Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 231 – 254

doi:10.3906/elk-1212-95

Mersenne twister-based RFID authentication protocol

Mehmet Hilal ÖZCANHAN, Gökhan DALKILIÇ*
Department of Computer Engineering, Dokuz Eylül University, Tınaztepe, İzmir, Turkey

Received: 19.02.2012 • **Accepted:** 05.04.2013 • **Published Online:** 12.01.2015 • **Printed:** 09.02.2015

Abstract: This work presents an ultra-lightweight, cryptographic, mutual authentication protocol for radio-frequency identification (RFID) tags. The proposed scheme is more secure than its predecessors. The vulnerabilities of previous schemes based on triangular functions and rotation have been demonstrated in traditional and rotational cryptanalysis. In this paper, we summarize the successful attacks on previous schemes and demonstrate further attacks on 3 recent ultra-lightweight protocols. Next, we present a proposal that resists all of the known passive attacks. The proposal obeys the properties and rules of addition-rotation-XOR with constants systems. The proposed scheme inserts parts of advanced encryption standard S-boxes into the temperings of the Mersenne twister, which is used as a deterministic random number generator. The proposal is supported by extensive performance and security analysis. Apart from previous work, our scheme is compared and contrasted against the results of some popular hashing and encryption algorithms, recommended for RFID tags.

Key words: RFID, mutual authentication, ultra-lightweight protocols, ubiquitous computing, ARX-c systems

1. Introduction

RFID technology is gradually replacing commercial barcodes in automatic object identification applications [1]. IDTechEX's 2010 report [2] puts RFID in the category of booming ubiquitous technologies. The focus of this work is low-cost passive-UHF tags, which are reportedly the most popular. The research community has also shown great interest in passive tags, whose popularity is due to their low cost [3,4]. Only these tags can compete with the prices of plastic/paper barcodes used in huge numbers in supply chains [4].

RFID rests upon wireless technology. A tiny tag consists of a primitive microcontroller and memory that are energized via an antenna by a reader. The tag carries the unique and sensitive identity number (ID), which uniquely identifies the item it is tagged on [4]. The reader energizes the tag to access the ID inside. The low cost limits the computational capacity, available memory, and power supply of the tags. Only 4K gates can be reserved for security due to limited resources [3]. The resulting weak security is reflected in many reports of serious attacks and counter efforts [4–6]. In the proposed protocols, where the reader interrogates the tag, parties exchange messages [4]. Each side decrypts the received messages and tries to mutually authenticate the other. Preshared secrets are used in the formation of the messages. To hide its real identity, a tag uses an index-pseudonym (IDS), updated at the end of every run [3]. The reader sends the IDS to a securely wired database server to obtain the full record of the tag.

The ISO-18000-6 (ISO) [7] and the EPCglobal Class-1 Generation-2 (Gen-2) standard [8] set the passive

*Correspondence: dalkilic@cs.deu.edu.tr

tag properties. Unfortunately, they do not cover the security and privacy issues [4,9], causing diverse proposals to emerge. A classification of the proposals is made in one of the proposals, based on the processing power and operations supported [5]. The widely accepted classification divides the proposed protocols into 3. The *fully-fledged* protocol class supports random number generation, hashing, and encryption functions, as in e-passport tags. The *simple* protocol class supports hashing and random number generation but not encryption. The *ultra-lightweight* class supports random number generation, cyclic redundancy check (CRC), and simple bitwise operations, as in Gen-2 tags. Accordingly, the prefix 'ultra-lightweight' in the front of the tags, authentication protocols, cryptography, and functions means only simple, bitwise operations like AND, OR, exclusive OR, rotation, shifting, and the modulo 2^m addition are supported [3,5,9]. Protecting the sensitive ID number in the exchanged messages is more difficult in ultra-lightweight tags than in high-cost tags, because the resources (computation power, memory, supported operations) of low-cost tags are very limited [5,9]. The dilemma of protecting the secrets with scarce resources creates one of the biggest challenges in RFID security, which draws the attention of many researchers, as well as ours.

In this work, we focus on removing the previous repeated weaknesses of ultra-lightweight protocols and provide efficient security for the secrets they carry, by:

- Pinpointing the repeated weaknesses,
- Demonstrating new attacks on the 3 latest schemes,
- Applying the rules of addition-rotation–XOR with constants (ARX-c) systems to RFID operations,
- Complementing operations by inserting parts of advanced encryption standard (AES) S-box for randomness, and
- Inspiring from a well-established random number generation algorithm.

In the rest of this paper, Section 2 gives a classification of previous ultra-lightweight proposals, details their major weaknesses, and demonstrates the cryptanalysis of the 3 latest protocols, due to the exposed weaknesses. In Section 3, a detailed alternative proposal is presented. Section 4 contains the explanation and the results of the statistical tests. The security and performance analysis of the proposed new features are in Section 5. In Section 6, we conclude and list future work.

2. Related works and repeated weaknesses

There has been a lot of research in the security of ultra-lightweight RFID protocols. Ultra-lightweight protocols can also be subcategorized on the basis of utilized functions. Table 1 summarizes some of the protocols that use a combination of the ultra-lightweight functions AND, OR, XOR, addition modulo 2^m, rotation, and those that qualify to be in the ARX-c systems category. The protocols in Table 1 are detailed in the next sections, but it is necessary to mention the works outside of Table 1, which are also in the ultra-lightweight category. These other works use linear feed-back shift registers (LFSRs), CRC functions, physically unclonable functions (PUFs), and pseudo-random number generators (PRNGs).

Although the hardware implementation of LFSRs is suitable for Gen-2 tags, many attacks like correlation, fast correlation, distinguishing, random fault, and strategic attacks have been announced on LFSRs, some even before RFID security took off. Therefore, protocols based on LFSRs are classified as outside the scope of this work. Gen-2 tags support CRC functions and because they exist, some proposals use CRC with XOR operation. However, CRC is a homomorphism function and CRC(a $\oplus$ b) = CRC(a) $\oplus$ CRC(b). This property leads to a

fundamental algebraic weakness, taking the attention off of CRC-based protocols [10]. Protocols based on PUFs have also been claimed as ultra-lightweight. Unfortunately, PUFs rely on uncontrollable physical properties that do not convince the community as a strong security primitive [11,12]. Gen-2 tags also support a 16-bit PRNG, which is used to encrypt the ID in some proposals [13]. To the best of our knowledge, there is no formal proof for using PRNGs as an encryption algorithm; therefore, they will not be considered here either.

Table 1. Classification of the protocols according to the functions used.

Function protocol used	And	Or	Addition 2^m	Rotation	XOR	ARX
UMAP [16–18]	√	√	√	×	√	×
SASI [5]	×	√	√	√	√	√
UMA [41]	√	√	×	√	√	×
DIDT [40]	√	√	×	√	√	×
ULAP [3]	×	√	√	×	√	×
Gossamer [9]	×	×	√	√	√	√
ULERAP	×	×	√	√	√	√

Hashing and encryption are universally accepted algorithms that provide strong security protocol designs, but they are considered as outside of the ultra-lightweight category [3,5,9]. However, they are compared to the results of our work in Section 5, to show that our design uses much less die area and fewer clock cycles.

