2^{Rdr} \oplus PWD''_L,\ R_2^{Rdr} \oplus PWD''_M) \end{aligned}$$

The tag then sends the two authentication responses (CCP_3, CCP_4) to the reader.

(5) *Reader* : Verify CCP_3 and CCP_4 .

The reader receives the responses CCP_3 and CCP_4 from the tag, computes CCP'_3 and CCP'_4 based on the information known to it, and then compares the received values with the computed values:

1. If $CCP_3 = CCP'_3$ and $CCP_4 = CCP'_4$, then verification is successful. The reader considers the tag as an authentic (or genuine) tag.
2. Otherwise, verification fails. The reader will emit an alarm to the back-end database to indicate this event (perhaps to inform the database that a fake tag or a counterfeit product is detected).

Fig. 2 depicts a single run of the authentication protocol.

6. The *MixBits* Function

RFID tags (e.g. EPCglobal Class I or Class II tags) are devices with severe limitations (in terms of computational, storage and circuitry requirements). Due to these hard restrictions, the use of standard cryptographic primitives lies beyond their capabilities. Hence, the design of a secure lightweight $MixBits$ function for our M^{3}AP protocol becomes a stimulating challenge. We contend that the basic requirements of this function should be as follows:

1. Only efficient operations that are easily implemented in hardware should be used. For example, rotations may be included, but multiplications should be excluded due to their high cost [18].
2. Triangular functions [20] (e.g. bitwise operations or addition $\bmod\ N$) and non-triangular functions (e.g. rotations) should be combined to prune simple linear cryptanalysis [21].
3. A highly non-linear function that provides a negligible relationship between the inputs and the outputs, should be used.
4. Temporal requirements will limit the number of operations a tag may compute. The read speed of a tag conforming to Gen-2 is about twice that of Gen-1, with average read rates of around 450 tags per second [1]. Additionally, power consumption also restricts the total account of operations consumed.

We obtain possible candidates for $MixBits$ through evolving compositions of extremely light operands by means of genetic programming. We refer the interested reader to [22] where a detailed description of the methodology used to obtain our function is explained. Several experiments are then conducted on the candidates to pick the best highly non-linear function. At the end of the experiments, the following function is selected for $MixBits$, which conforms to all the aforementioned requirements. Inputs (X and Y) and output (Z) are 32-bits of length and the function has a loop of 32 iterations:

```
Z = MixBits(X,Y)
----------------------------
Z = X;
for(i=0; i<32; i++) {
        Z = (Z<<1) + ((Z+Y)>>1);}
     ----------------------------
```

where addition is carried out modulo 2^{32}, $<<$ denotes bitwise left shift and $>>$ denotes bitwise right shift.

7. Analysis of the *MixBits* Function

In this section, we cryptanalyze the proposed function and conduct an statistical analysis over is output. Then, we estimate the hardware requirements to its implementation and analyze the throughput provided by this function. Finally, *MixBits* is compared with several lightweight primitives.

7.1 Security Analysis

Linear cryptanalysis, commonly used for block cipher cryptanalysis, is employed to study how the output of this function can be approximated by a linear function. In order to obtain a linear bias, the following experiment is carried out: two 32-bit masks (A, B) are randomly picked, and two consecutive outputs (Z_i, Z_{i+1}) are generated from them. With these two masks, the equality $A * Z_i = B * Z_{i+1}$ is evaluated ($*$ denotes scalar product, with a mod 2 operation carried out after the addition). This process is repeated 2^n times, from which we obtain the number of successes (m). The bias is then defined as:

$$BIAS = \frac{1}{2^{-log_2(|\frac{m}{2^n} - \frac{1}{2}|)}} \tag{7}$$

Several pairs of different masks A and B, are randomly picked and tested. For each mask pair, 2^{25} 32-bit outputs are generated, and the expression $A*Z_i = B*Z_{i+1}$ is evaluated over them. To obtain these outputs, the X and Y variables are initialized to random values in the beginning and as the experiment runs, the X variable remains unchanged (here, we attempt to consider a disadvantageous scenario) while the Y variable is set to a new random value each time a new output is computed. From the above experiment we can deduce that the bias of the *MixBits* function is bounded by $2^{-11.13}$.

