JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 25, 33-57 (2009)

An Ultra Light Authentication Protocol Resistant to Passive Attacks under the Gen-2 Specification

PEDRO PERIS-LOPEZ, JULIO CESAR HERNANDEZ-CASTRO,
JUAN M. ESTEVEZ-TAPIADOR AND ARTURO RIBAGORDA
Department of Computer Science
Carlos III University of Madrid
28911 Léganes, Madrid, Spain
E-mails: {pperis; jcesar; jestevez; arturo}@inf.uc3m.es

Low-cost Radio Frequency Identification (RFID) tags are devices with very limited computational capability, in which only 250-4K logic gates can be devoted to security-related tasks. Classical cryptographic primitives such as block ciphers or hash functions are well beyond the computational capabilities of low-cost RFID tags, as ratified by the EPCglobal Class-1 Gen-2 RFID specification. Moreover, the Gen-2 RFID specification does not pay due attention to security. For this reason, an efficient Ultra Light Authentication Protocol (ULAP) is proposed in this paper. This new scheme offers an adequate security level against passive attacks, and is compliant with Gen-2 RFID specification.

Keywords: RFID, security, tag, reader, pseudonym, privacy, mutual authentication, EPC Class-1 Gen-2 specification

1. INTRODUCTION

Acceptance of RFID technology has not come as quickly as expected, even though its usage has recently started to blossom in many companies. An indication of this is the report published by IDtechEx where it is envisioned that ten billions tags will be used in 2007, increasing to a trillion tags by 2015 [9]. One of the main problems of RFID technologies suffer and an obstacle to their success, is cost. Tag price should be in the range of 0.05-0.1 € to make it affordable in everyday packaging.

With price being such a limiting factor (with implications for gate count for example), using strong classic cryptographic primitives is unfeasible. For example, for a standard implementation of the Advanced Encryption Standard (AES), between 20K and 30K gates are needed. Additionally, the memory incorporated in a tag is usually limited to a few hundred bits. Their processing capability (measured as the number of equivalent logic gates) is commonly assumed not to be higher than a few thousand gates, and in the case of passive tags a maximum of 4K gates can be devoted to security functions [44]. With regard to power source, tags can be categorized as active or passive. Passive tags, without an internal source of power, are the most commonly used, so their power consumption should be very limited [22]. Furthermore, tags are not at all tamper resistant, so we must accept that tags are not able to store passwords securely.

Nowadays we are in a convergence period between barcodes and RFID technology. While RFIDs are only used in particular situations now, an increase of up to 96% in their

Received May 4, 2007; revised July 31, 2007 & March 24, 2008; accepted April 24, 2008.
Communicated by Tzong-Chen Wu.

usage scenarios is expected this year [49]. Experts believe that both systems will coexist for some time and that finally, RFID tags will completely replace classic barcodes [47].

The remainder of the paper is organized as follows. Section 2 introduces RFID standards, focusing on the Gen-2 RFID specification. A short review of the main problems associated with RFID systems is outlined in section 3. In section 4, related work is presented. Section 5 proposes an Ultra Light Authentication Protocol (ULAP) compliant with the Gen-2 specification. A security evaluation and a performance analysis of this new protocol is presented in section 6. In section 7, the proposed architecture for implementing our protocol is explained in detail. Finally, some concluding remarks are presented in section 8.

2. RFID STANDARDS

The benefits of standards are clear, and assumed by almost everyone. The growth of any new technology is in many cases due in part to the establishment of open standards. To foster and publicize RFID technology, several organizations including EPCglobal (which is a joint venture between EAN International and Uniform Code Council) and ISO have been actively working on RFID standardization [12, 20]. Table 1 summarizes the main standards linked to RFID technology.

Back in 2003, there was a clear lack of harmonization and major RFID vendors offered proprietary systems. Fortunately, things are changing rapidly. One of the most important standards proposed by EPCglobal is the EPCglobal Class-1 Generation-2 RFID specification (known as Gen-2 in short). This standard was adopted by EPCglobal in 2004 and was sent to ISO. Eighteen months later (March-April 2006), it was ratified by ISO, and published as an amendment to its 18000-6 standard. In conclusion, at least for low-cost RFID tags, it seems that we are clearly moving closer to a universal standard. Although standards are increasingly adopted by many companies, some others base the security of the tags they manufacture on proprietary solutions. The use of proprietary solutions is not too bad if algorithms are published so they can be analyzed by the

Table 1. RFID standards.

scope	Standards	
Contactless Integrated Circuit Cards	ISO 10536	Close-couple cards
	ISO 14443	Proximity cards
	ISO15693	Vicinity cards
Animal Identification	ISO 14223	Advanced transponders
	ISO 11784	Code structure
	ISO 11784	Technical concept
Item Management	ISO 18000-1	Reference architecture
	ISO 18000-(2-4)	135 KHz, 13,56 Mhz, 2,45 GHz
	ISO 18000-(6-7)	860-960 MHz, 433 MHz
Electronic Product Code	UHF Class -1 Generation-2	EPC
	ISO 18006-C	

research community. As time has shown, the security of an algorithm cannot reside in "its obscurity." Texas Instruments DST tags [5] and Philips Mifare cards [26] are recent examples of this: companies should learn form past errors and make the algorithm of the cryptographic primitives as widely-known as possible.

Although a rigorous analysis of the Gen-2 RFID specification [12] is out of the scope of this work, we briefly summarize its more relevant properties (we refer the reader to [42] where the specification is analyzed in depth):

- Tags are passive, harvesting energy from the reader signal. Communications must there fore be initiated by readers.
- Tags support on-chip a 16-bit Pseudo-Random Number Generator and a 16-bit Cyclic Redundancy Code.
- Tags are not tamper resistant, so memory is insecure.
- A 32-bit kill PIN is used to make the tag permanently unusable (*i.e.*, tags can be killed at point of sale on purchase) and a 32-bit access PIN is used to access the tag's memory (read/write).
- Readers are assumed to have a secure connection to a back-end database.

3. RISKS AND THREATS

Every now and then RFID news items appear in the mass media, and a fair amount of these represent inaccurate reports about the possibilities for abuse of the technology. Wile alarm concerning privacy issues is completely legitimate, misinformation and hysteria should be avoided so as not to distort the future of RFID technology.

As already predicted in 1991, one of the main problems that ubiquitous computing has to solve is privacy [54]. RFID technology is a pervasive technology, perhaps one of the most pervasive in history. Products labeled with insecure tags reveal sensitive information when queried by readers. Usually, readers need not be authenticated and tags answer in a transparent and indiscriminate way. As an example of the threat this could pose, consider the pharmaceutical sector where tagged medication is planned for the immediate future. Imagine that when you leave the pharmacy with a given drug – say an anti-depressive or AIDS treatment, an attacker standing at the door, equipped with a reader, could find out what medication you have bought. Similarly, wrong-doers equipped with tag readers could search people, pick out those with multiple tagged bank bills to rob, and at the same time know how much money they will stand to gain with each robbery.

Moreover – even if we could assure that tag's contents were secure – this does not mean that protection against tracking (violation of location privacy) is guaranteed. Tags usually answer the readers "*hello*" queries with the same identifier. Such predictable tag responses allow a third party to establish an association between a tag and its owner. Even where individual tags only contain product codes rather than an unique serial number, tracking could still be possible using assembly of tags (constellations) [52].

In addition to the threats mentioned above, some other security-related aspects must be considered. RFID technology operates over radio, so communication can be eavesdropped. In the reader-to-tag channel (forward channel) the reader broadcasts a strong signal, allowing its monitoring from a long distance. The signal transmitted by the tag-to-

reader (backward channel) is relatively weak, and it may be only monitored closed to the tag. Even if the communications are encrypted, traffic analysis is feasible, extracting information from the patterns of communication. In general, the outcome of this analysis improves as the number of observed messages increases.

Tag memory is insecure, and susceptible to physical attacks that can reveal the contents completely. As these physical attacks generally involve manipulating the tag with precision equipments, they are often carried out in laboratories. Probe attacks, material removal through shaped charges or water etching, radiation imprinting, circuit disruption and clock glitching are examples of these kind of attacks. Counterfeiting (modifying the identity of an item) usually involves some of these types of attacks.

Denial of service is another important threat. Attackers could disrupt the normal operation of tags and readers by jamming the RF channel. This attack consists of setting an active device to continuously broadcast radio signals into the channel. This is an inherent problem occurring in all the technologies that use radio as their communication channel. On the other hand, tags can be disabled, for example by cloaking a tag from readers. This attack must be prevented by means of traditional countermeasures such as cameras or security guards.