2.1. Related works in the ultra-lightweight category

The first proposals suggesting the idea of using lightweight cryptography in RFID came from Vajda and Buttyán [14] and Juels [15], before Gen-2 was released. The first ultra-lightweight protocols, the minimalist mutual-authentication protocol, lightweight mutual authentication protocol (LMAP), and efficient mutual authentication protocol, announced by Peris et al. [16–18], were named collectively as the ultra-lightweight mutual authentication protocol (UMAP) for short, and utilized only triangular functions ($\vee$, $\wedge$, $\oplus$) and addition. The UMAP protocols are shown in Table 1 because they have drawn a lot of attention [19–24]. Afterwards, Chien et al. pointed out the numerous attacks on the UMAP and suggested the inclusion of the rotation operation in their ultra-lightweight protocol, strong authentication and strong integrity (SASI) [5]. Unfortunately, the protocol used triangular functions ($\vee$, $\oplus$), addition, and only a limited number of rotations. SASI was quickly attacked in a number of works [25–29]. However, SASI inspired Peris et al. to announce Gossamer [9], which used addition and XOR encapsulated in nested rotation functions, evolving it into an ARX system [30]. Therefore, Gossamer is an 'on the up and up' protocol that is a result of the RFID authentication protocol evolution. No disclosure attack on Gossamer has been declared to date, except in very low probability special case situations. This is because ARX systems can be considered as the building blocks of contemporary encryption algorithms, which use repeated combinations of shifting, substitution, and XOR operations. Algorithms that do not use any of the ARX functions are considered weak [30].

After Gossamer, Pedro released the ultra-lightweight authentication protocol (ULAP) [3]. The ULAP uses only triangular functions and addition, a slightly improved version of the LMAP [17]. The ULAP contradicted the author's own guidelines in [31], where another protocol was attacked, declaring ".... main reason for the weaknesses ... be the noninclusion of any kind of rotations". It is therefore not surprising to see attacks on the ULAP in [32] and later in our work. Analyses of the UMAP, ULAP, SASI, and Gossamer are still coming, and the latest works on RFID authentication research can be found in [33].

2.2. Weaknesses of previous works

We divide previous ultra-light proposals into rotational and nonrotational schemes. The first section summarizes the weaknesses of nonrotational schemes and the second gives an account of rotational schemes weaknesses.

2.2.1. Weaknesses of triangular function-based protocols

The weaknesses of protocols based on triangular functions are outlined in [9,22–24]. Barasz et al. pinpointed the weakness of least significant bits (LSBs) in triangular bitwise operations and the misusage of OR and AND functions in some messages, which lead to exposure of the ID [22,23]. The authors in [20] launched a full disclosure attack by exploiting the algebraic weakness in the messages. By exposing the value of some terms in messages A and B, and substituting them in message D, the critical ID value was captured. Following the attack, the shared secrets were also exposed, by hopping from one message to another. In their security analyses, Lopez et al. [9] explained the weak diffusion properties of the OR and AND operations, listed the attacks made on the UMAP [19–24], and concluded that due to its capacity, ultra-lightweight tags cannot resist active attacks. In addition, a new protocol was proposed to resist the known passive attacks.

The authors in [10], defined 3 attacks and the RFID protocols where each is effective. Their work showed the success of algebraic attacks on protocols based on associative operations, such as triangular functions and modular addition. This analysis put all triangular-only protocols into jeopardy. Even so, the ULAP [3] tried to resist passive attacks by a new notion of 'sessionIDS', which is also a repetitive addition function, but failed. A passive attack against the ULAP was demonstrated in [32] and later in Section 2.4. It is obvious that changing or increasing the number of triangular functions or additions cannot save the UMAP, and the ULAP-type of protocols. These protocols do not contain rotation or shifting; therefore, they have been superseded by newer ARX schemes.

2.2.2. Weaknesses of rotation function based protocols

Protocols that encapsulate triangular functions (mainly XOR) and addition modulo 2^m in rotation operations are rotational schemes. These protocols are considered to be also in the ultra-lightweight category [5,9], because rotation is a simple shift operation with the most or least significant bits wrapped around it. The first ultra-lightweight protocol using rotation was SASI [5]. As the most pioneering work, SASI was analyzed in every detail [25–29]. This time, the most significant bits (MSBs) of the XORed messages are shown to be vulnerable to intentional bit flipping. Messages and updates with no rotation operations are the second point of weakness. Finally, the OR operation in one of the messages devastates the protocol. Using the above weaknesses, the authors in [28] outlined a detailed disclosure attack on SASI. In [9], the weaknesses of SASI were summarized and a new protocol, Gossamer, was suggested, which has stronger security. Gossamer is a scheme where 2 nested rotation operations encapsulate the addition and XOR operations, as shown in Figure 1. Gossamer also introduces a function called Mixbits, to provide randomness for the exchanged messages. The success of the rotation operation and the Mixbits function has drawn a lot of attention and they are still studied in the latest literature. In 2012, Bassil et al. praised Gossamer, but offered the PUF-based ultra-lightweight mutual-authentication RFID protocol (PUMAP), which replaces Mixbits with a PUF [34]. Unfortunately, the PUMAP and many other protocols that contain triangular functions and no rotation are immediately attacked [35,36].

After a first round of evaluations, some weaknesses were detected in Gossamer [6,37,38]. The first weakness appears in the updating of the IDS and the shared secrets. The lack of update-acknowledging and IDS collision result in desynchronization attacks [6]. A more serious weakness that exposes the ID value occurs

in 2 special cases when either the values of the shared secrets or the nonces equal 0 [38]. The messages exchanged reduce to a combination of addition and XOR operations, and after a few algebraic manipulations, the secret ID is disclosed. A similar approach also succeeds on the SASI protocol [38]. The most devastating attack is the disclosure attack, where the tag ID and the shared secrets are exposed with relatively little effort. This opens the way to creating a cloned tag without getting detected, which makes the attacked protocols very dangerous to use. This is the reason behind the attacks we devise in the following sections.