The serial correlation coefficients (at bit, byte and 4-byte level) are also studied to measure the extent to which a new intermediate output Z_i depends upon the previous value Z_{i-1}. To obtain a sizeable test sample, 2^{24} Z outputs are computed. As in the above experiment, the X and Y variables are randomly initialized at the beginning, and the Y variable is set to a new random value each time a new output is computed. Further analysis on the XOR ($Z_i \oplus Z_{i+1}$) and the difference ($Z_i - Z_{i+1}$) between two consecutive outputs are performed. The results are summarized in Table 1.

In addition, we evaluate how an attacker might predict an output if previous outputs are known. The bit-byte prediction tests [23] used to evaluate the randomness of the Konton2 stream cipher are employed for this purpose. Eight algorithms are used to predict the value of each bit (resp. byte) from the beginning to the end of the sequence. For a perfectly random sequence, the probability of success of any of the algorithms should be 1/2 (resp. $1/2^8$). The number of successes is counted, and a chi-squared statistic with 1 degree of freedom computed. Table 2 shows the results.

Table 1 Serial Correlation Test

	$Z = MixBits(X,Y)$		
Experiment	**Bit**	**Byte**	**4-Byte**
Z_i	0.000024	-0.000089	0.000279
$Z_i \oplus Z_{i+1}$	-0.000045	0.000026	0.000174
$Z_i - Z_{i+1}$	-0.000059	-0.000009	-0.000196

Table 2 Bit-Byte Prediction Tests for Randomness (Adapted from [23])

	$Z = MixBits(X,Y)$		
Test	Z_i	$Z_i \oplus Z_{i+1}$	$Z_i - Z_{i+1}$
Bit Prediction Test A	0.0032	0.8446	0.8453
Bit Prediction Test B	0.1284	0.7925	0.7928
Bit Prediction Test C	0.4094	0.9735	0.9729
Bit Prediction Test D	0.7448	0.9690	0.9687
Bit Prediction Test E	0.2975	0.6758	0.6717
Byte Prediction Test A	0.3049	0.6919	0.6970
Byte Prediction Test B	0.8854	0.8551	0.8522
Byte Prediction Test C	0.8549	0.8246	0.8209
Byte Prediction Test D	0.1717	0.9493	0.9483
Byte Repetition Test	0.7289	0.0684	0.0685

From our analysis, we find that *MixBits* has very good properties. Indeed, our analysis shows that the output of *MixBits* cannot be predicted significantly better than a pure random guess if the adversary does not have any knowledge of the secret access password. That is, output sequence of *MixBits* is not polynomial distinguishable from a truly random number.

7.2 Performance Analysis

The *MitBits* function only uses efficient operations and combines triangular (i.e. addition mod 2^{32}) and non-triangular functions (i.e. bitwise right/left shift) as requirements 1 and 2 demand. Specifically, the necessary architecture to implement this function can be divided into three main modules: 1) Memory blocks where all the 32-bit used variables are stored; 2) Arithmetic logic Unit in which the addition mod 2^{32} is supported; 3) Additional logic used for control purpose. An estimate of the gate count for *MixBits* can be easily obtained. Six logic gates are needed for each bit added in parallel[†]. The registers will be implemented by means of flip-flops, each of which requires 8 gate equivalents [24]. Specifically, two 32-bit registers are needed – one to store the output Z and another for the intermediate results. Hence, a total of around 740 gate equivalents are needed to implement *MixBits*, considering that a

[†] $S = A \oplus [B \oplus C_{ENT}]$ $C_{SAL} = BC_{ENT} + AC_{ENT} + AB$

Table 3 Performance Comparison

Cryptographic primitive	Gates Equivalent	Cycles per block	Throughput at 100 KHz (Kbps)	Price (Cents)
Mixbits	740	128	25	K
Block ciphers				
Present [28]	1,570	32	200	K + 0.83
DESL [29]	1,848	144	44.4	K + 1.12
HIGHT [30]	3,048	34	188.2	K + 2.31
AES [31]	3,400	1,032	12.4	K + 2.66
Stream ciphers				
Grain-80 [32]	1,294	1	100	K + 0.51
Grain-80, x16 [32]	3,239	1	1,600	K + 2.50
Trivium [32]	2,599	1	100	K + 1.86
Trivium, x16 [32]	3,185	1	1,600	K + 2.44
Hash functions				
MD5 [33]	8,400	612	20.91	K + 7.66
SHA-1 [34]	8,120	1,274	12.56	K + 7.38
SHA-256 [34]	10,868	1,128	22.69	K + 10.13

0.05 * (Memory block + Arithmetic logic Unit) is the circuit area taken up by the control unit.

