Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol

Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M.E. Tapiador,
and Arturo Ribagorda

Computer Science Department, Carlos III University of Madrid
{pperis,jcesar,jestevez,arturo}@inf.uc3m.es
http://www.lightweightcryptography.com

Abstract. The design of ultralightweight authentication protocols that conform to low-cost tag requirements is imperative. This paper analyses the most important proposals (except for those based in hard problems such as the HB [1–3] family) in the area [4–6] and identifies the common weaknesses that have left all of them open to various attacks [7–11]. Finally, we present Gossamer, a new protocol inspired by the recently published SASI scheme [13], that was lately also the subject of a disclosure attack by Hernandez-Castro et al. [14]. Specifically, this new protocol is designed to avoid the problems of the past, and we examine in some deep its security and performance.

1 Introduction

In a RFID system, objects are labeled with a tag. Each tag contains a microchip with a certain amount of computational and storage capabilities, and a coupling element. Such devices can be classified according to memory type and power source. Another relevant parameter is tag price[1], which creates a broad distinction between high-cost and low-cost RFID tags.

Each time a new protocol is defined, the tag's class for which it is envisioned should also be specified. We note that, depending on the class of the tag, the maximum security level that can be supported will also be very different. For example, the security level of a relatively high-cost tag as those used in e-passports should be much higher than that of a low-cost tag employed in supply chain management (i.e. tags compliant to EPC Class-1 Generation-2 specification).

In [13], Chien proposed a tag classification mainly based on which were the operations supported on-chip. High-cost tags are divided into two classes: "full-fledged" and "simple". Full-fledged tags support on-board conventional cryptography like symmetric encryption, cryptographic one-way functions and even public key cryptography. Simple tags can support random number generators and one-way hash functions. Likewise, there are two classes for low-cost RFID tags."Lightweight" tags are those whose chip supports a random number generation and simple functions like a Cyclic Redundancy Code (CRC) checksum,

[1] The *rule of thumb* of gate cost says that every extra 1,000 gates increases chip price by 1 cent [15].

K.-I. Chung, K. Sohn, and M. Yung (Eds.): WISA 2008, LNCS 5379, pp. 56–68, 2009.
© Springer-Verlag Berlin Heidelberg 2009

but not cryptographic hash function. "Ultralightweight" tags can only compute simple bitwise operations like XOR, AND, OR, etc. These ultralightweight tags represent the greatest challenge in terms of security, due to their expected wide deployment and very limited capabilities.

2 A Family of Ultralightweight Mutual Authentication Protocols

In 2006, Peris et al. proposed a family of Ultralightweight Mutual Authentication Protocols (henceforth referred to as the UMAP family of protocols). Chronologically, M^2AP [4] was the first proposal, followed by EMAP [5] and LMAP [6].

These protocols are based on the use of pseudonyms to guarantee tag anonymity. Specifically, an index-pseudonym is used by an authorized reader to retrieve the information associated with a tag (tag identification phase). Additionally, a key -divided in several subkeys- is shared between legitimate tags and readers (back-end database). Both readers and tags use these subkeys to build the messages exchanged in the mutual authentication phase.

In line with their real processing capabilities, tags only support on-board simple operations. Indeed, these protocols are based on bitwise XOR ($\oplus$), bitwise OR ($\vee$), bitwise AND ($\wedge$) and addition mod 2^m. By contrast, only readers need to generate pseudorandom numbers; tags only use them for creating fresh messages to the protocol.

In the UMAP family of protocols, the proposed scheme consists of three stages. First, the tag is identified by means of the index-pseudonym. Secondly, the reader and the tag are mutually authenticated. This phase is also used to transmit the static tag identifier (ID) securely. Finally, the index-pseudonym and keys are updated (the reader is referred to the original papers for more details).

2.1 Security Analysis of the UMAP Protocols

Since the publication of the UMAP family of protocols, their security has been analyzed in depth by the research community. In [7, 8] a desynchronization attack and a full disclosure attack are presented. These require an active attacker and several incomplete run executions of the protocol to disclose the secret information on the tag. Later, Chien et al. proposed -based on the same attack model- a far more efficient full-disclosure attack [9]. Additionally, Bárász et al. showed how a passive attacker (an attack model that may be, in certain scenarios, much more realistic) can find out the static identifier and on particular secrets shared by the reader and the tag after eavesdropping on a few consecutive protocol rounds [10, 11].