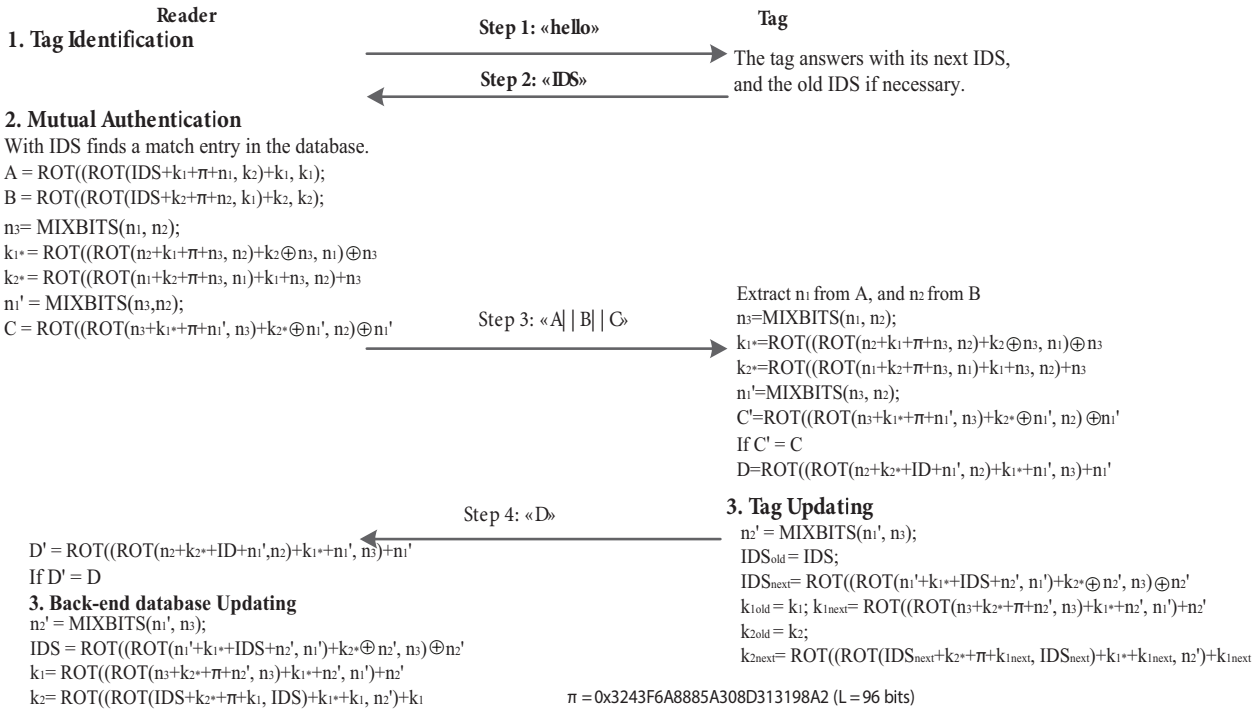

Figure 1. Gossamer protocol of [9].

Khovratovich and Nikolic studied ARX systems like hashing, SASI, and Gossamer in [30]. Their work declared that protocols containing rotation operations should be studied under rotational cryptanalysis. In [30], the authors showed that a combination of addition, rotation, and XOR was more secure, if a sufficient number of rotations and constants were used in the schemes. Our attacks below succeed because the above 2 requirements are not obeyed.

2.2.3. Cryptanalysis of a scheme using addition, rotation, xor operations

Our first focus is on Gossamer, shown in Figure 1. The scheme is based on a regularly updated IDS and the double rotation of shared secrets, obscured by random numbers. Two nonces are generated by the reader. First, (n_1) is passed in message A and the second (n_2) is passed in message B. Message C is the authenticator. The tag is expected to extract the nonces, calculate the extra terms n_3, k_1^*, k_2^*, $n_{1'}$, and C', and match C' to the authenticator C. A correct match authenticates the reader. The tag prepares and sends its authenticator D. Next, it immediately updates its secrets and IDS, without waiting for an acknowledgement from the reader. The reader authenticates the tag by matching the calculated D' and the D received. Finally, the reader also updates its values.

The scheme is an ARX-c system with the constant π, but message A omits the XOR operation and does not take the special case of ineffective rotation. The result of the rotation of all ones or all zeros gives the same result. Using this fact, if the addition IDS + $k_1 + \pi$ + n_1 in message A gives all bits as zeros (all zeros) or all bits as ones (all ones), the first rotation ROT(IDS + $k_1 + \pi$ + n_1, k_2) also yields all zeroes or all ones. The addition after the ineffective rotation, gives k_1 or $k_1 - 1$. For example 00_H (FF_H) + 03_H, yields 03_H (02_H). A thus reduces to A = ROT(k_1, k_1) or A = ROT(k_1- 1, k_1). Now, consider another case when IDS + $k_1 + \pi$ + n_1 is not all zeros, but all zeros except one bit. When rotated by an amount k_2 mod 96 as in message A, it will still be a value of all zeros, except that one of the bits is one. When this value is added to k_1, the result is still k_1, except at the position where the bit value is one. The binary addition of one flips the value of the added bit and may cause a carry. For simplicity, assuming no carry is generated and naming this flipped bit as k_1^f, it is obvious that there are many possibilities when message A collapses to a simple equation.

For rotations where no ones are wrapped around it, the operation is a left shift (multiplication by 2 for every shift). Hence, A = $2^{k1}\times$ k_1, or A = $2^{k1}\times$ $k_1 - 1$, or A = $2^{k1}\times$ k_1^f. A is public; therefore, possible k_1 values can be calculated easily. Next, n_1 is also revealed as IDS, k_1, π are known. Finally, there are 3 candidate sets of $\{k_1, n_1\}$ that can be true. For value k_1^f, k_1 is found by resetting the ones, one-by-one, and calculating as if calculating A = $2^{k1}\times$ k_1.

Without repeating the same argument for the similar message B, we can conclude that we can also have 3 candidate sets of $\{k_2, n_2\}$. An attacker can eavesdrop on messages A and B, and obtain values for k_1, k_2, n_1, and n_2, as if the assumptions are true. When calculating k_1^*, k_2^*, n_1', n_3, and C' in order, if C' matches C, then it is a successful assumption. From here on, the disclosure attacks shown in [19,27] can be launched. Only a number of the iterations given in [22] are needed. Since all of the other intermediate values, k_1^*, k_2^*, n_1', and n_3, depend on k_1, k_2, n_1, and n_2, the adversary calculates and inserts them into message D to capture the ID value. Our attack works on the versions in [6,37,38] as well.

The total probability of the success of the attacks is important. In [38], both k_1, k_2 or both n_1, n_2 were assumed to be zeros. Another extreme possibility is a case when all of the k_1, k_2, n_1, and n_2 are zeros. The probability of an attack where both k_1 and k_2, or both n_1 and n_2 are zeros is $2 \times (1 / 2^n) \times (1 / 2^n) = (2 / 2^{2n})$. The probability of all of them being zeros is $(1 / 2^{4n})$. For the attack in our work, the probability of all of the bits of addition in A or B being ones or zeros is $(1 / 2^n) + (1 / 2^n) = (2 / 2^n)$. In our extended attack, the probability of only one of the bits of addition in A or B being one is $(n/2^n)$. The overall probability of our assumptions is $[((2 / 2^n) + (n / 2^n)) \times ((2 / 2^n) + (n / 2^n))] = [(n + 2)^2/2^{2n}]$. Hence, the probability of attacking Gossamer is now increased to a total of $[(1 / 2^{4n}) + (2 / 2^{2n})] + [((n + 2)^2 / 2^{2n})]$. Neglecting the first term, the probability of a successful attack is $[(2 + (n + 2)^2) / 2^{2n}]$. The probability would increase if attacks for 2 bit flips in k_1 and k_2 are followed, but the point is made and we stop here.

The probability graph is dependent on the word length n. The probability of an EPCglobal Gen-2 tag is relatively high, because n = 16. It is also important to note that the probability of the attacks is equal at every reading and adds up with every reading. In theory, 1000 tags/s can be read [39] by a modern reader. By substituting n = 16, and multiplying by 1000 tags/s, the probability becomes $[((2 + 18^2)\times 1000) / 2^{32}]$ in 1 s. In 1 h, the probability of finding a tag that satisfies the assumptions, and thus the probability of capturing its ID, is 0.27.