An estimate of the temporal requirements can also be carried out. A tag has to spend around 128 clock cycles to compute an output ($Z = MixBits(X, Y)$). Assuming a clock frequency of 100 KHz , which is the most common operation frequency for RF transponders [25], a tag can compute around 780 updates per second. Hence, the timing requirements ($>$ 450 answers/sec) are completely fulfilled.

To complete the analysis, a comparison with several block/stream ciphers and hash functions is carried out and the results shown in Table 3 (for the price comparison, *MixBits* is fixed as the reference and every extra 1,000 gates is assumed to increase chip price by $0.01 [26]). We find that *MixBits* is the most efficient in circuit area and although throughput is not the highest, it is within the requirements of the intended applications (e.g. baggage tracking, electronic toll collection, pallet tracking, etc.).

Lightweight ciphers such as Present or Grain require only 1,570 or 1,294 gate equivalents respectively. However, this number of gates, even though small, may still exceed the capabilities of tags conforming to Gen-2 specification. Furthermore, where tag price is concerned, slight differences in tag prices can be greatly magnified under an operating environment where large numbers of tags are deployed. Imagine for a company that needs to deploy 500 million tags. A difference of US $0.0051 – 0.0083 (Grain-Present) per tag would amount to US $ 2,550,000 – 4,150,000 of extra costs in total, which is a significant sum. In this case, using Grain or Present cipher could be rather expensive. In addition, the above primitives can not take up the whole of the circuit area devoted to security because a significant number of logic gates has to be set aside for the 16-bit PRNG supported on-chip and a PRNG conforming to Gen-2 specification requires around 1,600 gate equivalents [27]. While the use of a cipher or hash function would increase the level of security, it would also incur hardware costs. In this work, our main objective is to design a lightweight authentication protocol under the EPCglobal Framework, and requires balancing tradeoffs between security and hardware restrictions. Summarizing, we find that our proposed *MixBits* function performs reasonably well and provides an appropriate security level for tags compliant with the EPC Class-1 Gen-2 specification.

8. Analysis of the M^{3}AP Protocol

In this section, we provide a proof sketch to show that our proposed M^{3}AP protocol provides mutual authentication between a tag and a reader. In addition, we also analyze the security of the protocol by examining how the protocol fares against previous attacks exposed on the TRMA schemes, as well as other passive and active attacks.

8.1 Verification of Mutual Authentication.

In an RFID context, authentication serves the purpose of validating and confirming the identities of tags and/or readers. With proper authentication, illegitimate readers and tags can then be identified from legitimate ones and the necessary action can then be taken thereafter. Our proposed M^{3}AP protocol have been designed to incorporate both reader-to-tag authentication (with CCP_1 and CCP_2) and tag-to-reader authentication (with CCP_3 and CCP_4). In this section, we show that M^{3}AP achieves these objectives.

Reader-to-Tag Authentication: The first two messages of our proposed scheme allow a legitimate reader that has knowledge of the tag's access password to authenticate itself to the tag. A malicious (or illegitimate) reader does not possess the access password to generate the corresponding responses (CCP_1 and CCP_2). Due to lack of authorization for the illegitimate reader, this information cannot be obtained from the manufacturer (EPC-IS). In addition, the computation of CCP_1 and CCP_2 uses only random challenges from the tag. In the original TRMA scheme, CCP_1 and CCP_2 are computed from random values generated by the tag, as well as random values generated by the reader. However, this provides an avenue for an illegitimate reader to specify the random values in such a way that allows it to circumvent the scheme and forge a successful authentication more easily. By having the reader compute the authentication responses based solely on random challenges generated by the tag and the shared secret (the tag access password), our scheme eliminates such a weakness.

Tag-to-Reader Authentication: The third message of our scheme is for a legitimate tag to authenticate itself to the reader after it has confirmed that the reader is a legitimate one. A fraudulent tag does not possess the access password that is necessary to compute the cover-codes (CCP_3 and CCP_4). In this case, cover-codes only depend on the random numbers picked by the genuine reader and avoids the vulnerability in the original TRMA scheme whereby the attacker has control over the inputs required to compute the authentication response. Hence, without knowledge of the correct access password, a tag impersonation attack cannot be successful and authentication would fail.