This leads us to the following conclusions: first, we must define what kind of attack scenarios are applicable. In our opinion, ultralightweight RFID tags have to be resistant to passive attacks but not necessarily to active attacks, because of their severe restrictions (storage, circuitry and power consumption). Regarding passive attacks, we can affirm the following:

- The UMAP family of protocols is based on the composition of simple operations like bitwise AND, XOR, OR and sum mod 2^m. Because all of these are triangular functions (T-functions) [16], the information does not propagate well from left to right. In other words, the bit in position i in the output only depends on bits j = 0,..., i of the input words.
- The use of the bitwise AND or OR operations to build public submessages is a weakness common to all these protocols. When a bitwise AND (OR) operation is computed even over random inputs, the probability of obtaining a one (zero) is $\frac{3}{4}$. In other words, the result is strongly biased. This poor characteristic is the basis of all the passive attacks proposed so far.

3 SASI Protocol

In 2007 Hung-Yu Chien proposed a very interesting ultralightweight authentication protocol providing Strong Authentication and Strong Integrity (SASI) for very low-cost RFID tags [13]. We briefly describe the messages exchanged between the reader (or back-end database) and the tag.

An index-pseudonym (IDS), the tag's private identification (ID), and two keys (k_1/k_2) are stored both on the tag and in the back-end database. Simple bitwise XOR ($\oplus$), bitwise AND ($\wedge$), bitwise OR ($\vee$), addition 2^m and left rotation (Rot(x,y)) are required on the tag. Additionally, random number generation (i.e. n_1 and n_2) is required on the reader. The protocol is divided into three states: tag identification, mutual authentication and updating phase. In the identification phase, the reader (R) sends a "hello" message to the tag (T), and the tag answers with its IDS. The reader then finds, in the back-end database, the information associated with the tag (ID and k_1/k_2), and the protocol continues to the mutual authentication phase. In this, the reader and the tag authenticate each other, and the index-pseudonym and keys are subsequently updated:

R $\rightarrow$ **T** : $A||B||C$

The reader generates nonces n_1 and n_2 to build the submessages as follows: $A = IDS \oplus k_1 \oplus n_1$; $B = (IDS \vee k_2) + n_2$; $C = (k_1 \oplus k_2^*) + (k_2 \oplus k_1^*)$; where $k_1^* = Rot(k_1 \oplus n_2, k_1)$; $k_2^* = Rot(k_2 \oplus n_1, k_2)$

Tag. From messages A and B, the tag can obtain values n_1 and n_2 respectively. Then it locally computes C' and checks if the result is equal to the received value. If this is the case, it sends D and updates the values of IDS, k_1 and k_2: $D = (k_2^* + ID) \oplus ((k_1 \oplus k_2) \vee k_1^*)$; $IDS^{next} = (IDS + ID) \oplus (n_2 \oplus k_1^*)$; $k_1^{next} = k_1^*$; $k_2^{next} = k_2^*$;

T $\rightarrow$ **R** : D

Reader. Verifies D and, if it is equal to the result of its local computation, updates IDS, k_1 and k_2 in the same way as the tag.

3.1 Vulnerability Analysis

From the analysis of the UMAP family of protocols, we conclude that it is necessary to incorporate a non-triangular function in order to increase the security of ultralightweight protocols. At first sight, the SASI protocol complies with this requirement as it includes the left rotation operation (which is non triangular). However, Hernandez-Castro et al. have recently showed that the protocol was not carefully designed [14]. Indeed, a passive attacker can obtain the secret static identifier of the tag (ID) after observing several consecutive authentication sessions. We now summarize the main weaknesses of the protocol (see the original paper for more details):

1. The second component of the IDS updating equation is dependent on the bitwise XOR between n_2 and k_1^*. This gives rise to poor statistical properties as k_1^* is also function of n_2.
2. The key updating equation has a kind of distributive operation that might be employed to attack the protocol, for example: $k_1^* = Rot(k_1 \oplus n_2, k_1) = Rot(k_1, k_1) \oplus Rot(n_2, k_1)$
3. As mentioned in *Section 2.1*, bitwise OR and bitwise AND should be used with extreme care. These operations result in a strongly biased output. For example, the nonce n_2 can be approximated with very good precision by simply computing $n_2 \simeq B - 1$. These operations might therefore be only employed in the inner parts of the protocol but should be avoided in the generation of public submessages (i.e. B and D submessages). In fact, all the exchanged messages should resemble random values as far as possible.

4 Gossamer Protocol

As a consequence of the above observations, we have derived a new protocol, called Gossamer[2], which is inspired by the SASI scheme but hopefully devoid of its weaknesses. Our main aim was to define a protocol with adequate security level and which can realistically be employed in ultralightweight RFID tags.

4.1 Model Suppositions

Each tag stores a static identifier (ID), an index-pseudonym (IDS) and two keys (k_1/k_2) in its memory. This information is also stored in the back-end database. The IDS is employed as a search index to allocate, in the database, all the information linked with each tag. These elements have a length of 96 bits, compatible with all the encoding schemes (i.e. GTIN, GRAI) defined by EPCGlobal. Additionally, tags are given the added requirement of storing the old and potential new values of the tuple (IDS, k_1, k_2), to avoid desynchronization attacks. In spite of this, resiliency against attacks which involve tag manipulation are not considered as these devices are not at all tamper-resistant.