4. RELATED WORK

The major challenge faced when trying to provide security for low-cost RFID tags is their very limited computational capability, making them unable to perform the most basic cryptographic operations. Surprisingly, most of the proposed solutions are based on the use of hash functions. Since the work of Sarma [46] in 2002, there has been a huge number of solutions based on this idea [7, 10, 18, 38, 57]. Although this apparently constitutes a good and secure approach, engineers face the nontrivial problem of implementing cryptographic hash functions with only 250-4K gates [44]. In most of the proposals, no explicit algorithms are suggested and finding one is not an easy issue, since traditional hash functions (MD5, SHA-1, SHA-2) cannot be used [14]. In [58] we find a recent work on the implementation of a new hash function with a reduced number of gates, but although this proposal seems to be light enough to fit in a low-cost RFID tag, the security of the hash scheme is questionable. Furthermore, the above mentioned proposals do not comply with the Gen-2 RFID specification, because hash functions are not supported on Gen-2 RFID tags.

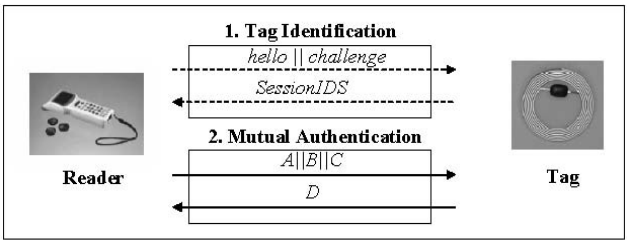
In 2003, and before the Gen-2 standard was proposed, some authors started to suggest the use of non-cryptographic primitives. Although the derived protocols were Gen-2 compliant, their security is weak. Vajda *et al.* proposed a set of extremely-lightweight challenge-response authentication algorithms [50]. These can be used for authenticating the tag, but they can easily be broken by a powerful adversary. Moreover, these algorithms do not solve important problems such as reader-to-tag authentication or tracking (to name just a few). In [23], Juels proposed a solution based on the use of pseudonyms, without the use of hash functions. The RFID tag stores a short list of pseudonyms, rotating them and releasing a different one with each reader query. An adversary can, however, gather all the names on the list by querying the tag multiple times. To avoid this, after a number of authentication sessions the list of pseudonyms must be updated on a different out-of-band communication channel.

Following the approach started by Vajda and Juels, Peris *et al.* proposed in 2006 some new lightweight protocols. Chronologically, M^{2}AP [40] was the first proposal, followed by EMAP [39] and LMAP [43]. All these protocols are based on the assumption that only simple operations are available on-tag. Although the protocols are different, the framework is similar. They can be divided into four main stages: tag identification, mutual authentication, index-pseudonym and key updating. M^{2}AP and LMAP use the bitwise or, and, xor and sum mod 2^M operators. EMAP uses a similar set of operators, excluding the sum. Instead, a parity function has been added to increase the security level of the key update stage. This is why EMAP needs around 60% less logic gates when compared with the other proposals. On the other hand, EMAP has to perform a higher number of operations, which means it can produce around half the number of answers as the other two protocols. The gate count of LMAP and M^{2}AP is very close, as both share the same set of operators. However, an important advantage that LMAP has over M^{2}AP is that the communication overhead in the forward channel has been reduced, because only one submessage has to be sent. This reduction means an increment of around 15% in the number of the answers per second, and a reduction of the power consumption. Finally, some weaknesses of M^{2}AP are identified and corrected in the LMAP protocol. LMAP can be considered an early version of ULAP, the protocol presented here, although there are major differences between them, notably the update of the *SessionIDS* independently of the result (success/fail) of the authentication, and some important modifications in the key updating algorithm.

Unfortunately, Gen-2 RFID specification does not pay due attention to security. That is why there have recently been many proposals to enhance the security of the standard. However, these schemes continue to demonstrate important security flaws. Juels suggested a scheme to prevent cloned tags from impersonating legitimate tags [24]. However, his protocol neglected to take eavesdropping and privacy issues into consideration, and therefore it provides no protection against privacy invasion and secret information leakage [37]. Karthikeyan *et al.*'s proposal is based on the use of xor and matrix operations [27]. This scheme is not resistant to DoS and replay attacks, and tracking is also possible [6]. Another interesting work is Due *et al.*, where only a PRNG and a CRC are used [37]. However, DoS attacks are also possible against this proposal, impersonated tags can not be identified, and forward privacy is not guaranteed [6]. In 2007, two protocols attempting to correct some of the security shortcomings of the EPC-C1G2 were proposed by Chien *et al.* [6] and Konidola *et al.* [28]. The first proposed scheme is based on a PRNG and a CRC and the second one uses a pad generation function. Later, Peris *et al.* [41] and Lim *et al.* [33] show important security vulnerabilities in both proposals.

5. ULTRA LIGHT AUTHENTICATION PROTOCOL

As many other authors, we think the security of low-cost RFID tags can be greatly improved without the use of classic cryptographic primitives (*i.e.* block/stream ciphers, hash functions, *etc.*). We also believe that new protocols should take standards into account, in particular the Gen-2 RFID specification. An Ultra Light Authentication Protocol (ULAP), compliant with the Gen-2 specification, is proposed in this article (see Fig. 1).



$$A = SessionIDS_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} \oplus n1 \quad (1)$$

$$B = (SessionIDS_{tag(i)}^{(n)} \vee K2_{tag(i)}^{(n)}) + n1 \quad (2)$$

$$C = SessionIDS_{tag(i)}^{(n)} + K3_{tag(i)}^{(n)} + n2 \quad (3)$$

$$D = (SessionIDS_{tag(i)}^{(n)} + ID_{tag(i)} \oplus n1 \oplus n2) \quad (4)$$

Fig. 1. ULAP protocol.

5.1 Suppositions of the Model

Our protocol is based on the usage of pseudonyms, specifically *index-pseudonyms* (*IDSs*), and the related *session index-pseudonyms* (*SessionIDSs*). An (*L*-bit length) *IDS* is a unique index of a table row where all the information about a tag is stored. The *IDS* is never sent through the channel, instead an *L*-bit *SessionIDS* is transmitted. Each tag has an associated key, which is divided into four *L* bits components ($K = K1 \,||\, K2 \,||\, K3 \,||\, K4$). As the *IDS*, *SessionIDS* and key (*K*) should be updated, we need 6*L* bits of rewritable memory (EEPROM or FRAM) in total. A ROM memory to store the *L*-bit static identification number (*ID*) is also required.

For the implementation of our protocol, all costly operations such as random-number generation will be performed by the reader. On the other hand, as tags are very computationally limited devices, only the simplest operations are available: bitwise xor ($\oplus$), bitwise or ($\vee$), and sum mod 2^m (+). Multiplication can hardly be used, as it is quite a costly operation [34]. Although it would seem that this seriously limits the strength of the resulting protocol, there are other proposals in cryptography that obtain an adequate security level by composing very simple and efficient operations, such as Salsa20 [4], TEA [55] and XTEA [56]. Furthermore, it is interesting to remember that any boolean function can be implemented by using {AND, OR, NOT} or {NAND} gates [16].

As most low-cost tags are passive, the communication must be initiated by the reader. We also suppose that both the backward and forward channel can be passively listened by an attacker. Note that this assumption implies that the air channel cannot be actively manipulated by an adversary; we assume therefore that man-in-the-middle and other active attacks are not feasible [30]. Our protocol is not secure against active attacks (see section 6.1). Finally, we also assume that the communication channel between the reader and the database is secure.

5.2 The Protocol

We can split our protocol proposal in four main stages: tag identification, mutual authentication, index-pseudonym updating, and key updating.