The following Lemma highlights the fact that the security level for this scheme is bounded by 2^{-32}. Note that access and kill passwords are 32-bits conforming to Gen-2 specification. Indeed, the proposed scheme represents a trade of between security and keeping the scheme under the EPCglobal framework, instead of using a more demanding resources and completely secure solution (e.g. hash-based RFID protocol).

Lemma. The security of the M^3AP *protocol against an adversary* A *making at most* q_{rt} *or* q_{tr} *queries is bounded by* $2^{-2\cdot l}$*, where* l *denotes the length of the cover-codes:*

$$Adv_A^{M^3AP} = max\{q_{rt} \cdot Succ_{CCP_1\&CCP_2}(l), \quad q_{tr} \cdot Succ_{CCP_3\&CCP_4}(l)\} \tag{8}$$

where $Succ_{CCP_i\&CCP_j}(l) = \frac{1}{2^l} \cdot \frac{1}{2^l} = \frac{1}{2^{2\cdot l}} = \frac{1}{2^{2\cdot 16}} = \frac{1}{2^{32}}$ and q_{rt}/q_{tr} symbolizes the number of queries sent by an adversary to the tag/reader respectively.

8.2 Resistance against Previous Attacks on TRMA

Resistance against the Correlation Attack. In order to perform the correlation attack described in [9], the adversary first needs to find a correlation between the access password PWD and the $MixBits$ output (PWD' and PWD''). Once this is found, the adversary can make use of the relationships derived in [9] (the relationships between PWD and the output of $PadGen$, where in our new scheme, $PadGen$ is applied to PWD' and PWD'' instead of PWD) to attack the scheme. However, as witnessed in the last section, we have shown that it is highly difficult to obtain any correlation between the input and output of $MixBits$. Hence, we contend that our proposed scheme provides strong resistance against the correlation attack.

Resistance against the Dictionary Attack. In the original TRMA scheme, the value of each bit of the authentication response CCP_i ($i = \{1, 2, 3, 4\}$) is only dependent on the value of a particular hex-digit in R_i^{Tag} or R_i^{Rdr}. For example, the first bit of CCP_1 depends on the first hex-digit of R_1^{Tag}. If the value of a hex-digit in any R_i^{Tag} or R_i^{Rdr} is repeated (i.e. it had the same value in a previous authentication session), then the adversary would be able to successfully predict the value for the corresponding bit in R_i to forge a successful authentication. In our proposed scheme, we find that each bit in any R_i is dependent on all four 16-bit random numbers generated by the tag and the reader. For example, each bit in CCP_1 or CCP_2 is dependent on all four of R_1^{Tag}, R_2^{Tag}, R_1^{Rdr} and R_2^{Rdr}. This is because all of them are involved in the computation of PWD' in $MixBits$. Moreover, the nature of $MixBits$ ensures that the bits of the four random numbers are diffused within PWD'. In order to successfully predict the value of a bit in CCP_1 or CCP_2, the adversary must encounter a situation whereby all four random numbers contain the same values that have appeared together in a previous session. The probability of this occurring is extremely low, since with a total of 64 bits between them, the number of possible combinations amounts to 2^{64}. Hence, the dictionary attack is still possible but becomes extremely difficult. In fact, this attack can be completely prevented if we update or refresh the access password after every authentication session. For example, we can change the access password from PWD to PWD'' at the end of the protocol after both parties are mutually authenticated. The new access password will then be used for the next authentication session, and so on.

Resistance against the Tag Killing Attack. Unlike the extended TRMA scheme, the kill password is not used in our proposed authentication scheme. Furthermore, all messages exchanged during the protocol are independent of the kill password of the communicating tag. Hence, an adversary would not be able to gather any information about the kill password of tags from authentication sessions under our proposed scheme.

8.3 Resistance against Other Attacks

Resistance against Replay Attacks. In a replay attack, the adversary eavesdrops on the messages exchanged between a legitimate reader and a legitimate tag, and replays the authentication responses to masquerade as the reader or the tag. Such an attack would be successful only if all the four random challenges have appeared together and in the right sequence in a previous authentication session. With the legitimate parties generating fresh random challenges for each authentication session, the probability of success for a replay attack would be extremely low ($1/2^{64}$).