For the implementation of the proposed protocol, only simple operations are available on tags, in accordance with their restrictions: specifically, bitwise XOR

[2] Gossamer: Noun describing a thin film of cobwebs floating in the air (this meaning dates from the 14th century) and an adjective meaning light, delicate, thin enough to let light through, nearly transparent.

($\oplus$), addition mod 2^m ($+$), and left rotation (Rot(x,y)). Rotation may be performed in several different ways. However, the original SASI scheme does not clearly specify the rotation method used. Sun et al., who recently published two desynchronization attacks on SASI, contacted the author to clarify the issue [17]. Chien asserted that $Rot(x, y)$ is a circular shift of x, $wht(y)$ positions to the left where $wht(y)$ denotes the Hamming weight of y. This is probably not optimal from the security point of view as the argument that determines the number of positions rotated is far from uniform. Indeed, this variable follows the following probability distribution:

$$Prob(wht(B) = k) = \frac{\binom{96}{k}}{2^{96}} \tag{1}$$

In our proposed scheme $Rot(x, y)$ is defined perform a circular shift on the value of x, (y mod N) positions to the left for a given value of N (in our case 96).

Random number generation, required in the protocol to supply freshness, is a costly operation, so it is performed by the reader. Moreover, random numbers cannot be indiscriminately employed because their use increases both memory requirements and message counts (which could be costly in certain applications). To significantly increase security, we have also added a specially designed and very lightweight function called $MixBits$. In [18], a detailed description of the methodology used -basically, to evolve compositions of extremely light operands by means of genetic programming, in order to obtain highly non-linear functions- is included. $MixBits$ has an extremely lightweight nature, as only bitwise right shift (>>) and additions are employed. Specifically,

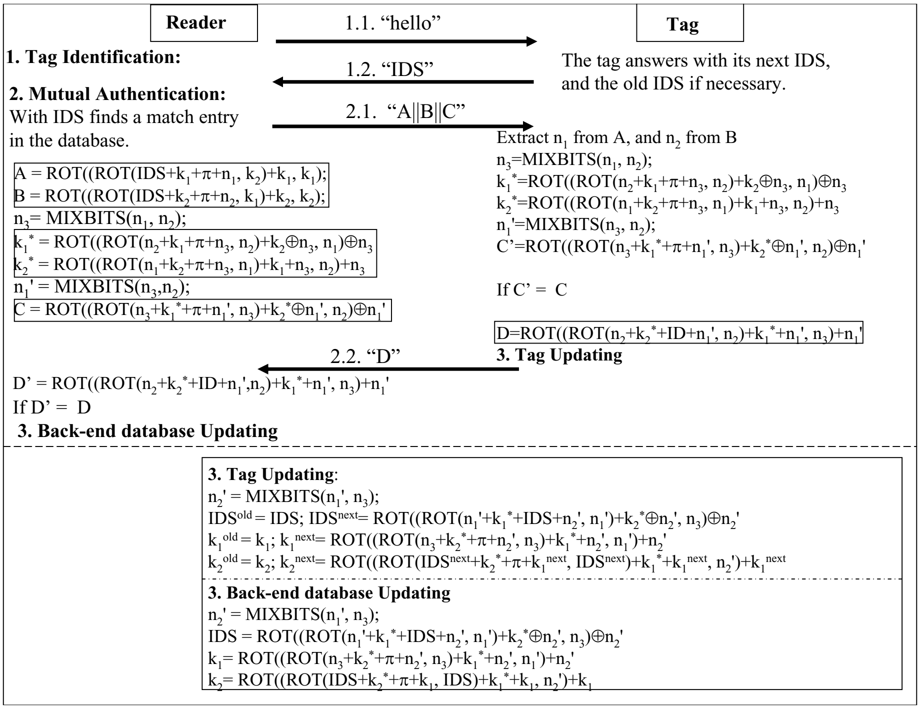
```
Z = MixBits(X,Y)
----------------------------
Z = X;
for(i=0; i<32; i++) {
Z = (Z>>1) + Z + Z + Y  ;}
----------------------------
```

Communication has to be initiated by readers, since tags are passive. The communication channel between the reader and the database is generally assumed to be secure, but the channel between the reader and the tag can be eavesdropped on. Attacks involving modification of the exchanged messages, the insertion of fraudulent new messages, or message blocking (active attacks), can be discounted.

4.2 The Protocol

The protocol comprises three stages: tag identification phase, mutual authentication phase, and updating phase. *Figure 1* shows the exchanged messages.

Tag Identification. The reader first sends a “hello” message to the tag, which answers with its potential next IDS. With it, the reader tries to find an identical entry in the database. If this search succeeds, the mutual authentication phase starts. Otherwise the identification is retried but with the old IDS, which is backscattered by the tag upon request.