2.3. Cryptanalysis of a latest scheme using only triangular functions

The second protocol of interest is the ULAP [3]. Having observed the attacks in [17], the authors tried to stop the escalation of known attacks on the IDS with a weak encryption algorithm called SessionIDS. The ULAP is similar to that in [17] and if the IDS is revealed the same passive attacks in [32] expose the sensitive ID number. Therefore, we will just show how the IDS is exposed. Our attack is inspired by the attacks in [22,24]. SessionIDS is given by the following function:

$$\text{for } (i=0; i<32; i++)\{Z=(Z>>1)+Z+Z+N;\},$$

where IDS = Z and N is a nonce. Denoting the bit position values with a superscript, after the first round the addition becomes:

$$\begin{array}{l} 0 \quad Z^{95} \,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\, Z^{3}\,Z^{2}\,Z^{1} \\ Z^{95}\,Z^{94} \,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\, Z^{2}\,Z^{1}\,Z^{0} \\ Z^{95}\,Z^{94} \,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\, Z^{2}\,Z^{1}\,Z^{0} \\ \underline{+\; N^{95}\,N^{94} \,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\, N^{2}\,N^{1}\,N^{0}} \end{array}$$

The result of the addition of the LSBs is revealed as $Z^1 + N^0$, because $Z^0 + Z^0$ is always 0 (only carry_0 is affected) and N^0 is public. Assuming a zero value for Z^0, carry_0 is calculated. Next, calculating every bit for $0 < n < 95$: $Z^n = Z^{n+1} + Z^n + N_0^n + \text{carry}_n$. Continuing until the addition of the MSBs, $0 + Z^{95} + Z^{95} + N^{95} + \text{carry}_{94}$ ($Z^{95} + Z^{95}$ results in 0, carry_{95} is unused). Since N^{95} is known, carry_{94} is revealed. Going through the 'for' loop leads to an equation for SessionIDS and exposes Z (IDS). If the obtained binary equations are not satisfied, predict $Z^0 = 1$ and repeat the bit calculations.

2.4. Cryptanalysis of the latest scheme using only XOR and rotation

The protocol DIDT [40] uses rotation and XOR operations but leaves out addition (Figure 2). The proposal is an update of the same author's work in [41]. The main design is the same, only the terms of the rotations are diversified to counter the attacks demonstrated in [42]. By updating the steps of the profound attack in [42], we were able to launch the same full disclosure attack on the new scheme. However, instead of repeating the same attacks, we demonstrate the effectiveness of an 'all ones' and 'all zeros' attack. An all zeros initial message A_i dissolves the scheme and resets DIDT_{i+1}, as well as the shared secret K_{i+1} to 0. All zeros A_i means $R_i = K_i$, due to a simple XOR. Hence, the XOR in B_i also yields all zeros. Similarly, C_i, DIDT_{i+1} and K_{i+1} all collapse to zeros.

The argument is the same for all ones in A_i, except this time $R_i = \text{NOT}(K_i)$. Now, B_i, C_i, DIDT_{i+1}, and K_{i+1} all collapse to all ones. All ones or all zeros can be forced on the tag by simply sending 0, 0 to the tag, in step 2. Next, DIDT_{i+1}, K_{i+1} are reset. For values where only 1 bit of R_i is different than K_i, the attack can also be successfully approximated, resulting in updated values with only a single bit set or reset. One has to wait passively for A_i values very close to all zeros or all ones. The probability of A_i values being all zeros or all ones, or only a 1-bit zero or one is $1 / 2^n + 1 / 2^n + n / 2^n + n / 2^n = (2n + 2) / 2^n$. For a 16-bit Gen 2 tag, t is $34 / 2^{16}$. After 964 readings, the probability of an expected A_i is 0.50. A slow reader reads around 600 tags/s [39]. In a supply chain, within 2 s, a tag can be found with a suitable A_i. Notice that

blocking message C_i twice causes desynchronization of the reader with the tag. The rest of the known attacks are in [42].

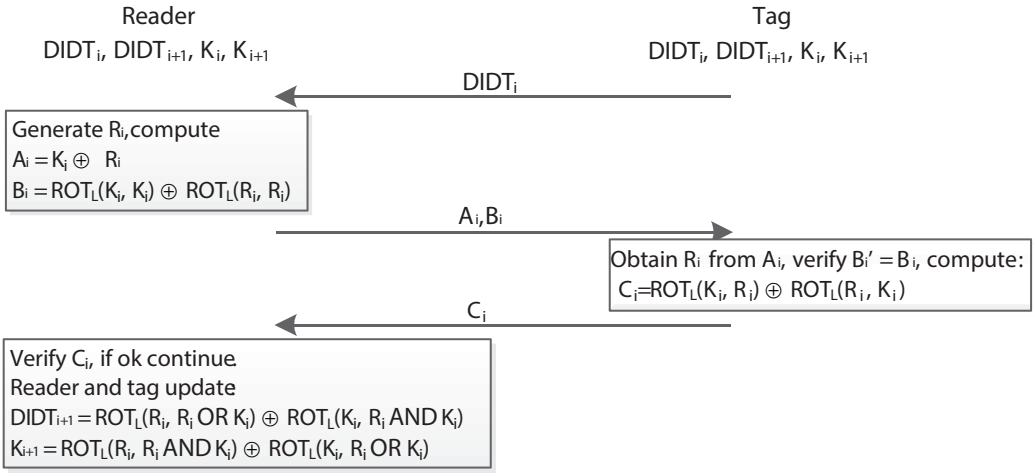

Figure 2. Protocol of [40].

Following the above examples, a list can be drawn to recapitulate the repeated weaknesses in previous ultra-lightweight protocols:

1. The designed functions in the UMAP- and ULAP-type protocols are not strong cryptologic functions, but consist of compositions of weak triangular functions and need to be combined with rotation operation, as outlined in [20,29,31],
2. XOR, mod 2^m addition, and triangular functions are associative operations that are nonresistant to algebraic replay attacks [10,20]; therefore, the messages of a protocol should not be reduced to these operations,
3. The operations follow a weak chain towards the main secret (ID), where intermediary messages are decrypted to capture the terms of the message that contains the ID [20,22,23],
4. The terms of the rotations are not carefully designed for special cases when some of the terms are all zeros or ones [6,38], leaving the used secrets vulnerable to analysis,
5. The rotational cryptanalysis outlined in [30] is not considered; for example, protocols [40,41], and
6. Secret update is neither signaled nor acknowledged [20].

3. The proposed ultra-lightweight extended authentication protocol

Taking the above weaknesses into consideration, we design the scheme in Figure 3. The scheme is made up of 3 phases. As in previous works, random numbers are used to provide freshness in the authentication messages, a measure against replay and other attacks. The resourceful reader generates a nonce n_1, which is used as a seed to generate other terms (n_2, n_3) for messages B, C, D, and IDS. The first improvement is the use of the Mersenne twister (MT) [43] to obtain the deterministic but unbiased terms n_2 and n_3. The messages and IDS must resemble random numbers to make tracing difficult [3,9]. In addition, message C should have good statistical properties, as it carries the secret ID. The MT is a well-proven random number generation algorithm that can serve the above requirements, because it passes the Diehard tests [43,44]. However, the first version

of the MT [43] had the disadvantages of poor initialization (outputs close to 0 for at least 1000 rounds) and unfitting size into a low-cost tag. Therefore, we used the improved version in [44], which does not fit into a tag either. Two changes have been made, which makes the scheme fit into a tag and removes the poor initialization. The matrix operations have been eliminated and temperings have been enforced with substitution of parts of the AES S-box [45] into the generation of n_2 and n_3. We call our version the enforced Mersenne twister (eMT) function, shown in Figure 4. The 96-bit word length and additional temperings allow the eMT to have a period of 259,872, longer than that of its inspired MT version. We also named our protocol the ULERAP, short for ultra-lightweight extended RFID authentication protocol.