Resistance against Offline Brute Force Attacks. In an offline brute force attack, an adversary eavesdrops on a single pass (for example, the reader-to-tag authentication) of an authentication session to obtain a set of random challenges and the valid response based on those challenges. Next, the adversary assumes a value for the access password and computes a response based on the collected challenges (by executing the $MixBits$ function, the $PadGen$ function, and other necessary operations). If the computed response matches with the collected response, then the value assumed for the access password was correct. Otherwise, the adversary tries the next probable value for the access password, repeating until a correct match is found. The complexity of this attack is $O(2^l)$, where l is the number of bits in the access password. To offer adequate resistance against such an attack, l should be sufficiently large, yet not so large as to violate constraints in the tag implementation. We note that under the Gen-2 standard, $l = 32$ bits and this may not be sufficient to thwart an attack where computations are carried out on a PC.

Resistance against Active Brute Force Attacks. Active brute force attacks generally require an adversary to actively take part in the authentication protocol by masquerading as a tag or a reader. A number of scenarios are possible. In the first scenario, an adversary can programme a malicious reader/tag to repeatedly probe a legitimate tag/reader. During each authentication attempt, the reader/tag tries a different value for the access password. This continues until the adversary authenticates successfully to the tag/reader. In another scenario, an adversary can iteratively issue challenges to the legitimate reader and record valid sets of challenges and responses to form a dictionary. Both attacks can be made infeasible with sufficiently large access passwords and random challenges, or the use of password updating.

Resistance against De-synchronization Attacks. Under our proposed scheme, since the access password is constant, there is no threat of de-synchronization. However, as discussed earlier, to completely prevent some of the attacks, it would be necessary to update the access password at the end of each successful mutual authentication. In this case, the copies of the access password kept at the tag and the reader (or the back-end database, as in most cases) must be the same at all times, i.e. they must be synchronized. Once any party fails to update its copy of the access password at the end of a successful authentication session, both parties will be de-synchronized. Hence, with password updating, extra measures may need to be taken to ensure that the protocol is robust against de-synchronization.

Resistance against Skimming Attacks. In a skimming attack, an adversary reads the contents of a legitimate tag and copies those contents over to the target tag (the clone). For an RFID tag that complies with the EPC Class-1 Gen-2 standard, the access password of the tag would be stored in protected memory and cannot be revealed through skimming. The only contents that could be copied to the clone are the EPC and the data stored in the unprotected memory. These information would not be sufficient for the cloned tag to perform a successful authentication. Hence, our protocol is resistant against unauthorized skimming.

Resistance against Reverse Engineering. For reverse engineering, we assume that the adversary is equipped with the expertise to disclose contents stored in both the unprotected and protected memory of the tag. Possible methods used could include probing, fault injection, power and timing analysis, etc. Obviously, the low-cost EPC Class-1 Gen-2 tags would be vulnerable to a highly skilled adversary who is well-versed in reverse engineering techniques. With all the tag contents disclosed, the adversary can forge a successful authentication with a legitimate reader. However, since the access password for each tag is expected to be different, the adversary would only be able to masquerade as the compromised tag and not other tags. This attack could be rather expensive to launch on a large scale tag constellation and may not be cost-effective since the benefits reaped to the adversary are highly likely to outweigh the costs incurred.

Resistance against Unauthorized Tracking. As mentioned earlier, privacy is not a focus in this work and the current EPCglobal Framework does not seem to address privacy issues. The transmission of the EPC in clear implies that unauthorized tracking of tags is possible. We contend that it is possible to integrate previously proposed methods with our scheme to guard against privacy violation. For example, the EPC can be replaced with a pseudonym (as proposed in [35]) or be relabelled (as in [36]) to prevent tracking of the tag. The EPC can also be protected using masking or RF jamming techniques (e.g. [37]), or through controls provided by an RFID proxy device (e.g. [38]). Naturally, implementing these solutions for privacy protection incurs in higher costs on the resulting system.

9. Conclusions

In this paper, a new authentication protocol named M^3AP and based on the protocol of Konidala *et al.*, is proposed. The security deficiencies in [3] were corrected in M^3AP with the introduction of the complex but lightweight $MixBits$ function. $MixBits$ has been obtained by means of Genetic Programming. Its security and performance has been studied in some depth. In addition, a security analysis of the whole M^3AP protocol has been accomplished, and we find that there is greater resistance against common attacks. In conclusion, we expect that our M^3AP protocol can help to increase the security level of EPC Class-1 Gen-2 / ISO

18000-6C specification.