Reader
Tag
1.1. “hello”
1. Tag Identification:
1.2. “IDS”
The tag answers with its next IDS, and the old IDS if necessary.
2. Mutual Authentication:
With IDS finds a match entry in the database.
2.1. “A||B||C”
$A = ROT((ROT(IDS+k_1+\pi+n_1, k_2)+k_1, k_1);$
$B = ROT((ROT(IDS+k_2+\pi+n_2, k_1)+k_2, k_2);$
$n_3= MIXBITS(n_1, n_2);$
$k_1^* = ROT((ROT(n_2+k_1+\pi+n_3, n_2)+k_2 \oplus n_3, n_1) \oplus n_3$
$k_2^* = ROT((ROT(n_1+k_2+\pi+n_3, n_1)+k_1+n_3, n_2)+n_3$
$n_1' = MIXBITS(n_3,n_2);$
$C = ROT((ROT(n_3+k_1^*+\pi+n_1', n_3)+k_2^* \oplus n_1', n_2) \oplus n_1'$
Extract n_1 from A, and n_2 from B
$n_3=MIXBITS(n_1, n_2);$
$k_1^*=ROT((ROT(n_2+k_1+\pi+n_3, n_2)+k_2 \oplus n_3, n_1) \oplus n_3$
$k_2^*=ROT((ROT(n_1+k_2+\pi+n_3, n_1)+k_1+n_3, n_2)+n_3$
$n_1'=MIXBITS(n_3, n_2);$
$C'=ROT((ROT(n_3+k_1^*+\pi+n_1', n_3)+k_2^* \oplus n_1', n_2) \oplus n_1'$
If C’ = C
$D=ROT((ROT(n_2+k_2^*+ID+n_1', n_2)+k_1^*+n_1', n_3)+n_1'$
3. Tag Updating
2.2. “D”
$D' = ROT((ROT(n_2+k_2^*+ID+n_1',n_2)+k_1^*+n_1', n_3)+n_1'$
If D’ = D
3. Back-end database Updating
3. Tag Updating:
$n_2' = MIXBITS(n_1', n_3);$
$IDS^{old} = IDS; IDS^{next}= ROT((ROT(n_1'+k_1^*+IDS+n_2', n_1')+k_2^* \oplus n_2', n_3) \oplus n_2'$
$k_1^{old} = k_1; k_1^{next}= ROT((ROT(n_3+k_2^*+\pi+n_2', n_3)+k_1^*+n_2', n_1')+n_2'$
$k_2^{old} = k_2; k_2^{next}= ROT((ROT(IDS^{next}+k_2^*+\pi+k_1^{next}, IDS^{next})+k_1^*+k_1^{next}, n_2')+k_1^{next}$
3. Back-end database Updating
$n_2' = MIXBITS(n_1', n_3);$
$IDS = ROT((ROT(n_1'+k_1^*+IDS+n_2', n_1')+k_2^* \oplus n_2', n_3) \oplus n_2'$
$k_1= ROT((ROT(n_3+k_2^*+\pi+n_2', n_3)+k_1^*+n_2', n_1')+n_2'$
$k_2= ROT((ROT(IDS+k_2^*+\pi+k_1, IDS)+k_1^*+k_1, n_2')+k_1$

† $\pi = 0x3243F6A8885A308D313198A2$ ($L = 96$ bits).

Fig. 1. Gossamer Protocol

Mutual Authentication. With *IDS*, the reader acquires the private information linked to the tag, identified from the database. Then the reader generates nonces n_1 and n_2 and builds and sends to the tag $A||B||C$ (see *Figure 1*). Note that the equations used in the generation of public messages, as do those used in the computation of internal values, generally follow the scheme below:

$$n_{i+2} = MIXBITS(n_i, n_{i+1}) \tag{2}$$

$$M_i = ROT((ROT(n_{i+1} + k_i + PI + n_{i+2}, n_{i+1}) + k_{i+1} \oplus n_{i+2}, n_i) \oplus n_{i+2} \tag{3}$$

$$M_{i+1} = ROT((ROT(n_i + k_{i+1} + PI + n_{i+2}, n_i) + k_i + n_{i+2}, n_{i+1}) + n_{i+2} \tag{4}$$

From submessages A and B, the tag extracts nonces n_1 and n_2. Then it computes n_3/n'_1 and k^*_1/k^*_2 and builds a local version of submessage C'. This is compared with the received value. If it is verified, the reader is authenticated. Finally, the tag sends message D to the reader. On receiving D, this value is compared with a computed local version. If comparison is successful, the tag is authenticated; otherwise the protocol is abandoned.