5.2.1 Tag identification

Before starting the mutual authentication, the reader has to identify the tag. In a first naive approximation, the following mechanism could seem appropriate: the reader sends a *hello* message to the tag, which answers by sending its current *IDS*. By means of this *IDS*, the authorized reader (and only he) will be able to access the tag's secret key ($K = K1 \,||\, K2 \,||\, K3 \,||\, K4$), which is necessary to carry out the next authentication stage. After a successful authentication, both the reader and the tag update the *IDS*. In these conditions, however, tracking could be possible, since *IDS* updating is only accomplished after a successful mutual authentication. This suggests the importance of carrying out an *IDS* update each time the tag is interrogated, no matter if the protocol completion results in a successful or unsuccessful authentication. As mentioned in section 4, the most common solution for this is based on the use of a hash function. This solution has three problems: it exceeds the capabilities of low-cost RFID tags, it is non-compliant with the Gen-2 specification and implies the necessity of an exhaustive search in the back-end database. To solve these problems the following mechanism is proposed: the reader sends a *hello* message concatenated with a random number (challenge) to the tag, and the tag answer will consist of a *SessionIDS*. The *SessionIDS* will be computed in the following way:

```
SessionIDS = IDS
for (i = 0; i < 32; i++) {
SessionIDS = (SessionIDS >> 1) + SessionIDS + SessionIDS + challenge; }
```

In order to obtain this *SessionIDS* update function, we used Genetic Programming [29] and the lil-gp library [1] for finding highly nonlinear functions. This was accomplished using the avalanche effect as the key component of the fitness function. In fact, an even more demanding property was used: the Strict Avalanche Criterion [15], mathematically described by:

$$\forall x, y \mid H(x, y) = 1, H(F(x), F(y)) \approx B(n, \frac{1}{2}). \tag{5}$$

So, if F has the Strict Avalanche Criterion, the hamming distance between the output of a random input vector and one generated by randomly flipping one of its bits follows a binomial distribution with parameters n and $\frac{1}{2}$.

To fix the length of the challenge, the following scenario is considered: imagine that an attacker is eavesdropping the answers provided by a tag over one week, and the tag provides 500 answers/second, which is quite beyond current commercial rates. Under these conditions, the tag would generate around 2^{23} *SessionIDSs*, so we suggest setting the length of challenges to 32-bits.

We have used a technique named linear cryptanalysis (commonly used for block cipher cryptanalysis) to examine wether the *SessionIDS* update function can be approximated by a linear relation. 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 are generated (O_i, O_{i+1}). With these two masks, the equality $A * O_i = B * O_{i+1}$ is evaluated. This process is repeated 2^n times, computing the numbers of successes (m). The * sym-

bolizes a scalar product, but a mod 2 operation is carried out after the addition. The bias is defined as:

$$bias = \frac{1}{2^{-\log_2(|\frac{m}{2^n}-\frac{1}{2}|)}}. \quad (6)$$

Due to the high computational cost of bias estimation, the length of variables are fixed to 32-bits. Instead of picking a random number for the challenge in each simulation, once the challenge is randomly initialized it will be incremental updated (challenge + 1, challenge + 2, challenge + 3, *etc.*) in order to consider a partial advantageous scenario. Many pairs of different masks, A and B, have been randomly tested. After a random initialization of the *IDS* and the challenge value (obtained from http://random.org), and for each mask pair, 2^{25} 32-bits outputs are generated, and the expression $A * O_i = B * O_{i+1}$ is evaluated over them. From the calculations described above, we can deduce that the bias of the *SessionIDS* update function ($L = 32$) is bounded by $\frac{1}{2^{14.56}}$.

The serial correlation coefficient (autocorrelation) has also been studied, to measure the extent to which a new *SessionIDS* depended upon the previous *SessionIDSs* value. For that, the following experiment is implemented: the *IDS* and the challenge are randomly initialized, obtained from http://random.org. After this initialization, 2^{16} *SessionIDS* are computed. As in bias estimation, after the random initialization of the challenge, it is incrementally updated. The experiment is repeated five times, counting in each case the autocorrelation coefficient at bit, byte, and 4-byte levels over the obtained *SessionIDS*. Table 2 summarizes the observed results, showing that the new *SessionIDS* update function has very good properties.

Table 2. Serial correlation test.

	SessionIDS		
Experiment	Bit	Byte	4-Byte
1-Experiment	0.00931	0.001033	0.005278
2-Experiment	0.001104	0.003032	0.006295
3-Experiment	– 0.000198	– 0.000122	0.007116
4-Experiment	0.001257	– 0.002537	0.009762
5-Experiment	0.000435	0.002272	0.005875

The use of the *SessionIDS* update function resolves the three connected problems found in schemes based on hash functions:

1. Simple operations have been ratified by Gen-2 specification, thus guaranteeing the conformance of ULAP protocol with the standard.
2. Simple operations are not highly demanding in terms of resources. Implementation can therefore be initiated realistically, without exceeding the limited resources of low-cost RFID tags.
3. A linear search ($f(x, challenge)$) is accomplished in the database each time a tag is read. The function f symbolizes the *SessionIDS* update function.

In this way, each time a tag is interrogated by a reader, it will answer with a fresh *SessionIDS*. As described in section 5.2.3, once a mutual authentication has been successfully accomplished, the *IDS* will be updated. For example, suppose that the tag (A) is interrogated twice by an attacker (D), before an authorized reader (R) interrogates it, and the attacker then again interrogates the tag. In this scenario the following messages will be transmitted:

(D) -> (A): "hello" || challenge
(A) -> (D): *SessionIDS

(D) -> (A): "hello" challenge′
(A) -> (D): *SessionIDS′

(R) -> (A): "hello" || challenge″
(A) -> (R): *SessionIDS″
Mutual Authentication …
Update(IDS) = IDS′

(D) -> (A): "hello" || challenge‴
(A) -> (D): **SessionIDS‴
---
* SessionIDS = IDS
for (i = 0 ; i < 32; i++) {
SessionIDS = (SessionIDS >> 1) + SessionIDS + SessionIDS + challenge; }
** SessionIDS = IDS′
for (i = 0; i < 32; i++) {
SessionIDS = (SessionIDS >> 1) + SessionIDS + SessionIDS + challenge; }

5.2.2 Mutual authentication

This phase of our protocol consists of the exchange of two messages between the reader and tag. The protocol works as follows:

1. **Reader Authentication** The reader will generate two nonces $n1$ and $n2$. With $n1$ and subkeys $K1$ and $K2$ the reader will generate submessages A and B. With $n2$ and $K3$, it will generate submessage C.
2. **Tag Authentication** With submessages A and B, the tag will authenticate the reader and obtain $n1$. From submessage C, the tag will obtain the random number $n2$. Nonces $n1$ and $n2$ will be used in the *IDS* and key ($K = K1 \,||\, K2 \,||\, K3 \,||\, K4$) updating. Once these verifications are performed, the tag will generate the answer message D to authenticate and send its static identifier securely.

To evaluate the protocol, the following experiment was carried out: *IDS*, *challenge*, and $K = K1 \,||\, K2 \,||\, K3 \,||\, K4$ are randomly initialized with values from http://ramdom.org. After the initialization, 2^{25} executions of the protocol are simulated (tag identification,

Table 3. Message sequence analysis (ENT, DIEHARD and NIST).

	A	B	C	D
Entropy (bits/byte)	7.999999	7.999999	7.999999	7.999999
Compression Rate	0%	0%	0%	0%
X2 Statistic	253.50 (50%)	247.10 (50%)	246.90 (50%)	255.94 (50%)
Arithmetic Mean	127.5054	127.4977	127.4901	127.5035
Monte Carlo π Estimation	3.1413 (0.01%)	3.1419 (0.01%)	3.1414 (0.01%)	3.1414 (0.01%)
Serial Correlation Coefficient	– 0.000145 (byte)	0.000094 (byte)	0.000009 (byte)	0.000113 (byte)
	– 0.000040 (4-byte)	0.0000980 (4-byte)	– 0.000398 (4-byte)	0.0003921 (4-byte)
Diehard Battery (Overall p-value)	0.943171	0.569317	0.267451	0.851599
NIST Battery	√	√	√	√

mutual-authentication, index-pseudonym and key update), storing the submessages sent by the channel (A, B, C, D) in a file. In each simulation, the challenge is incrementally updated (disadvantageous scenario) in spite of the fact that it should be a random number, and the nonces $n1$ and $n2$ are randomly picked. The statistical properties of these four submessages have been analyzed with three well-known suites of randomness tests, namely ENT [51], DIEHARD [35] and NIST [48]. The results are summarized in Table 3. Due to the huge amount of p-values generated by the NIST statistical battery, the report is not shown here.[1] Results indicate that messages are not easily distinguished from a random source, not even for the eavesdropper/cryptanalyst.

We have put particular emphasis on the security and statistical properties of submessage D, since it is in this that the tag sends its more valuable information: the static identification number – ID. As shown in Eq. (4), message D uses an xor of two nonces ($n1$, $n2$) sent by the reader to disguise the information.