Acknowledgments

We are grateful to Tong-Lee Lim for his helpful comments and invaluable help.

References

[1] EPCglobal, EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz-960MHz Version 1.2.0, 2008.

[2] International Organization for Standards (ISO), ISO/IEC 18000-6: Radio frequency identification for item management – Part 6: Parameters for air interface communications at 860 MHz to 960 MHz.

[3] D. M. Konidala and K. Kim, "RFID tag-reader mutual authentication scheme utilizing tag's access password", *Auto-ID Labs, White Paper WP-HARDWARE-033*, 2007.

[4] D. N. Duc, J. Park, H. Lee, K. Kim, "Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning", in *Proc. of the Symposium on Cryptography and Information Security*, 2006.

[5] H. Y. Chien, C. H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Gen 2 standards", in *Computer Standards & Interfaces*, vol. 29(2), pp. 254 - 259, 2007.

[6] A. Juels, "RFID security and privacy: a sesearch survey", in *IEEE Journal on Selected Areas in Communications*, vol. 24(2), pp. 381-394, Feb. 2006.

[7] A. Mitrokotsa, M. R. Rieback and A. S. Tanenbaum, "Classification of RFID Attacks", in *Proceedings of the 2nd International Workshop on RFID Technology*, 2008.

[8] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. "RFID Specification Revisited". Book Chapter in *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems*, 6:pp.127156. Auerbach Publications, Taylor & Francis Group, 2008.

[9] T. L. Lim, and T. Li, "Addressing the weakness in a lightweight RFID tag-reader mutual authentication scheme", in *Proc. of IEEE Globecom*, Nov. 2007.

[10] D. Bailey and A. Juels. "Shoehorning security into the EPC standard". In *International Conference on Security in Communication Networks – SCN'06*, volume 4116 of *LNCS*, pp. 303–320. Springer-Verlag, September 2006.

[11] S. Karthikeyan, M. Nesterenko, "RFID security without extensive cryptography", in *Proc. of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 63-67, 2005.

[12] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard, in *Hand. of RFIDsec'07*, July 2007.

[13] M. Burmester, B. de Medeiros, J. Munilla and A. Peinado. "Secure EPC Gen2 compliant Radio Frequency", in *Cryptology ePrint Archive*, Report 2009/149 `http://eprint.iacr.org/`, 2009.

[14] P. Peris-Lopez, J. C. Hernandez-Castro, J.M. Estevez-Tapiador, T. Li, and J. C.A. van der Lubbe. "Weaknesses in two recent lightweight RFID authentication protocols", in *Hand. of RFIDSec '09*, July 2009.

[15] D. M. Konidala, Z. Kim, and K. Kim, "A simple and cost-effective RFID tag-reader mutual authentication scheme", in *Proc. of RFIDSec'07*, pp. 141-152, July 2007.

[16] P. Peris-Lopez, T. Li, T. L. Lim, J. C. Hernandez-Castro and J. M. Estevez-Tapiador. "Vulnerability Analysis of a Mutual Authentication Scheme under the EPC Class-1 Generation-2 Standard", in *Hand. of RFIDSec'08*, July 2008.

[17] T. L. Lim, T. Li. "Exposing an effective denial of information attack from the misuse of EPCglobal standards in an RFID authentication scheme", in *Proc. of IEEE PIMRC*, Sep. 2008.

[18] T. Lohmmann, M. Schneider, and C. Ruland, "Analysis of power constraints for cryptographic algorithms in mid-cost RFID tags", in *Proc. of CARDIS'06*, vol. 3928 of *LNCS*, pp. 278-288, 2006.

[19] J. R. Koza, "Evolving a computer program to generate random number using the genetic programming paradigm", in *Proc. of the 4th Int'l Conf. on Genetic Algorithms*, 1991.

[20] A Klimov and A. Shamir. "Cryptographic applications of T-functions". In *Proc. of SAC'03*, volume 3006 of *LNCS*, pp. 248–261. Springer-Verlag, 2003.

[21] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda. "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol", in *Workshop on Information Security Applications*, Volume 5379 of *LNCS*, pages 56-68. Springer-Verlag, September 23-25, 2008.

[22] J. C. Hernandez-Castro, J. M. Estevez-Tapiador, A. Ribagorda-Garnacho, B. Ramos-Alvarez, "Wheedham: an automatically designed block cipher by means of genetic programming", in *Proc. of CEC '06*, pp. 192–199, 2006.