Index-Pseudonym and Key Updating. After successfully completing the mutual authentication phase between reader and tag, they locally update *IDS* and keys (k_1/k_2) as indicated in *Figure 1*. As we have just seen, submessages C/D allow reader/tag authentication, respectively. Moreover, the use of submessages C/D results in confirmation of synchronization for the internal secret values (n_3/n'_1 and k^*_1/k^*_2) used in the updating phase, preventing straightforward desynchronization attacks.

4.3 Security Analysis

We will now analyze the security of the proposed scheme against relevant attacks:

Data Confidentiality. All public messages are composed of at least three secret values shared only by legitimate readers and genuine tags. Note that we consider private information (ID, k_1, k_2), random numbers (n_1, n_2), and internal values (n_3, n'_1, n'_2, k^*_1, k^*_2) as secret values. The static identifier and the secret keys cannot, therefore, be easily obtained by an eavesdropper.

Tag anonymity. Each tag updates IDS and private keys (k_1, k_2) after successful authentication, and this update process involves random numbers (n_3, n'_1, n'_2). When the tag is interrogated again, a fresh IDS is backscattered. Additionally, all public submessages ($A||B||C||$ and D) are anonymized by the use of random numbers (n_1, n_2, n_3, n'_1). Tag anonymity is thus guaranteed, and location privacy of the tag owner is not compromised.

Mutual Authentication and Data Integrity. The protocol provides mutual authentication. Only a legitimate reader possessing keys (k_1, k_2), can build a valid message $A||B||C$. Similarly, only a genuine tag can derive nonces n_1, n_2 from $A||B||C$, and then compute message D.

Messages C and D, which involve the internal secret values (n_3, n'_1, k^*_1, k^*_2) and nonces (n_1, n_2), allow data integrity to be checked. Note that these values are included in the updating equations (potential next index-pseudonym and keys).

Replay attacks. An eavesdropper could store all the messages exchanged in a protocol run. To impersonate the tag, he could replay message D. However, this response would be invalid as different nonces are employed in each session -this will frustrate this naive attack. Additionally, the attacker could pretend that the reader has not accomplished the updating phase in the previous session. In this scenario, the tag is identified by the old index-pseudonym and the attacker may forward the eavesdropped values of $A||B||C$. Even if this is successful, no secret information is disclosed and the internal state is unchanged in the genuine tag, so all these attacks are unsuccessful.

Forward Security. Forward security is the property that guarantees the security of past communications even when a tag is compromised at a later stage. Imagine that a tag is exposed one day, making public its secret information (ID, k_1, k_2). The attacker still cannot infer any information from previous sessions as two unknown nonces (n_1, n_2) and five internal secret values (n_3, n'_1, n'_2, k^*_1, k^*_2) are involved in the message creation (mutual authentication phase). Additionally, these internal values are employed in the updating phase. Consequently, past communications cannot be easily jeopardized.

Updating Confirmation. The Gossamer protocol assumes that tags and readers share certain secret values. As these values are locally updated, synchronization is mandatory. Submessages C and D provide confirmation of the internal secret values (n_3, n_1', k_1^*, k_2^*) and nonces (n_1, n_2). These values are employed in the updating stage. So the correct update of values IDS and keys (k_1, k_2) is implicitly ensured by submessages C and D.

Unintentional transmission errors can happen in the received messages since a radio link is used. This is an extremely serious issue for message D, since it can result in a loss of synchronization. However, the tuple (IDS, k_1, k_2) is stored twice in the tag memory -once with the old values, the other with the potential next values. With this mechanism, even in the event that message D is incorrectly received, the tag and the reader can still authenticate with the old values. So the reader and the tag will be able to recover their synchronized state.

4.4 Performance Analysis

Our proposed protocol is now examined from the point of view of computational cost, storage requirements and communication cost. Additionally, *Table 1* compares the most relevant ultralightweight protocol proposals (see *Section 1*) from a performance perspective.

Table 1. Performance Comparison of Ultralightweight Authentication Protocols

	U-MAP family [4–6]	SASI [13]	Gossamer
Resistance to Desynchronization Attacks	No	No	Yes
Resistance to Disclosure Attacks	No	No	Yes
Privacy and Anonymity	Yes	Yes	Yes
Mutual Authentication and Forward Security	Yes	Yes	Yes
Total Messages for Mutual Authentication	4-5L	4L	4L
Memory Size on Tag	6L	7L	7L
Memory Size for each Tag on Database	6L	4L	4L
Operation Types on Tag	$\oplus$, $\vee$, $\wedge$, $+$	$\oplus$, $\vee$, $\wedge$, $+$, Rot[2]	$\oplus$, $+$, Rot[3], *MixBits*

[1] L designates the bit length of variables used.

[2] $Rot(x, y) = x << wht(y)$, being $wht(y)$ the Hamming weight of vector y.

[3] $Rot(x, y) = x << (y \bmod L)$ for a given value of L -in our case $L = 96$.