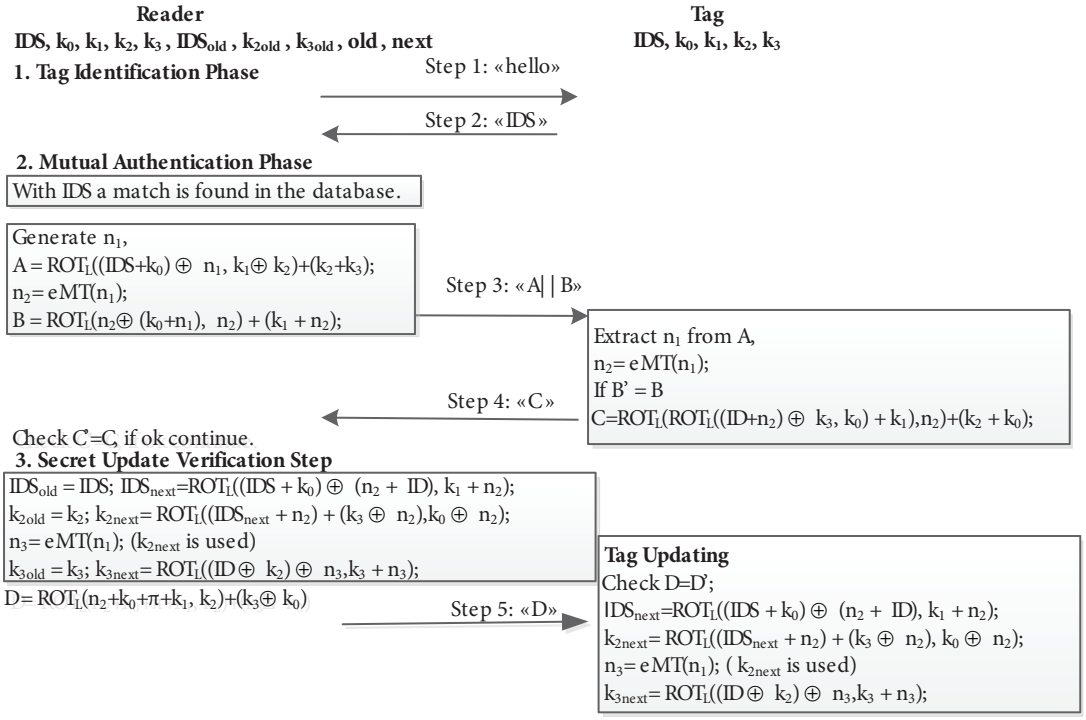

Figure 3. Proposed protocol, ULERAP.

3.1. The assumptions of ULERAP

It is generally assumed that the wired communication between the reader and the back-end server is secure, while the wireless channel between the tag and the reader is insecure. It is openly declared that as with none of the previous protocols, the ULERAP cannot resist active attacks. We assume that active intruders are detected by perimeter security measures such as security guards, cameras, and other detection equipment, which is not a great disadvantage in tags that will be used in a monitored environment like supply chains. The final assumption is that both sides have an IDS, ID, and the preshared secrets, k_0, k_1, k_2, and k_3, prior to the start of the message exchange. Secrets k_0 and k_1 are derived from one of the S-boxes of the AES algorithm. Secret k_0 can be the first 12 entries of the 1st row, while k_1 is the first 12 entries of the 3rd row [45]. Each tag application can choose a different row. Secret k_2 is rotated before being partially inserted in the eMT function to avoid using a repeated constant.

Rotation ROTL(x, y) is the circular shift of x to the left. Term y means the hamming weight of y. Rotation is a simple but clock-cycle consuming operation. The implementation is easy and requires 2 registers.

The term x is rotated for a minimum of 0 or a maximum of 96 times, but testing of y forces 96 clocks for each rotation.

3.2. The phases of our proposed protocol

There are 3 phases in the protocol. The first is the tag identification phase. The mutual authentication takes place in the second phase. The third phase involves the update of IDs and the secret keys.

Phase 1: Tag identification phase (Steps 1 and 2, in Figure 3).

The reader sends 'hello' and the tag replies with the IDS. The tag is identified in the database and its secrets are sent to the reader. If a match for the IDS is not found, the IDS_{old} values are searched. Upon locating IDS_{old}, a new exchange is started using it and the tag is authenticated. The tag always repeats the same IDS and does not store old IDS values.

Phase 2: Mutual authentication phase (Steps 3 and 4, in Figure 3).

The reader generates a nonce n_1 and using the preshared secrets obtained from the database, it hides n_1 in message A. Next, the reader produces variable n_2, by passing n_1 to function eMT (Figure 4). The eMT uses k_0, k_1, and k_2, forcing both sides to possess and use the secrets, correctly. Only n_1 is passed to the eMT because the secrets are used in order from the memory by a sequencer mechanism, explained in detail in [3]. The ARX-c rules of [30] and parts of the AES S-box are used in k_0, k_1 to obscure and rotate n_1 a varying number of times. Our proposed eMT produces unbiased, nonzero numbers, while Mixbits produces a zero output if a zero input is given [38]. The quality of n_2 is important and measured in Section 4.2.

```
Y = eMT(X){
      Y= k0 ⊕ X
      Y= k1 ⊕ Y
      Y = ROTL(k 2,Y)
      Y = Y AND 0x 0000FFFF0000FFFF00000000
      X = X AND 0x FFFF0000FFFF0000FFFFFFFF
      X = Y OR X
      Y = ROTR (X, 0x001FFFFFFFFFFFFF)
      Y = X ⊕ Y
      X = SHIFTL(Y, 0x003F)
      X = X AND 0x 9D2C56809D2C56809D2C5680
      Y = Y ⊕ X
      X = SHIFTL(Y, 0x7FFF )
      X = X AND 0xEFC60000EFC60000EFC60000
      Y = Y ⊕ X
      X = SHIFTR (Y, 0x0003FFFF )
      Y = Y ⊕ X    }
```

Figure 4. Proposed eMT function.

Using k_0, k_1, n_1, n_2, message B is formed and passed, as an authenticator. The tag extracts n_1 from A, if it knows the preshared secrets. Tag then calculates n_2 and its own version of B' to check against the received B. If B = B', n_1 has not been tampered with and the reader is authentic. The eMT guarantees that the tag has extracted n_1 correctly, and knows the scheme and all of the secrets. The ULERAP has one message less than Gossamer at this stage. Next, the tag prepares and sends its authenticator C. Message C contains the sensitive information ID number. Therefore, it is protected by substitution through addition and XOR operations, followed by transformation through double rotation operations. This makes the relationship

between the statistics of the encrypted data and the encryption key value complex, by providing confusion, nonlinearity, and preventing statistical analysis [45,46]. The reader calculates C' and authenticates the tag, if C' = C.

Phase 3: IDS and key update verification phase (Step 5, in Figure 3).

After verifying the tag, the reader updates its secrets first and with D, it signals the completion of mutual authentication and the update of the tag to go ahead. In our scheme, the reader updates first to resist the known IDS_{next} exploits, in the next exchange. The attacks that use IDS_{next} calculation to expose the terms of A, B, and C are prevented by obeying ARX-c rules [30]. The IDS and secret values of the past exchange, denoted with $_{old}$ suffixes, are stored in the server only, relieving memory of the tag.

If the tag receives no message or a bogus message, it will not update its secrets avoiding desynchronization. The update takes place only after verifying D = D', where D' is a value calculated by the tag. After IDS_{next} and k_{2next} are updated, the tag calculates n_3 and writes it over n_2. Finally, k_{3next} is calculated and the protocol finishes at this point.