[23] David Sexton, Randomness Analysis of Konton2, `http://www.geocities.com/da5id65536`, 2005.

[24] M. Hell, T. Johansson, W. Meier. "Grain - a stream cipher for constrained enviroments". in *Proc. of RFIDSec'05*, 2005.

[25] M. Feldhofer, J. Wolkerstorfer, V. Rijmen. "AES implementation on a grain of sand". in *Proc. on Information Security*, vol. 152, pp. 13–20. IEEE Computer Society, 2005.

[26] M. Lehtonen, et al., "Networked RFID systems and lightweight cryptography", in *Chapter from identification to authentication - A review of RFID product authentication techniques*, pp. 169-187. Springer, 2007.

[27] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. "LAMED A PRNG for EPC Class-1 Generation-2 RFID Specification", in *Computer Standards & Interfaces*, Vol. 31(1), pp. 88-97, January 2009.

[28] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: an ultra-lightweight block cipher", in *Proc. of CHES'07*, vol. 4727 of *LNCS*, pp. 450–466, 2007.

[29] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New light-weight crypto algorithms for RFID", in *Proc. of IEEE International Symposium on Circuits and Systems, ISCAS'07*, pp. 1843–1846, 2007.

[30] D. Hong, *et al.*, "HIGHT: a new block cipher suitable for low-resource device, in *Proc. of CHES'06*, LNCS vol. 4249, pp. 46–59, 2006.

[31] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES implementation on a grain of sand", in *IEEE Proc. of Information Security*, vol. 152(1), pp. 13–20, 2005.

[32] T. Good, and M. Benaissa, "Hardware results for selected stream cipher candidates", in `http://www.ecrypt.eu.org/stream/`, 2007.

[33] M. Feldhofer and C. Rechberger, "A case against currently used hash functions in RFID protocols", in *Hand. of RFIDSec'06*, 2006.

[34] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, Matt J.B. Robshaw, and Y. Seurin. "Hash functions and

RFID tags: mind the gap", in *Proc. of CHES'08*, vol. 5154 of *LNCS*, pp. 283–299, 2008.

[35] A. Juels, "Minimalist cryptography for low-cost RFID tags", in *Proc. of SCN'04*, vol. 3352 of *LNCS*, pp. 149-164, 2004.

[36] S. Inoue, and H. Yasuura, "RFID privacy using user-controllable uniqueness", in *RFID Privacy Workshop*, 2003.

[37] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "Keep on Blockin' in the free world: personal access control for low-cost RFID tags", in *Proc. of the 13th Int'l Workshop on Security Protocols*, Apr 2005.

[38] A. Juels, P. Syverson, and D. Bailey, "High-power proxies for enhancing RFID privacy and utility", in *Proc. of PET '05*, 2005.

Pedro Peris-Lopez He has a M.Sc. in Telecommunications Engineering and Ph.D. in Computer Science. His research interests are in the field of protocols design, primitives design, lightweight cryptography and cryptanalysis. Nowadays, he is focusing on Radio Frequency Identification (RFID) systems and doing a PosDoc on RFID security at the Information and Communication Theory Group of Delft University of Technology. For additional information visit: `http://www.lightweightcryptography.com`

Tieyan Li He is a senior researcher at Institute for Infocomm Research (I^{2}R, Singapore) from Oct. 2001. He obtained his Ph.D. degree in 2003 at School of Computing, National University of Singapore. Dr. Li is experienced in practical system developments such as networking, system integration and software programming. He is also active in academic security research fields with tens of journal and conference publications and several patents. Currently his areas of research are in applied cryptography and network security, as well as security issues in RFID, sensor, multimedia and tamper resistant hardware/software, etc. Dr. Li has served as the PC member and reviewer for a number of security conferences and journals.

Julio C. Hernandez-Castro He has a degree in Maths and a MSc in Coding Theory and Network Security. He has worked heavily in the past on the applications of artificial intelligence techniques (notably evolutionary computation) to Cryptography and Cryptanalysis. He got his Ph.D. in Computer Science from Carlos III University in Madrid in 2003 and was there till February 2009 when he was appointed Senior Lecturer in the School of Computing of Portsmouth University in the UK. He has published more than 30 papers in international journals and more than 50 in international conferences. He is a keen chess player.