Computational cost. The protocol we have proposed only requires simple bitwise XOR, addition 2^m, left rotation, and the *MixBits* function on tags. These operations are very low-cost and can be efficiently implemented in hardware.

When comparing Gossamer with the protocol SASI, we can observe that the bitwise AND and OR operations are eliminated, and the light *MixBits* operation is added for increased security. *MixBits* is very efficient from a hardware perspective. The number of iterations of this function is optimized to guarantee a good diffusion effect. Specifically, it consumes $32 \times 4 \times (96/m)$ clock cycles, m being the word length used to implement the protocol (i.e. m = 8, 16, 32, 64, 96). As this may have a cost impact on the temporal requirements, we have minimized the number of *MixBits* calls.

Storage requirement. Each tag stores its static identifier (ID) and two records of the tuple (IDS, k_1, k_2) -with old and potential new values. A 96-bit length is assumed for all elements in accordance with EPCGlobal. The ID is a static value, thus stored in ROM. The remaining values ($96 \times 6 = 576$ bits) are stored in a rewritable memory because they need to be updated.

In the protocol SASI, two temporal nonces are linked to each session. We include an additional value derived from the previous nonces ($n_{i+2} = MixBits(n_i, n_{i+1})$). As these nonces are updated three times in the internal steps of the protocol, our scheme is roughly equivalent to the use of five fresh random numbers. So, with the relatively light penalty of storing an extra nonce, the security level seems to be notably increased.

Communication cost. The proposed protocol performs mutual authentication and integrity protection with only four messages, so in this sense it is similar to the SASI scheme. In the identification phase, a "hello" and IDS message are sent over the channel. Messages $A||B||C$ and D are transmitted in the authentication phase. So a total of 424 bits are sent over the channel - considering 5 bytes for the "hello" message.

5 Conclusions

We now present some conclusions: firstly those related with RFID security in general, then specifically related to the security of ultralightweight protocols.

5.1 General Considerations

Price and operability are the main issues whenever a new technology appears (i.e. bluetooth, wireless, etc.), security frequently being only a side consideration. To avoid past errors, however, the use of secure solutions should be generalized. Otherwise, the massive deployment of RFID technology runs the risk of being significantly delayed. Since 2003, it seems that the general awareness on the security issues of RFID systems (notably privacy) has been considerably increased, as reflected by a steady increment in the number of research publications on the field. However, the majority of proposals to secure RFID tags make the same two errors:

Tag Class. The tag's class for which the proposed protocol should be intended is not clearly specified in most of the proposals. However, the number of available resources (memory, circuitry, power consumption, etc.) hugely varies from one to another. In other words, not all tags will support the same operation set. For example, public cryptography is applicable for the most expensive RFID tags [19, 20], but it clearly exceeds the capabilities of low-cost RFID tags.

Additionally, the same security level cannot be asked to each RFID class. It is not sensible for a low-cost RFID tag (eg. a tagged biscuit packet) to have the same security level as that of an e-passport.

Tag Resources. Most of the proposed schemes are not realistic with respect to tag resources. Many lightweight cryptographic primitives have been recently proposed, and significant progress is being made in each research area. Clearly, there have been great improvements in the design of lightweight stream/block ciphers [21–25], but the design of lightweight hash functions [26, 12] and PRNGs [27] remains a pending task.

Hash functions are considered a better choice within the RFID security community regarding implementation. As a result, most of the proposed protocols are based on the use of hash functions, and some of these also include a PRNG. In spite of this, many authors claim that the proposed schemes are appropriate for low-cost RFID tags (lightweight and ultralightweight). However, standard hash functions demand more than 5.5K gates (130 nm) [28] - 8K gates (350 nm) [29], which is over the maximum number of gates (3K - 4K gates) that can be devoted to security functions in this tags. Note that the additional resources, needed to support on-chip the PRNG, would increase the total number of logic gates required.

Regarding standardization, there was previously a clear lack of harmonization, and major RFID vendors offered proprietary systems in the earlier implementations. Fortunately, things are changing rapidly. One of the most important standards is the EPCglobal Class-1 Generation-2 RFID specification (known as Gen-2 for short) [30, 31]. Gen-2 specification represents a significant advance for the widespread introduction of this technology, but its security level is extremely low (i.e. privacy is compromised as the EPC is transmitted in clear on the channel). Some authors intending to increase its security level proposed slight modifications in this specification [32–35]. Despite the fact that standards are being increasingly adopted by many companies, other developers base the security of their tags on proprietary solutions. However, the use of proprietary solutions is not altogether bad if algorithms are published so they can be scrutinized by the research community. As time has shown, the security of an algorithm cannot reside in its obscurity. Good examples of this are Texas Instruments DST tags [36] and Philips Mifare cards [37–39]. Companies should learn from past errors and make their proprietary algorithms public.