To complete the analysis we will examine how an eavesdropper might obtain advantage by listening to previous communications between a reader and a tag. For this purpose, the bit-byte prediction test of David Sexton's battery has been employed [2]. Specifically, various algorithms are used to predict the value of each bit (byte) from the beginning of the sequence to the end. In a random sequence the probability of success of any algorithm is 1/2 (respectively 1/256). The number of successes is counted and a chi-squared statistic is computed. The following tests have been carried out:

- Bit Prediction A Test: the numbers of zeros and ones in all the previous bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted. Otherwise the prediction is the same as for the previous bit.
- Bit Prediction B Test: the numbers of zeros and ones in the previous 9 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Bit Prediction C Test: the numbers of zeros and ones in the previous 17 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber

[1] The whole report is available in http://163.117.149.208/pperis/ulap/.

the ones, a one is predicted.

- Bit Prediction D Test: the numbers of zeros and ones in the previous 33 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Bit Prediction E Test: the numbers of zeros and ones in the previous 65 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Byte Prediction A Test: the next byte is predicted to be equal to all the previous bytes bitwise XORed together. The first byte of the sequence is predicted to equal zero.
- Byte Prediction B Test: the next byte is predicted to be equal to the sum of all the previous bytes, modulo 256. The first byte of the sequence is predicted to equal zero.
- Byte Prediction C Test: the next byte value is predicted to be zero until the first zero is found. From that point on, the next byte value is predicted to be the byte value whose last appearance was furthest back in the sequence.
- Byte Prediction D Test: a given byte value is predicted to be followed by the same byte value it was followed by the last time it appeared in the sequence. A byte value that has not previously appeared in the sequence is predicted to be followed by the byte value of the first byte in the sequence. The first byte of the sequence is predicted to equal zero.
- Byte Repetition Test: This test is equivalent to a byte prediction test where each byte is predicted to be equal to its preceding byte. The first byte of the sequence is predicted to equal the last byte of the sequence.

Under the aforementioned conditions, 2^{32} executions of the protocol were executed to generate data for Sexton's random batteries. The results obtained from these analyses are summarized in Table 4. Prediction analysis does indicate that the exchanged messages can be predicted significantly better than just by knowledge of the prior protocol executions without acquaintance with the *IDS*, *K*, and *ID*.

As mentioned in section 3, one of the main security problems that RFID technology must solve is tracking, or the violation of location privacy. Even if tag answers are not predictable, a slight relation between consecutive answers could help the attacker. Table 3 shows the autocorrelation coefficient at byte and 4-byte level. Additionally, the

Table 4. Bit-byte prediction tests of David Sexton's battery.

	A	B	C	D
Bit Prediction A Statistic	0.1029	0.6287	0.8053	0.1737
Bit Prediction B Statistic	0.3144	0.2104	0.9050	0.1898
Bit Prediction C Statistic	0.2844	0.3111	0.2210	0.8481
Bit Prediction C Statistic	0.8706	0.5648	0.1164	0.4428
Bit Prediction E Statistic	0.8979	0.5868	0.5702	0.6794
Byte Prediction A Statistic	0.6523	0.3045	0.1670	0.8995
Byte Prediction B Statistic	0.3612	0.7907	0.2883	0.9126
Byte Prediction C Statistic	0.8215	0.2178	0.2302	0.1838
Byte Prediction D Statistic	0.1411	0.8353	0.9312	0.1075
Byte Prediction E Statistic	0.5527	0.6221	0.7480	0.6738

Table 5. Correlation coefficient between K and IDS.

	A	*B*	*C*	*D*
A	–	– 0.004964	0.005054	0.002706
B	– 0.004964	–	0.004221	– 0.001973
C	0.005054	0.004221	–	– 0.004059
D	0.002706	– 0.001973	– 0.004059	–

correlation coefficient is computed to determine the relation between different submessages ($X - Y$). Results are presented in Table 5. From the results obtained, there is no evidence of relation between different submessages.

5.2.3 Index-pseudonym and key updating

After the mutual authentication phase, our protocol prescribes an index-pseudonym and key updating stage. Again, this should be carried out using only very efficient operations ($\oplus$, $\vee$, and sum mod 2^m). As all these operations have already been implemented in the tag for the previous protocol stages, its will not imply an increase in the gate count. Nevertheless, we should take into account the temporary requirements to ensure an adequate answer ratio. Considering all these constraints, the equations for the index-pseudonym and key updating phase are the following:

$$IDS_{tag(i)}^{(n+1)} = (IDS_{tag(i)}^{(n)} + (n2 \oplus K4_{tag(i)}^{(n)})) \oplus ID_{tag(i)}, \tag{7}$$

$$K1_{tag(i)}^{(n+1)} = K1_{tag(i)}^{(n)} \oplus n2 \oplus (K3_{tag(i)}^{(n)} \oplus ID_{tag(i)}), \tag{8}$$

$$K2_{tag(i)}^{(n+1)} = K2_{tag(i)}^{(n)} \oplus n2 \oplus ID_{tag(i)}, \tag{9}$$

$$K3_{tag(i)}^{(n+1)} = (K3_{tag(i)}^{(n)} \oplus n1) + (K1_{tag(i)}^{(n)} \oplus ID_{tag(i)}), \tag{10}$$

$$K4_{tag(i)}^{(n+1)} = (K4_{tag(i)}^{(n)} \oplus n1) + ID_{tag(i)}. \tag{11}$$

The analysis done on the statistical properties of the output of these four subkey updating sequences show good behavior, something that is hardly surprising as an xor with a nonce ($n1$ or $n2$) is performed in each.

On the other hand, in the index-pseudonym update, we have used a sum instead of an xor operation, so we have no guarantees of good statistical properties and we should analyze its output. Under the same initialization conditions used to analyze the mutual-authentication stage, 2^{25} updates of the *IDS* were simulated, storing the resulting values in a file. Table 6 shows the obtained results. The whole report is also available in http://163.117.149.208/pperis/ulap/. From these results, we can conclude that there is no evidence of the resulting *IDS* being significantly different from a random variable.

An additional study has been performed to show that the relation between consecutive index-pseudonyms (*IDS*) or keys (*Ki*) is negligible. Tables 6-8 show the autocorrelation and correlation coefficient respectively, for these variables. In view of the results, it is not possible to infer any signs of dependence between these variables.

Table 6. IDS analysis.

	IDS
Entropy (bits/byte)	7.999999
Compression Rate	0%
X2 Statistic	251.75 (50%)
Arithmetic Mean	127.4930
Monte Carlo π Estimation	3.1417
Serial Correlation Coefficient	0.000078
	0.000393
Diehard Battery (Overall p-value)	0.619483
NIST Battery	√

Table 7. Serial correlation coefficient – K.

	Serial Corr. (byte)	Serial Corr. (4-byte)
$k1$	– 0.000286	– 0.003269
$k2$	0.000472	0.004281
$k3$	0.000592	0.000592
$k4$	0.000112	0.003935

Table 8. Correlation coefficient between K and IDS.

	$k1$	$k2$	$k3$	$k4$	*IDS*
$k1$	–	– 0.004246	0.001400	– 0.002992	0.001578
$k2$	– 0.004246	–	– 0.000806	0.004388	– 0.000200
$k3$	0.001400	– 0.000806	–	– 0.000520	– 0.001907
$k4$	– 0.002992	0.004388	– 0.000520	–	– 0.000552
IDS	0.001578	– 0.000200	– 0.001907	– 0.000552	–

Table 9. Bit-byte prediction tests of David sexton's battery.

	$k1$	$k2$	$k3$	$k4$
Bit Prediction A Statistic	0.5851	0.6649	0.8915	0.1137
Bit Prediction B Statistic	0.7729	0.7194	0.6326	0.4167
Bit Prediction C Statistic	0.7775	0.1012	0.8077	0.6562
Bit Prediction C Statistic	0.4677	0.1834	0.6672	0.7069
Bit Prediction E Statistic	0.5146	0.1374	0.8718	0.8279
Byte Prediction A Statistic	0.5353	0.8930	0.9307	0.7516
Byte Prediction B Statistic	0.1310	0.2657	0.1103	0.2303
Byte Prediction C Statistic	0.4930	0.5000	0.3021	0.6024
Byte Prediction D Statistic	0.1815	0.1147	0.6374	0.1852
Byte Prediction E Statistic	0.5868	0.8161	0.8209	0.2562

Although the key ($K = K1 \,||\, K2 \,||\, K3 \,||\, K4$) is not sent in clear over the channel, and as tags are not tamper resistant, if the attacker has physical access to the tag, keys might be obtained. The bit-byte prediction tests of David Sexton's battery have been computed [2] in order to study if the (*IDS*) or keys (Ki) can be easily derived from other (previous or past) occurrences. Much as in the analysis of the submessage sequences (mutual-authentication stage) and under the same conditions, 2^{32} executions of the index-pseudonym and key update are completed. The results of the analysis, shown in Table 9, do not identify any significant advantage for the attacker.