4. Statistical tests

The RFID tags security and performance are the main focus of RFID security. We will analyze our proposed protocol's tag security in 2 areas. First of all, the results of the statistical tests for randomness will be discussed. Next, a security and performance analysis will be made. To prove the success of our protocol, the test results, security, and performance are compared with those of selected previous work. The UMAP, ULAP, SASI, and DIDT [40] protocols are left out, because they are not ARX systems and fail to resist passive attacks. It is not logical make comparisons against fully broken protocols or non-ARX systems. Gossamer is the only protocol that qualifies to be in the ARX-c category. In addition, Gossamer and ULERAP are the only protocols that propose the inclusion of a function for increasing the randomness of the exchanged messages and the IDS.

Our tests presented in the following sections are based on full 96-bit test software, instead of the present 32-bit versions. The software can be obtained on request from the authors. During the tests, true random numbers from http://random.org are used. The full test results and the data used in the tests are also available at http://srg.cs.deu.edu.tr/publications/2012/rfid. We only summarize part of the full results that match the declared results of previous work.

4.1. Explanation of the used statistical tests

Strongly biased results in an authentication protocol are undesirable, as they facilitate the cryptanalysis of the protocol. Bitwise AND and OR operations are examples of 2 strongly biased functions. When 2 random inputs are ANDed (ORed), the probability of obtaining a 0 (one) is 3/4. Since the XOR operation can be expressed in a combination of AND and OR operations, it is also strongly biased. The bitwise modulo 2^m addition is sometimes approximated to the XOR operation, by neglecting the carry at the end of the bit addition. Therefore, addition can also be biased. Based on this argument, the protocols using triangular functions and addition are expected to produce biased messages. This is depicted in the attacks cited and demonstrated in Section 2, on the UMAP and ULAP. To resist passive attacks based on the cryptanalysis of previous biased messages, exchanging correlated or analogous messages has to be avoided. To ensure that our designed protocol is not producing biased results, statistical tests are made for proving the randomness of messages A, B, C, D, and IDS, and the outputs of the eMT. The most popular statistical randomness tests are the famous Diehard, National Institute of Standards and Technology (NIST) (http://csrc.nist.gov/groups/ST/toolkit/rng/batteries_stats_test.html),

David Sexton's battery [47], and the avalanche test [48]. Since MT passes the Diehard test [43,44], the same test is not repeated for the eMT. Furthermore, neither the eMT of the ULERAP, nor the Mixbits of Gossamer are independent PRNGs. Therefore, there is no point in carrying out the full NIST battery tests, which are designed for testing the random number generators. Only the relevant pseudorandom number sequence test (ENT) randomness tests (also available from the above NIST link) of the NIST suite are carried out. Gossamer has not declared any test results. We chose the ENT, and David Sexton's and avalanche tests specifically. Many works have declared only partial test results, but the ULAP has detailed test results. Therefore, we use the same tests to compare the ULERAP and Gossamer objectively, 2 compatible ARX-c systems. We perform the tests for both and compare the results. With the comparisons, we aim to prove that our proposal is in the ultra-lightweight category, performs better, and has stronger security. However, first, an explanation of the ENT, and David Sexton's and the avalanche effect test is made in separate subsections.

4.1.1. The ENT test of NIST

The NIST gives a list of tests as batteries of statistical tests, to prove the randomness of the numbers generated by random number generators. ENT is one of the tests; it performs batteries of statistical tests on the input stream of bytes and bits, and produces 7 results. The first 2 are the serial correlation coefficient results, 1 for each byte and 1 for each bit of the tested input. The byte test measures the correlation of a byte to the previous byte, and bit test measures the correlation of a bit to the previous bit. For random sequences, the results (positive or negative) must be close to 0. The second, the chi-square test is the most commonly used test for randomness and it is extremely sensitive to errors in pseudorandom sequence generators. The chi-square distribution is calculated for the stream of bytes in the input and is expressed as an absolute number and a percentage, which indicates how frequently a truly random sequence would exceed the value calculated. The percentages given are interpreted as the degree to which the sequence tested is suspected of being nonrandom. If the percentage is greater than 99% or less than 1%, the sequence is almost certainly not random. For percentages of 95%–99% or 1%–5%, the sequence is suspect and for 90%–95% or 5%–10% the sequence is 'almost suspect'. Outside of these values the sequence is random. An arithmetic mean test is simply the result of the summing of all the bytes in the input and dividing by the input length. If the data are close to random, this test result should be about 127.50. In the next test, the result of the Monte Carlo value for the Pi test approaches the correct value of Pi, if the sequence is close to random. The last tests are entropy and compression tests, which are coupled to each other during evaluation. Entropy gives the information density of the contents of the file, expressed as a number of bits per character. A value close to 8.0 means an extremely dense file. Entropy is supported by the compression rate, such that as the information density increases, the compression rate decreases. A compression rate of 0 means the tested data are so random that they cannot be compressed any further.

4.1.2. The David Sexton battery tests

In David Sexton's tests, a P value is given to each test result. The P value is a percentile that shows whether the probability of the test result is equal to or greater than the one reported. P values for a given test should be more-or-less evenly distributed between 0 and 1. Numbers less than 0.1 are cautioned with asterisks to the right of a P value. Numbers greater than 0.9 are cautioned by carets ($\wedge$) placed to the right of a P value. The individual test entries are as follows:

Bit prediction tests try to predict the value of each bit of the sequence from the beginning to a predefined point in the sequence. The numbers of zeros and ones in the previous predefined number of bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted. In the

bit prediction A, B, C, D, E tests, the number of bits counted are, respectively, all, 1, 9, 17, 33, and 65 previous bits. In a random sequence the probability of success of any such prediction is 1/2. The number of successes is counted and a chi-squared statistic is calculated, whose degrees-of-freedom is 1.

The byte prediction A, B, C test suit tries to predict the value of each byte of the sequence from the beginning of the sequence to the end. The first byte of the sequence is predicted to equal 0. In a random sequence the probability of success of any such prediction is 1/256. The number of successes is counted. A chi-squared statistic is calculated. The degrees-of-freedom is 1. In the byte prediction A test, the next byte is predicted to be equal to all of the previous bytes bitwise XORed together. In the byte prediction B test, the next byte is predicted to be equal to the sum of all of the previous bytes, modulo 256. In the 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 the furthest back in the sequence. In the 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.

4.1.3. The avalanche test

The final test is the avalanche test, which measures nonlinearity. If a change in one of the input bits changes at least half of the output bits (avalanche), then it meets the desirable property of nonlinearity [48]. This ensures that if an adversary tries to flip one of the bits of the input and analyze the effect on the output (an attack cited in Section 2), the avalanche in the output will give no information about the function inside. Mathematically, the avalanche effect on F: $2^m \rightarrow 2^n$ is defined as:

$$\text{For all } x, y | H(x,y) = 1, \text{ Average } (H(F(x), F(y))) = n/2. \tag{1}$$

In our test, we follow the procedure for 8192 values, which is experimentally proved to be enough numbers for testing [48]. A total of 8192 true random numbers, each 96-bits long, are used from randomnumber.org. Next, 1 bit of the input is randomly flipped and the effect on the output is analyzed. For each flip, the input bits are compared with the output bits and the number of flips (which is the difference) is divided by 96. Dividing the avalanche effect in Eq. (1) by 96, we make this test independent of the number of bits n. The expected average is 0.5.