5.2 Ultralightweight Protocols

In 2003, Vajda et al. published the first article proposing the use of lightweight cryptography [42]. The following year, Juels introduced the concept of minimalist cryptography [43]. In 2005, there was no proposal in this area, the majority of proposals being based on the use non-lightweight hash functions. The year after, Peris et al. proposed the UMAP family of protocols. From the aforementioned protocols, we can infer the following considerations:

Interest. The protocols arouse interest in the design of new ultralightweight protocols. Indeed, they have inspired the proposal of other protocols [13, 40, 41]. Additionally, as can be seen below, the security of the UMAP family of protocols has been carefully examined by the research community.

Security Weaknesses. The security of the UMAP family of protocols has been analyzed under different assumptions. First, security vulnerabilities were revealed under the hypothesis of an active attacker [7–9]. Secondly, Bárász et al. showed how a passive attacker can disclose part of the secret information stored in the tag's memory [10, 11].

As mentioned in *Section 4.1*, only attacks that do not alter or interfere with communications are considered a real threat in most scenarios. In other words, active attacks are discounted when designing a protocol to meet the requirements of ultralightweight RFID tags.

Operations. Only bitwise AND, XOR, OR and sum mod 2^m are required for the implementation of the UMAP protocol family. At first sight, the choice seems well-conceived as these operations can be efficiently implemented in hardware. However, they are all T-functions, which have a very poor diffusion effect; the information does not propagate well from left to right [16]. Also, as a consequence of the use of bitwise AND and OR operations in the generation of certain messages, the latter were highly biased. These two operands should therefore be avoided in messages passed on the channel, but may be used in inner parts of the protocol.

The protocol SASI was a step further towards a secure protocol compliant with real ultralightweight tag requirements. However, it recently came under attack when Hernandez-Castro et al. showed how a passive attacker can obtain the secret ID by observing several consecutive authentications sessions. Despite this, we consider that the protocol design shows some interesting new ideas (specifically, the inclusion of rotations). The analysis of SASI and the UMAP protocol family has led to the proposal of Gossamer, a new protocol inspired by SASI and examined here both from the security and performance perspective. Indeed, the resources needed for the implementation of Gossamer are very similar to those of SASI the scheme, but Gossamer seems to be considerably more secure because of the use of dual rotation and the $MixBits$ function. The price to be paid, of course, is the throughput (number of authenticated tags per second) of the Gossamer protocol. However, preliminary estimations seem to show that the commonly required figure of 100 responses per second is still achievable.

References

1. Weis, S.: Security parallels between people and pervasive devices. In: Proc. of PERSEC 2005, pp. 105–109. IEEE Computer Society Press, Los Alamitos (2005)
2. Piramuthu, S.: HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In: Proc. of CollECTeR 2006 (2006)
3. Munilla, J., Peinado, A.: HB-MP: A further step in the HB-family of lightweight authentication protocols. Computer Networks 51(9), 2262–2267 (2007)
4. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 912–923. Springer, Heidelberg (2006)
5. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In: Hand. of Workshop on RFID and Lightweight Crypto (2006)

6. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 352–361. Springer, Heidelberg (2006)
7. Li, T., Deng, R.: Vulnerability analysis of EMAP - an efficient RFID mutual authentication protocol. In: Proc. of AReS 2007 (2007)
8. Li, T., Wang, G.: Security analysis of two ultra-lightweight RFID authentication protocols. In: Proc. of IFIP-SEC 2007 (2007)
9. Hung-Yu, C., Chen-Wei, H.: Security of ultra-lightweight RFID authentication protocols and its improvements. SIGOPS Oper. Syst. Rev. 41(4), 83–86 (2007)
10. Bárász, M., Boros, B., Ligeti, P., Lója, K., Nagy, D.: Breaking LMAP. In: Proc. of RFIDSec 2007 (2007)
11. Bárász, M., Boros, B., Ligeti, P., Lója, K., Nagy, D.: Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags. In: Proc. of First International EURASIP Workshop on RFID Technology (2007)
12. Shamir, A.: SQUASH - A New MAC With Provable Security Properties for Highly Constrained Devices Such as RFID Tags. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 144–157. Springer, Heidelberg (2008)
13. Chien, H.-Y.: SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. IEEE Transactions on Dependable and Secure Computing 4(4), 337–340 (2007)
14. Hernandez-Castro, J.C., Tapiador, J.M.E., Peris-Lopez, P., Quisquater, J.-J.: Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol. IEEE Transactions on Dependable and Secure Computing (submitted) (April 2008)
15. Weis, S.: Security and Privacy in Radio-Frequency Identification Devices. Master Thesis, MIT (2003)
16. Klimov, A., Shamir, A.: New Applications of T-functions in Block Ciphers and Hash Functions. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 18–31. Springer, Heidelberg (2005)
17. Sun, H.-M., Ting, W.-C., Wang, K.-H.: On the Security of Chien's Ultralightweight RFID Authentication Protocol. Cryptology ePrint Archive, http://eprint.iacr.org/2008/083
18. Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda-Garnacho, A., Ramos-Alvarez, B.: Wheedham: An automatically designed block cipher by means of genetic programming. In: Proc. of CEC 2006, pp. 192–199 (2006)
19. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., Verbauwhede, I.: Public-Key Cryptography for RFID-Tags. In: Proc. of PerCom 2007, pp. 217–222 (2007)
20. Kumar, S., Paar, C.: Are standards compliant elliptic curve cryptosystems feasible on RFID. In: Proc. of RFIDSec 2006 (2006)
21. Hell, M., Johansson, T., Meier, W.: Grain: a stream cipher for constrained environments, http://www.ecrypt.eu.org/stream/
22. Hell, M., Johansson, T., Meier, W.: A stream cipher proposal: Grain-128, http://www.ecrypt.eu.org/stream/
23. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
24. Feldhofer, M., Wolkerstorfer, J., Rijmen, V.: AES implementation on a grain of sand. In: Proc. on Information Security, vol. 152, pp. 13–20. IEEE Computer Society, Los Alamitos (2005)