6. EVALUATION

6.1 Security Needs

As any other mission-critical system, it is important to minimize the threats to confidentiality, integrity, and availability (CIA) in data and computing resources. These three factors are often refereed to as "The Big Three." However, not all systems need the same security level. For example, not all systems need 99,999% availability or require that users be authenticated via retinal scans. Because this, it is necessary to analyze and evaluate each system (data sensitivity, loss potential due to incidents, criticality of the mission, *etc.*) to determine the confidentiality, integrity, and availability requirements.

Therefore, although from a theoretical viewpoint active attacks might be considered, not all system must provide resistance to this kind of attack. In the RFID context, the security level of a high-cost tag used in e-passports should not equal that of a low-cost tag employed in supply chain (*i.e.* tags compliant to Gen-2). Table 10 summarizes the specifications that are realistic with a current low-cost RFID tag and an e-passport (high-cost tag).

Table 10. Specifications for a low-cost and a high-cost RFID tags.

	Low-cost RFID Tag	High-cost RFID tag
Standards	EPC Class-1 Generation-2 ISO/IEC 18006-C	ISO/ICE 14443 A/B
Power Source	Passively powered	Passively powered
Storage	32 - 1K bits	32 KB – 70 KB
Circuitry (security processing)	250 – 4K gates Standard cryptographic primitives cannot be supported	Microprocessor Implement 3DES, SHA-1, RSA
Reading distance (commercial devices)	Up to 3m	Around 10 cm
Price	0.05 – 0.1 €	Several euros
Physical attacks	Not resistant	Tamper resistant EAL 5+ security level
Resistance to passive attacks	Yes	Yes
Resistance to active attacks [8, 25, 27, 30, 37]	No	Yes

6.2 Security Analysis

Complementing the statistical study of the exchanged messages shown above, a more detailed security analysis is presented below. Although a formal analysis is possible, it is common practice when a new protocol is presented [6, 10, 23, 37, 38, 50] to analyze its security against the most relevant attacks, as done here. For each attack, we describe it then show the ULAP defense against it.

1. **User Data Confidentiality** The tag *ID* must be kept secure to guarantee the user's privacy. The tag sends message D $(D = (SessionIDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2)$ thus

hiding the tag *ID* to any eavesdropper equipped with an RFID reader.

2. **Tag Anonymity** The location privacy of tag holders can be revealed even when the information on the tag is securely protected. For example, if the messages sent by the tag are well encrypted, the leakage of information is not possible. However, as the tag answers are constant, a tag could be easily associated with its holder. Specifically, location privacy can be more significant when a certain tag is exposed to long-term tracking [21]. It is therefore crucial to make all the information sent by the tag anonymous. This property is guaranteed in all phases of the protocol by generating new *SessionIDSs* in each session.
   As seen in section 5.2.1, tags always send a fresh *SesionIDS*, which is obtained by means of a *SessionIDS* update function with a high avalanche-effect. This property prevents successive tag answers from disclosing any kind of information. In the mutual-authentication stage tags have to send their static identifier *ID* and other additional information to accomplish the tag-to-reader authentication. To avoid tracking, all the exchanged messages should be sent in seemingly random wraps (*i.e.* to an eavesdropper, random numbers are sent). As we have seen, the reader generates the message $A \,||\, B \,||\, C$. This message serves to authenticate him, as well as to transmit nonces $n1$ and $n2$ to the tag securely. These two nonces will be used to hide tag *ID*, as well as to update the *index-pseudonym* and the associated key.
   By means of these two mechanisms, we are able to put most of the computational load on the reader side. Tag anonymity is thus guaranteed and privacy location of the tag owner is not compromised.
3. **Data Integrity** A portion of the tag's memory is rewritable, so modifications are possible. In this part of the memory, the tag stores the index-pseudonym (*IDS*), session index-pseudonym (*SessionIDS*) and its key (*K*). If an attacker does succeed in modifying this part of the memory, the reader will not recognize the tag (and should implement an update protocol on the database).
   Furthermore, as we mentioned in sections 3 and 6.1, low-cost tags are not tamper resistant. So this kind of tag is susceptible to physical attacks, with the resulting revelation of their content.
4. **Mutual Authentication** We have designed the protocol with both reader-to-tag authentication (message $A \,||\, B \,||\, C$) and tag-to-reader authentication (message D).
5. **Forward Security** Forward security is the property that guarantees that the security of messages sent today will be valid tomorrow [38]. Since the key updating is fulfilled after the mutual authentication, a security breach of an RFID tag will not reveal data previously transmitted.
6. **Replay Attack Prevention** An eavesdropper could store all the messages exchanged between the reader and the tag (from different protocol runs). Then he could try to impersonate a reader, re-sending the message $A \,||\, B \,||\, C$ seen in any of the previous protocol runs. It may seem that this could cause loss of synchronization between the database and the tag, but this is not the case because after a protocol execution, a new *SessionIDS* will be used. Additionally, if the authentication is successful, the index-pseudonym (*IDS*) and the key K ($K = K1 \,||\, K2 \,||\, K3 \,||\, K4$) will be updated.
7. **Forgery Resistance** The information stored in the tag is sent disguised ($\oplus$, $\vee$, and $+$) with nonces ($n1$, $n2$). Straightforward copying of information of the tag by eavesdropping is therefore not possible.

Additionally, our protocol is secure against the "skimming" attack described by Juels [24]. In this attack, an attacker scans the exchanged messages between an authorized reader and a tag. Then these messages sent by the tag, in our case *SessionIDS* and *D*, are stored in a fake RFID tag. The attacker would wish to pretend this were the legitimate tag. However, each time a tag is read, a new *SessionIDS* is computed. The *SessionIDS* is used in the answer of the *hello* || *challenge* message and in the *D* message. So the reader will see that these messages are a replay. In fact, the protocol will be stopped when the replay *SessionIDS* message is sent.

8. **Data Recovery** The interception or blocking of messages is a DoS attack preventing tag identification. As we do not consider these attacks a serious problem for very low-cost RFID tags, our protocol does not particularly focus on providing data recovery.
9. **Active Attacks** As mentioned in section 5.1, we assume that the attacker is unable to perform man-in-the-middle attacks. ULAP can then be implemented in low-cost RFID tags and provides defense against a great number of attacks, excluding active attacks. A detailed explanation justifying this assumption is found in section 6.1.
   In [31, 32], the security of the ultra lightweight MAP protocols family proposed by Peris *et al.* (M^2AP, EMAP, LMAP) have been recently analyzed . First, a desynchronization attack is proposed. This attack is based on a man-in-the-middle; an attacker intercepts the answer provided by the reader and alters the message content (*i.e.* flipping one bit of the submessages sent). Then the answer backscattered by the tag is modified in the same way. This attack can be accomplished on A and/or B and/or C submessages. Secondly, a full-disclosure attack which enables an attacker to disclose tag *ID* is proposed. However, this attack can not be applied to ULAP. The above attacker consists on listening to several authentication sessions with the same *IDS*. In the proposed scheme, when an attacker interacts with a tag (tag identification + tag authentication) and independently of whether authentication is successful, a new *SessionIDS* is employed, frustrating the attack. Additionally, the attack assumes that an attacker can run the in-complete protocol many times, which is an incorrect assumption. In [19], a more efficient full-disclosure attack is proposed. As the attack is based on the same assumptions as Li *et al.*'s attack, it is not feasible against ULAP either.

Table 11 shows a comparison of the security requirements of different proposals in the literature. We have added our proposal (ULAP) in the last column.

The departure point for this analysis is Avoine's work, who proposed an adversary model suitable for RFID environments where existential and universal untraceability are defined [3]. As we have seen in the security analysis, the information sent by the tag gives no useful information to an attacker. The reader sends no useful information either. Consequently, ULAP is *Existential-UNT-QSE* according to Avoine's classification. By tampering with the tag, an attacker can obtain its current *index-pseudonym*, *SessionIDS*, and key, but he can not track the tag's past events because the key and *index-pseudonym* is updated after each mutual authentication. ULAP is therefore *Forward-UNT-QSER*.