4.2. The statistical test results for eMT

ENT batteries of statistical tests were run for the eMT, Mixbits outputs, and XORed values of 2 consecutive outputs. The results of the 7 tests are given in Table 2. The first 2 rows are the results of the serial correlation coefficient for the byte and bit levels. As the results must be close to zero for randomness, the eMT and Mixbits results satisfies the randomness criteria. The next test is the chi-square test. In order not to suspect randomness, the values must be between 10% and 90%; except for the Mixbits (99.75%), the results satisfy the criteria. According to this test, Mixbits may not be a good random number generator. The following test is the arithmetic mean test and closeness to 127.5 is required. The eMT and XORed consecutive eMT results are closer to the required value. The result of the Monte Carlo value for the Pi test approaches the correct Pi value for both functions; hence, both functions pass this test with error rates of less than 0.1%. The final 2 tests are the entropy and compression rate. In Table 2, both the eMT and Mixbits have fair scores for entropy (very

close to 8) and 0 compression values. The ENT test results comparison is in favor of the eMT function as it has no values outside of those required, while Mixbits fails a critical test.

Table 2. Comparison of ENT test results of eMT and Mixbits.

Test	eMT	eMT ($Z_i \oplus Z_i+1$)	Mixbits	Mixbits ($Z_i \oplus Z_i+1$)
Byte	0.000946	0.000223	–0.000152	–0.000331
Bit	–0.000437	0.000067	–0.000043	0.000522
Chi-square statistics	269.73 (25.16%)	271.84 (22.39%)	196.23 (99.75%)	250.50 (56.68%)
Arithmetic mean (127.5 = random)	127.4784	127.5183	127.3920	127.6035
Monte Carlo π estimation	3.143188477	3.141167764	3.14262390	3.138634786
	(0.05% error)	(0.01% error)	(0.03% error)	(0.09% error)
Entropy (bits/byte)	7.999876	7.999875	7.99991	7.999885
Compression rate	0%	0%	0%	0%

The David Sexton's battery test results also show that our scheme has better results. In Table 3, the eMT has only the bit prediction B statistic result close to the unacceptable 0 value. However, Mixbits has 2 values close to the unacceptable value 1; one in the bit prediction C and one in the D statistics tests. The eMT also performs better than Mixbits in the David Sexton battery tests.

Another result in favor of the eMT is in the critical avalanche effect (Table 4). Gossamer mentions the test but fails to provide results; therefore, we run the tests for Mixbits. Our proposed function complies with the avalanche effect with a result of 0.50633035325733, but Mixbits fails the test with a score of 0.32932535797499. This result proves that for a 1-bit flip in the input of Mixbits, half of the output bits are not flipped. In other words, Mixbits produces biased nonces for Gossamer.

Table 3. Comparison of eMT and Mixbits David Sexton test results.

Test	eMT (P value)	Mixbits [9] (P value)
Bit prediction A statistic	0.6225	0.7398
Bit prediction B statistic	0.0220*	0.5521
Bit prediction C statistic	0.1815	0.9093∧
Bit prediction D statistic	0.4112	0.9422∧
Bit prediction E statistic	0.1705	0.5012
Byte prediction A statistic	0.2871	0.5837
Byte prediction B statistic	0.7071	0.6498
Byte prediction C statistic	0.5730	0.6054
Byte prediction D statistic	0.7661	0.8756

Table 4. Avalanche effect results.

	Avalanche results
A	0.019550
B	0.495568
C	0.494485
D	0.493333
IDS	0.494567
eMT	0.506331
Mixbits	0.329325
k_2	0.492003
k_3	0.496311

As an overall evaluation of the results, it can be concluded that the eMT produces unbiased and nonlinear results. Based on this success, we use the eMT in the design of messages and IDS. The next step is to see the effects of the eMT on the test results of IDS.

4.3. The statistical tests for IDS

Producing biased IDS values results in predicting its next value or tracing a tag. Therefore, the IDS is one of the most important variables in the RFID tag. To prevent traceability, the IDS is changed at the end of every run. An adversary aims to estimate the next IDS value by analyzing the previous values. If a relation between the IDS values is found, the tag can be singled out, wherever it goes. Therefore, it is fundamental to test the statistical properties of the produced IDS values. The proof that our scheme does not produce biased IDS values is given in Tables 4, 5, and 6.

The avalanche test result of the IDS (0.494567) in Table 4 is in line with the requirement that it should be around 0.5. This result shows that an adversary cannot gain much information by flipping a bit and analyzing the result on the output.

The test results of the ENT for an input of 2^{17} bytes of IDS data are shown in Table 5. The second column of Table 5 is acquired after testing the results of the IDS. The last column is the result for the 2 consecutive IDS values XORed with each other, which examines the relation between consecutive IDS values. The serial correlation coefficient results in both columns are very close to 0, meaning that the next bits of bytes of IDS values cannot be guessed. The chi-square statistics results are far from the 2 undesirable extremes of 0% and 100%, which shows that both the IDS and XORed IDS values cannot be suspected of being nonrandom. Since the arithmetic mean results are close to 127.50, it is proof that the IDS and XORed IDS data are random. The deviation from the correct Pi value is 0.07% and 0.19%, respectively. This result in the Monte Carlo Pi estimation also supports the randomness of our IDS values. With entropy results very close to 8 and compression results of 0, our IDS values are random.

Table 5. ENT test results of IDS.

Test	IDS	IDS ($Z_i \oplus Z_i+1$)
Byte	0.001224	0.000408
Bit	–0.000160	–0.000523
Chi-square statistics	250.71 (56.42%)	253.01 (52.35%)
Arithmetic mean (127.5 = random)	127.4951	127.4550
Monte Carlo π estimation	3.139434814 (error, 0.07%)	3.147591763 (error, 0.19%)
Entropy (bits/byte)	7.999885	7.999884
Compression rate	0%	0%

The IDS performs well in David Sexton's battery tests, too (Table 6). It has 2 in the undesired high score range but they are below the risky 0.95 value. The rest of the tests are satisfactory and to an adversary listening in the air, the IDS seems random. Therefore, using the eMT function, the randomness of IDS is provided.

4.4. Statistical test results of messages A, B, C, D, k_2, and k_3

The ENT, avalanche, and David Sexton's battery tests are carried out for messages A, B, C, and D, and regularly updated shared secrets k_2 and k_3. During the tests, 2^{17} bytes of data are used for the ENT and David Sexton's battery tests, and 8192 are used for the avalanche test. Detailed evaluation of each message and secret is not practical, but a brief summary is provided next. According to the ENT test results in Table 7, only message D's

chi-square test result is in the almost suspect range, all of the other messages and secrets are in the acceptable range. Therefore, they all pass the ENT test suite.

Table 6. David Sexton test results of messages and secrets.

Statistic (P value)	A	B	C	D	k_2	k_3	IDS
Bit prediction A	0.2832	0.8678	0.0375*	0.7036	0.4100	0.0118*	0.8010
Bit prediction B	0.5212	0.0655*	0.8505	0.4442	0.9076∧	0.9983∧∧	0.6382
Bit prediction C	0.2645	0.5043	0.9884∧	0.6259	0.7466	0.8246	0.4336
Bit prediction D	0.5526	0.7067	0.9389∧	0.8803	0.1764	0.9076∧	0.1731
Bit prediction E	0.9422∧	0.6108	0.2091	0.1879	0.6441	0.7628	0.9433∧
Byte prediction A	0.1543	0.5108	0.6611	0.1938	0.1543	0.2048	0.7071
Byte prediction B	0.4714	0.3806	0.8143	0.3089	0.6498	0.4909	0.9252∧
Byte prediction C	0.1212	0.1175	0.2531	0.9127∧	0.3639	0.9501∧	0.5945
Byte prediction D	0.2531	0.4246	0.8879	0.0795*	0.5519	0.3892	0.2731

Table 7. ENT test results of A, B, C, D, k_2, and k_3.