25. Poschmann, A., Leander, G., Schramm, K., Paar, C.: New Light-Weight Crypto Algorithms for RFID. In: Proc. of ISCAS 2007, pp. 1843–1846 (2007)
26. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: An Efficient Authentication Protocol for RFID Systems Resistant to Active Attacks. In: Denko, M.K., Shih, C.-s., Li, K.-C., Tsao, S.-L., Zeng, Q.-A., Park, S.H., Ko, Y.-B., Hung, S.-H., Park, J.-H. (eds.) EUC-WS 2007. LNCS, vol. 4809, pp. 781–794. Springer, Heidelberg (2007)
27. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: LAMED – A PRNG for EPC Class-1 Generation-2 RFID specification. Journal of Computer Standards & Interfaces (2008), doi:10.1016/j.csi.2007.11.013
28. O'Neill, M. (McLoone): Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In: Hand. of Conference on RFID Security (2008)
29. Feldhofer, M., Rechberger, C.: A case against currently used hash functions in RFID protocols. In: Hand. of Workshop on RFID and Lightweight Crypto (2006)
30. Class-1 Generation-2 UHF air interface protocol standard version 1.0.9: "Gen-2" (2005), http://www.epcglobalinc.org/standards/
31. ISO/IEC 18000-6:2004/Amd:2006 (2006), http://www.iso.org/
32. Duc, D.N., Park, J., Lee, H., Kim, K.: Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. In: The 2006 Symposium on Cryptography and Information Security (2006)
33. Chien, H.Y., Chen, C.H.: Mutual authentication protocol for RFID conforming to EPC Class-1 Generation-2 standards. Computer Standards & Interfaces 29(2), 254–259 (2007)
34. Konidala, D.M., Kim, K.: RFID Tag-Reader Mutual Authentication Scheme Utilizing Tag's Access Password. Auto-ID Labs White Paper WP-HARDWARE-033 (January 2007)
35. Burmester, M., de Medewiros, B.: The Security of EPCGen2 Anonymous compliant RFID Protocols. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 490–506. Springer, Heidelberg (2008)
36. Bono, S., Green, M., Stubblefield, A., Juels, A., Rubin, A., Szydlo, M.: Security analysis of a cryptographically-enabled RFID device. In: Proc. of 14th USENIX Security Symposium, pp. 1–16 (2005)
37. Garcia, F.D., de Koning Gans, G., Muijrers, R., van Rossum, P., Verdult, R., Wichers Schreur, R.: Dismantling MIFARE Classic. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283. Springer, Heidelberg (2008)
38. de Koning Gans, G., Hoepman, J.-H., Garcia, F.D.: A Practical Attack on the MIFARE Classic. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 267–282. Springer, Heidelberg (2008)
39. Karten, N., Plotz, H.: Mifare little security, despite obscurity (2007), http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html
40. Li, T., Wang, G.: SLMAP-A Secure ultra-Lightweight RFID Mutual Authentication Protocol. In: Proc. of Chinacrypt 2007 (2007)
41. Lo, N.-W., Shie, H.-S., Yeh, K.-H.: A Design of RFID Mutual Authentication Protocol Using Lightweight Bitwise Operations. In: Proc. of JWIS 2008 (2008)
42. Vajda, I., Buttyán, L.: Lightweight authentication protocols for low-cost RFID tags. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864. Springer, Heidelberg (2003)
43. Juels, A.: Minimalist cryptography for low-cost RFID tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 149–164. Springer, Heidelberg (2005)