6.3 Performance Analysis

It is important to carefully analyze the performance of the proposed scheme in order to show that it can be implemented even in very low-cost tags, such as tags conforming to the Gen-2 specification. Computation, storage, and communication overheads will be

Table 11. Security analysis.

Protocol	HLS [53]	EHLS [53]	EHC [38]	PSL [36]	SA [13]	HBVI [18]	MAP [57]	ULAP
User Data Confidentiality	×	△	○	○	○	△	○	○
Tag Anonymity	×	△	○	○	△	△	○	○
Data Integrity	△	△	△	△	△	○	○	△
Mutual Authentication	△	△	△	○	△	△	○	○
Forward Security	△	△	○	×	×	○	○	○
Replay Attack	△	△	△	○	△	○	○	○
Forgery Resistance	×	×	○	○	○	△	○	○
Data Recovery	×	×	×	×	×	○	○	×

Notation: ○ Satisfied △ Partially Satisfied × Not Satisfied

Table 12. Computational loads and required memory.

Protocol	Entity	HLS [53]	EHLS [53]	EHC [38]	PSL [36]	SA [13]	HBVI [18]	MAP [57]	**ULAP**
N. of Hash Operation	T	1	2	2	¬	¬	3	2	¬
	R+B	¬	Nt	2Nt	¬	¬	3	2Nt	¬
N. of Keyed Hash Operation	T	¬	¬	¬	1	¬	¬	¬	¬
	R+B	¬	¬	¬	Nt	¬	¬	2	¬
N. of RNG Operation	T	¬	1	¬	1	¬	¬	¬	¬
	R+B	¬	¬	¬	1	1	¬	1	¬
N. of Lightweight Crypt. Operation	T	¬	¬	¬	¬	¬	¬	¬	1^2
	R+B	¬	¬	¬	¬	¬	¬	¬	1^2
N. of Basic Operation[1]	T	¬	¬	¬	2	¬	¬	4	16
	R+B	¬	¬	¬	Nt	¬	¬	2(Nt+1)	19
N. of En-Decryption	T	¬	¬	¬	¬	1	¬	¬	¬
N. of En-Decryption	R+B	¬	¬	¬	¬	Nt	¬	2	¬
N. Authentication Steps		6	5	2	3	2	5	5	4
Required Memory Size[2]	T	$1\frac{1}{2}L$	$1L$	$2L$	$4\frac{1}{2}L$	$1\frac{1}{2}L$	$3L$	$2\frac{1}{2}L$	$7L$
	R+B	$2\frac{1}{2}L$	$1\frac{1}{2}L$	$2L$	$4\frac{1}{2}L$	$2\frac{1}{2}L$	$3L$	$9\frac{1}{2}L$	7L

Notation: ¬ Not Required Nt: Number of Tags L: Size of Required Memory
[1] Basic Operations: $\oplus$, $\vee$ and + (sum *mod* 2^m) [2] SessionIDS Update Function

analyzed. Additionally, a comparison of the performance of different proposals in the literature, including ULAP protocol, is shown in Table 12.

1. **Computation Overhead** Classic cipher algorithms or hash functions are well beyond the capability of low-cost RFID tags, as explicitly stated in the Gen-2 RFID specification. Additionally, one of the main drawbacks of hash-based solutions is that the load (for tag identification) on the server side (reader and back-end database) is proportional to the number of tags. In our proposal, this problem has been completely solved by using a *SessionIDS* that allows a tag to be unequivocally identified and its data directly accessed in the database. The *SessionIDS* (*SessionIDs = f*(*IDS*, *challenge*)) is obtained by means of a very efficient function which only uses simple operations, allowing its implementation in low-cost tags. As we see in section 5.2.1, the reader sends a challenge (*hello* || *challenge*) and the tag answers a *SessionIDS*, thus preventing an exhaustive search in the back-end database.
2. **Storage Overhead** We assume that all components are L-bit sized, that the hash function and the RNG are h, h_k: $\{0, 1\} \rightarrow \{0, 1\}^{(1/2)L}$ and $r \in_U \{0, 1\}^L$. Our protocol is based on L-bit *session index-pseudonym* (*SessionIDS*), L-bit *index-pseudonym* (*IDS*)

and an associated key of length $4L$, which is used for mutual authentication between the reader and the tag. Moreover, the tag has to store a unique identification number (ID) of length L. Both the reader and the tag have to store this information, so they require a memory of $7L$ bits.

3. **Communication Overhead** The proposed protocol accomplishes mutual authentication between the tag (T) and the reader ($R + B$), requiring only four rounds. Many other protocols require at least one or two additional messages. In the Ohkubo [38] and Feldhofer [13] protocols, only two messages are exchanged, but a mutual authentication is not accomplished with subsequent security problems. Molnar's protocol can be thought to be more efficient as only three messages are used [36]. However, in this protocol tags should be equipped with a pseudo-random number generator and a keyed hash function, which is far beyond the capabilities of low-cost RFID tags.
Taking into account that low-cost tags are passive and that communication can only be initiated by a reader, four rounds may be considered a reasonable number for mutual authentication in RFID environments.

7. IMPLEMENTATION

It is a common assumption that between 50-100 tags at least should be authenticated per second [45]. As in [13], due to the low-power restrictions of RFID tags, the clock frequency must be set to 100 KHz. So a tag may use up 2000 clock cycles at the most to answer a reader. Because of these characteristics, it is not necessary to resort to a parallel implementation. As shown in Fig. 2, we have decided not to process all the message at the same time, but in m-bit blocks.

L is set to 96-bits, which is a length compatible with all the encoding schemes (GTIN, SSCC, GLN, GRAI, GIAI, GID) defined by the EPC [12]. The proposed architecture is, however, independent of the word length used. We have analyzed the features of four different word lengths (m = 8, 16, 32, 96). In Fig. 2 we can see a logic scheme of the proposed architecture and the logical memory map conforming to Gen-2 RFID specification.

To the left, the memory is shown filled with the *index-pseudonym* (*IDS*), the *session index-pseudonym* (*SessionIDS*), the key K ($K1 \,||\, K2 \,||\, K3 \,||\, K4$), and the ID. Memory access is controlled by a sequencer. Since messages consist of three or more components, we will need an m-bit register to store intermediate results (*register*_1). In the middle of the figure we have the Arithmetic Logic Unit (ALU). This unit will make the following m-bit word length operation: bitwise xor ($\oplus$), bitwise or ($\vee$), and sum mod 2^m ($+$). The ALU has two inputs, each one selected by means of a multiplexor. The first input is selected (signal c_3) between the values stored in the memory, and the internal state of the *SessionIDS* update function (register_2). By means of the signal c_4 the second input is selected, consisting of the bitstream or the value stored in the auxiliary register_1. The control signal c_2 will select the operation that will be used in the ALU.

The *SessionIDS* update function is implemented with the same simple operations as in mutual authentication, so this function will use the ALU. As the new *SessionIDS* is obtained after 32 rounds, a new auxiliary register (resister_2) is necessary to store the results of each round. The signal c_1 controls the initialization of this function (*SessionIDS* = *IDS*).

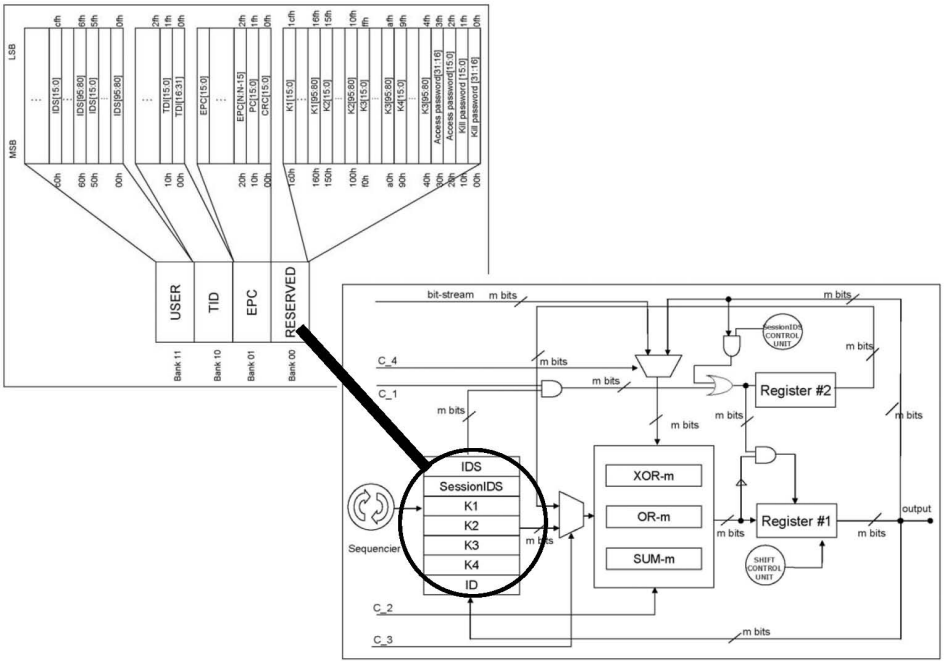


Fig. 2. Logic scheme and logical memory map.

Around 85% of all clock cycles needed to implement the four stages of the protocol are consumed by the *SessionIDS* update function. In spite of this, the total number of cycles employed is below the limit of 2000 cycles, considering that the clock frequency is set to 100KHz. Although in the case of $m = 8$ bits we are near the inferior limit (50 answers/sec), in the rest of cases the temporary requirements are accomplished with a relative high margin, for example in the 32-bit architecture a tag is able to provide 180 answers/sec. So the temporary requirements are fulfilled in all scenarios.

Another important aspect to consider is the number of logical gates necessary for implementing our protocol. The functions bitwise xor ($\oplus$), bitwise or ($\vee$) are implemented with m (the word length) logic gates. For implementing the add with carry circuit, a parallel architecture is proposed. Six logic gates are needed for each bit added in parallel.[2] A gate count of 8 has been chosen for implementing a flip-flop as in [17]. Additionally, 30% of logic gates are considered for control functions.

Table 13. Proposed architectures features.

Word length		8-bit	16-bit	32-bit	96-bit
N. of	ALU	64	128	256	7350
	Control	22	43	86	8400
	Total	86	171	342	8120
N. of clock cycles (@100KHz)		2194	1402	706	10868
Answers/second		46	71	142	8100

[2] Add one bit with carry: $S = A \oplus [B \oplus C_{ENT}]$, $C_{SAL} = BC_{ENT} + AC_{ENT} + AB$.

Table 14. Core comparison of hash functions.

	Hash output size (bits)	Cycles per block	Throughput at 100 KHz (Kbps)	Area GE
MD4	128	456	28	7350
MD5	128	612	20.9	8400
SHA-1	160	1274	12.55	8120
SHA-256	256	1128	22.7	10868
NAME	256	96	266.67	8100

As we can see in Table 13, in the best case (m = 8) the protocol only needs around 100 gates. In Table 14, we also show the number of logical gates needed for implementing various hash functions. The best implementation of traditional hash functions such as MD5 or SHA needs more than 8K gates which is far higher than the capabilities of low-cost RFID tags [14]. Additionally, there is also a proposal of an implementation of a new universal hash function for ultra low-power cryptographic hardware applications [58]. Although this solution only needs around 1.7K gates, a deeper security analysis of it is necessary and has not yet been accomplished. Furthermore, this function only has a 64-bit output, which does not guarantee an appropriate security level because finding collisions is a relatively easy task due to the birthday paradox (around 2^{32} operations).

Finally, although we have not yet implemented the circuit physically, due to the known fact that power consumption and circuit area are proportional to the number of logical gates, it seems that our implementation will be suitable even for very low-cost RFID tags.

8. CONCLUSIONS

Despite the common usage of hash functions in theoretical proposals to secure RFID systems, its implementation is well beyond the current capabilities of these devices. Moreover, the use of hash functions has not been included in the Gen-2 specification.

Since the publication and later ratification of the EPC Class-1 Generation-2 specification, its security has been deeply analyzed. After a detailed examination, important security pitfalls have been discovered. For example, the EPC is transmitted in plain text, which implies privacy and spoofing problems. Moreover, tracking could be done in a very straightforward way, since EPC is fixed. Some authors have proposed new solutions in order to solve some of these problems [6, 24, 27, 28, 37]. However, the protocols presented so far still present security weaknesses. For this reason we propose a new ultra light authentication protocol, named ULAP.

As it has been ratified by the Gen-2 specification, only simple operations are supported on-chip on the tag. Moreover, we consider 4K gates as the maximum number of gates that can be devoted to security [44]. Taking into account these two considerations, the following operations are used in the protocol: bitwise xor ($\oplus$), bitwise or ($\vee$), and sum mod 2^m (+). Tag memory has been extended to store the key, *index-pseudonym*, and *SessionIDS*. This use fits well with Gen-2 specification, where memory size is unlimited. The first two phases of our protocol (tag-identification and reader-authentication) will be

equivalent to the inventory tag operation described in the Gen-2 specification [11], but correct its security problems.

In spite of demanding very few resources, the main security aspects of RFID systems (privacy, tracking) have been addressed efficiently. As shown in Table 11, ULAP improves on many of the hash-based solutions proposed so far, in many aspects. So, ULAP is not only able to avoid privacy and tracking-related problems, but it is also resistant to forgery, replay attacks, *etc.*

Finally, another paramount characteristic in our scheme is efficiency: tag identification by a valid reader does not require exhaustive search in the back-end database, allowing for an easy scalability of our scheme to RFID systems with huge numbers of tags. Furthermore, only two messages need to be exchanged in the identification stage and another two in the mutual authentication stage, keeping the communication overhead to a minimum. The same number of messages are used in the inventory tag population described in the Gen-2 specification. So our protocol does not alter the framework of the EPC Class-1 Generation-2. In conclusion, ULAP is a fully EPC Gen-2 compliant protocol.

REFERENCES

1. The lil-gp Genetic Programming System, http://garage.cse.msu.edu/software/lil-gp/.
2. David Sexton's Battery, http://www.geocities.com/da5id65536, 2005.
3. G. Avoine, "Adversary model for radio frequency identification," Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPC), Security and Cryptography Laboratory (LASEC), 2005.
4. D. J. Bernstein, "Salsa20 specifications," http://www.ecrypt.eu.org/stream/.
5. S. Bono, M. Green, A. Stubblefield, A. Juels, A. Rubin, and M. Szydlo, "Security analysis of a cryptographically-enabled RFID device," in *Proceedings of the 14th Conference on USENIX Security Symposium*, 2005, pp. 1-16.
6. H. Y Chien and C. H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class-1 Generation-2 standards," *Computer Standards and Interfaces*, Vol. 29, 2007, pp. 254-259.
7. E. Y. Choi, S. M. Lee, and D. H. Lee, "Efficient RFID authentication protocol for ubiquitous computing environment," in *Proceedings of the 1st International Workshop on Security in Ubiquitous Computing Systems* (*SecUbiq*), LNCS 3823, 2005, pp. 945-954.
8. Y. Cui, K. Kobara, K. Matsuura, and H. Imai, "Lightweight asymmetric privacy-preserving authentication protocols secure against active attacks," in *Proceedings of the 5th IEEE International Conference on Pervasive Computing and Communications Workshops*, 2007, pp. 223-228.
9. R. Das, "RFID explained," IDTechEx, White paper, 2004.
10. T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1607559.
11. Class-1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9: Gen-2, http://www.epcglobalinc.org/standards/.

12. EPC Global, http://www.epcglobalinc.org/.
13. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems*, LNCS 3156, 2004, pp. 357-370.
14. M. Feldhofer and C. Rechberger, "A case against currently used hash functions in RFID protocols" in *Proceedings of Workshop on RFID Security*, 2006, pp. 109-122.
15. R. Forré, "The strict avalanche criterion: Spectral properties of Boolean functions and an extended definition," in *Proceedings of Advances in Cryptology – CRYPTO*, LNCS 403, 1990, pp. 450-468.
16. J. P. Hayes, *Introduction to Digital Logic Design*, Addison Wesley, New York, 1994.
17. M. Hell, T. Johansson, and W. Meier, "Grain – A stream cipher for constrained environments," in *Proceedings of Workshop on RFID and Lightweight Crypto*, 2005, http://events. iaik.tugraz.at/RFIDandLightweightCrypto05/.
18. D. Henrici and P. Müller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, pp. 149-153.
19. H. Y. Chien and C. W. Huang, "Security of ultra-lightweight RFID authentication protocols and its improvements," *ACM SIGOPS Operating Systems Review*, Vol. 41, 2007, pp. 83-86.
20. ISO – International Organization for Standardization, http://www.iso.org/.
21. J. Yang, "Security and privacy on authentication protocol for low-cost radio frequency identification," Master Thesis, School of Engineering, Information and Communications University, 2004.
22. "Frequently asked questions," *RFID Journal*, http://www.rfidjournal.com, 2006.
23. A. Juels, "Minimalist cryptography for low-cost RFID tags," in *Proceedings of the 4th International Conference on Security in Communication Networks*, LNCS 3352, 2004, pp. 149-164.
24. A. Juels, "Strengthening EPC tags against cloning," in *Proceedings of the 4th ACM Workshop on Wireless Security*, 2005, pp. 67-76.
25. A. Juels, D. Molnar, and D. Wagner, "Security and privacy issues in e-passports," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005, pp. 74-88.
26. N. Karten and H. Plotz, "Mifare little security, despite obscurity," http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.htm, 2007.
27. S. Karthikeyan and M. Nesterenko, "RFID security without extensive cryptography," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2005, http://deneb.cs.kent.edu/~mikhail/Research/RFID.pdf.
28. D. M. Konidala and K. Kim, "RFID tag-reader mutual authentication scheme utilizing tag's access password," White Paper wp-hardware-033, Auto-ID Labs, 2007.
29. J. R. Koza, "Evolving a computer program to generate random number using the genetic programming paradigm," in *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991, pp. 37-44.
30. M. Lehtonen, T. Staake, F. Michahelles, and E. Fleisch, "From identification to authentication – A review of RFID product authentication techniques," *Networked RFID*

Systems and Lightweight Cryptography, Springer Berlin Heidelberg, 2007, pp. 169-187.
31. T. Li and R. Deng, "Vulnerability analysis of EMAP-An efficient RFID mutual authentication protocol," in *Proceedings of the 2nd International Conference on Availability, Reliability and Security*, 2007, pp. 238-245.
32. T. Li and G. Wang, "Security analysis of two ultra-lightweight RFID authentication protocols," in *Proceedings of IFIP International Federation for Information Processing*, Vol. 232, 2007, pp. 109-120.
33. T. L. Lim and T. Li, "Addressing the weakness in a lightweight RFID tag-reader mutual authentication scheme," in *Proceedings of Global Telecommunications Conference*, 2007, pp. 59-63.
34. T. Lohmmann, M. Schneider, and C. Ruland, "Analysis of power constraints for cryptographic algorithms in mid-cost RFID tags," in *Proceedings of International Conference on Smart Card Research and Advanced Applications*, LNCS 3928, 2006, pp. 278-288.
35. G. Marsaglia and W. W. Tsang, "Some difficult-to-pass tests of randomness," *Journal of Statistical Software*, Vol. 7, 2002, pp. 37-51.
36. D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 2004, pp. 210-219.
37. D. N. Duc, J. Park, H. Lee, and K. Kim, "Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning," in *Proceedings of Symposium on Cryptography and Information Security*, 2006, pp. 97.
38. M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to 'privacy-friendly' tags," in *Proceedings of RFID Privacy Workshop*, 2003, pp. 624-654.
39. P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags," in *Proceedings of the 2nd Workshop on RFID Security*, 2006, http://events.iaik.tugraz.at/RFIDSec06/Program/papers/013%20-%20Lightweight%20Mutual%20Authentication.pdf.
40. P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, "M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags," in *Proceedings of International Conference on Ubiquitous Intelligence and Computing*, LNCS 4159, 2006, pp. 912-923.
41. P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, "Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard," in *Proceedings of International Conference on RFID Security*, 2007, http://fidsec07.etsit.uma.es/.
42. P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, "RFID specification revisited," *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems chapter*, Auerbach Publications, 2008, pp. 127-156.
43. 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 *Proceedings of OTM Federated Conferences and Workshop: IS Workshop*, LNCS 4277, 2006, pp. 352-361.

44. D. Ranasinghe, D. Engels, and P. Cole, "Low-cost RFID systems: confronting security and privacy," in *Proceedings of Auto-ID Labs Research Workshop*, 2004, http://www.autoidlabs.org/single-view/dir/article/6/80/page.html.
45. C. M. Roberts, "Radio frequency identification (RFID)," *Computers and Security*, Vol. 25, 2006, pp. 18-26.
46. S. E. Sarma, S. A. Weis, and D. W. Engels, "RFID systems and security and privacy implications," in *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, LNCS 2523, 2002, pp. 454-470.
47. W. Sean and L. Thomas, "Automatic identification and data collection technologies in the transportation industry: BarCode and RFID," Technical Report, 2001.
48. C. Suresh, J. Charanjit, J. R. Rao, and P. Rohatgi, "A cautionary note regarding evaluation of AES candidates on smart-cards," in *Proceedings of the 2nd Advanced Encryption Standard Candidate Conference*, 1999, pp. 133-147.
49. San Francisco Business Times, "Surging market for RFID security predicted," 2004, http://sanfrancisco.bizjournals.com/sanfrancisco/stories/2004/03/22/daily22.html.
50. I. Vajda and L. Buttyán, "Lightweight authentication protocols for low-cost RFID tags," in *Proceedings of Workshop on Security in Ubiquitous Computing*, 2003, http://www.hit.bme.hu/~buttyan/publications/VajdaB03suc.pdf.
51. J. Walker, "Randomness battery," http://www.fourmilab.ch/random/, 1998.
52. S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Proceedings of the 1st International Conference on Security in Pervasive Computing*, LNCS 2802, 2003, pp. 454-469.
53. S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Proceedings of the 1st International Conference on Security in Pervasive Computing*, LNCS 2802, 2004, pp. 201-212.
54. M. Weiser, "The computer for the 21st century," *Scientific American*, Vol. 265, 1991, pp. 94-104.
55. D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," in *Proceedings of the 2nd International Workshop on Fast Software Encryption*, LNCS 1008, 1994, pp. 363-366.
56. D. J. Wheeler and R. M. Needham, "TEA extensions," Technical Report, Computer Laboratory, University of Cambridge, 1997.
57. J. Yang, J. Park, H. Lee, K. Ren, and K. Kim, "Mutual authentication protocol for low-cost RFID," in *Proceedings of Workshop on RFID and Lightweight Crypto*, 2005, pp. 17-24
58. K. Yüksel, J. P. Kaps, and B. Sunar, "Universal hash functions for emerging ultra-low-power networks," in *Proceedings of the Communications Networks and Distributed Systems Modeling and Simulation Conference*, 2004, http://www.crypto.wpi.edu/Publications/Documents/YukselKapsCnds04.pdf.

Pedro Peris-Lopez is Assistant Professor at the Computer Science Department of Carlos III University of Madrid. He has a M.Sc. in Telecommunications Engineering. His research interests are in the field of protocols design, authentication, privacy, lightweight cryptography, *etc*. Nowadays, his research is focused on radio frequency identification systems (RFID). In these fields, he has published a great number of papers in specialized journals and conference proceedings.

Julio C. Hernandez-Castro is Associate Professor at the Computer Science Department of Carlos III University of Madrid. He has a B.Sc. in Mathematics, a M.Sc. in Coding Theory and Network Security, and a Ph.D. in Computer Science. His interests are mainly focused in cryptology, network security, steganography and evolutionary computation. He loves chess and dreams of becoming, one day, a professional chess player. He also loves Recreational Mathematics and has published some fun articles in Journals specialized in this area.

Juan M. Estevez-Tapiador is Associate Professor at the Computer Science Department of Carlos III University of Madrid. He holds a M.Sc. in Computer Science from the University of Granada (2000), where he obtained the Best Student Academic Award, and a Ph.D. in Computer Science (2004) from the same university. His research is focused on cryptology and information security. In these fields, he has published around 40 papers in specialized journals and conference proceedings. He is member of the program committee of several conferences related to information security and serves as regular referee for various journals.

Arturo Ribagorda is Full Professor at Carlos III University of Madrid, where he is also the Head of the Cryptography and Information Security Group and currently acts as the Director of the Computer Science Department. He has a M.Sc. in Telecommunications Engineering and a Ph.D. in Computer Science. He is one of the pioneers of computer security in Spain, having more than 25 years of research and development experience in this field. He has authored 4 books and more than 100 articles in several areas of information security. Additionally, he is member of the program committee of several conferences related to cryptography and information security.