Test	A	B	C	D	k_2	k_3
Byte	0.001624	–0.000237	–0.000006	0.001039	–0.000024	0.000306
Bit	0.000083	–0.000279	–0.000216	0.000021	0.000261	0.000190
Chi-square statistics	244.10 (67.73%)	264.93 (32.14%)	269.71 (25.19%)	219.72 (94.64%)	263.92 (33.72%)	236.32 (79.35%)
Arithmetic mean (127.5)	127.4395	127.4449	127.4078	127.5115	127.5978	127.4093
Monte Carlo π estimation	3.149490356 (0.25% error)	3.141967773 (0.01% error)	3.144104004 (0.08% error)	3.146194458 (0.15% error)	3.139389038 (0.07% error)	3.133438110 (0.26% error)
Entropy (bits/byte)	7.999888	7.999879	7.999876	7.999899	7.999879	7.999892
Compression rate	0%	0%	0%	0%	0%	0%

In the David Sexton's battery tests (Table 6), only k_3 performs poorly in the prediction tests. Hence, k_3 is deliberately not used in the eMT or IDS_{next}. The performance of k_3 can be improved by calling the eMT once more. However, to avoid additional computation costs, no more calls are made. Messages A, C, and D are freed from k_3's poor performance by the ARX operations. This is depicted in the satisfactory results in Tables 6 and 7. The avalanche test result for message A is poor, which contains the nonce n_1 generated by the reader. However, this is not a disadvantage as long as an adversary cannot isolate n_1. Messages B, C, D, and IDS depend on message A. In spite of message A's poor performance, the eMT introduces randomness into all of the messages and IDS.

5. Security and performance analysis

A mutual authentication algorithm is considered secure if the parties involved in the communication prove themselves to each other without exposing the IDS, shared secrets, or locations. It is also important that no outsider succeeds to prove herself as a legitimate party in the protocol. The algorithms studied in Section 2 clearly fail these goals. The security of an algorithm can be increased if it can be demonstrated that it does not suffer from the known attacks effective on previous protocols. Below, we show that our proposal does not suffer from vulnerabilities and has the extra benefit of consuming considerably less die area and fewer clock cycles. A rotational cryptanalysis and resistance against known attacks is presented in the security analysis. In the

performance analysis, the ULERAP, Gossamer, and some popular hashing/encryption schemes are compared. Possible attacks against the ULERAP are also considered.

5.1. Rotational cryptanalysis and use of AES s-boxes

Rotational cryptanalysis concentrates on addition-rotation-XOR (ARX) systems [30], such as the ULERAP and Gossamer. A test of vulnerability against rotational analysis is provided. For a given number of additions (q), the logarithmic probability (pr) of any ARX scheme for $q < -t / [\log_2(pr)]$ is claimed to be vulnerable to rotational cryptanalysis [30]. Word length t is L = 96 in our scheme. The value of $\log_2$ (pr) tends to be 2, resulting in q = 48. That is to say, any scheme with less than 48 rotations, i.e. ROT(x, y), where ham(y) < 48, is considered insecure. In our scheme, around 10 rotation operations are executed, each for a random number of times. In the eMT, a rotation ROT_R(X, 0x001FFFFFFFFFFFFF) is included. Ham(0x001FFFFFFFFFFFFF) is 53, putting our scheme in the secure category. A larger number of rotations increases $\log_2$ (pr) and decreases the vulnerability risk.

In addition, parts of the reliable AES S-box are substituted into the terms guaranteeing a nonzero, large, random number of rotations that avoids the regular DES S-box weaknesses hinted at in [49]. Finally, message C is obtained by 2 consecutive rotation operations, one for k_0 times and one for a random number of n_2 times. Randomness obtained by a random number of rotations is an important security improvement [50].

5.2. Applying known attacks to our proposed ULERAP scheme

The known passive attacks [19–24] launched against the previous triangular protocols [16–18] cannot be launched on our scheme because they are incapable of rotational analysis. The only attacks that can be tried on our proposal are those in [28,38] and our attack, in Section 2.3.

The authors in [28] flipped the bits of the first message of the reader, one by one, and examined the outcome reflected in the responses of the tag. Using the changed bit values, the attack was escalated by exploiting the algebraic weaknesses of XOR/OR equations. The attack was finalized by a probabilistic approach on the binary status of the XORed secrets. The attacks elaborated in [28] do not work on the ULERAP because the XOR and additions are encapsulated in rotations of unpredictable amounts making the analysis of the status of a specific bit position impossible.

The attack of all zeros [38] does not work on the ULERAP because k_0 and k_1 are taken from the AES S-boxes and are never zeros. If only k_2 and k_3 are zeros, the attack cannot succeed because messages B, C, D, IDS_{next}, and k_{2next} do not reduce to simple equations, because the eMT does not produce zero-valued n_2, n_3. The only reductions are $A = ROT_L((IDS + k_0) \oplus n_1, k_1)$ and $k_{3next} = ROT_L((ID \oplus n_3, n_3)$, which cannot be used in other equations. Even if the k_2, k_3, and n_1 values are all zeros, the attack does not escalate. Only A reduces to an unsolvable equation, $A = ROT_L(IDS + k_0, k_1)$. The increased security is due to nonzero n_2, n_3 and a different combination of terms in B, C, D, and IDS_{next}.

Next, the attack of all ones is tried on each message. The reduced equations are $A = (k_2 + k_3) - 1$, $B = (k_1 + n_2) - 1$, $C = ROT_L(k_1 - 1, n_2) + (k_2 + k_0)$, and $D = (k_3 + k_0) - 1$. C does not reduce any further because $k_1 - 1$ or k_0 is never zero, or all ones. Each equation is an unsolvable 2-unknown equation. With a probability of $1 / 2^{96}$ for each reduction, the total probability is $1 / 2^{4*96}$; 2×2^{192} times less than the attack in [9]. The probability of IDS_{next} being all ones is also $1 / 2^{96}$. In conclusion, none of the known attacks succeed in exposing the hidden secrets in our scheme.

5.3. Tag data confidentiality, integrity, and forgery resistance

The confidentiality of the tag ID and hence the user privacy, is of outmost importance. In our scheme, the ID appears only in message C. It is concealed inside 2 consecutive rotations obscured by nonces and preshared secrets, of which none are passed in clear text.

The integrity of the data shared has 2 aspects. One is the integrity of the messages passed, and hence the data inside of them. The other is the data stored on the tag. If the contents of a message are changed and go undetected, the parties can be tricked into agreeing to update to malintended values. Any changes on the messages are detected through nonce n_1 and variable n_2. Nonce n_1 is passed in encrypted form, inside A to the tag. Any tampering with A will cause the tag to end up with a different n_1, resulting in a different n_2. This will cause a mismatch between the calculated B' and passed B. Similarly, any tampering with C or D will also be detected. Hence, data integrity during communication is guaranteed.

The Gen-2 tags are not improvised to resist tampering with their memory; therefore, the tags are susceptible to physical attacks. An attacker may obtain the memory contents and hence the IDS, k_0, k_1, k_2, and k_3. Because this attack requires technical equipment, the tag has to be removed from the premises and brought back. Supply chains must remove sold items immediately from their databases to avoid this type of attack.

5.4. Tag untraceability and forward security

If the tag outputs recognizably similar messages, it can be traced and the location privacy is revealed. The entropy and avalanche test results show that our IDS values have better behavior than those of Gossamer's. By sending seemingly random values, the tag cannot be singled out and its anonymity is provided. As the tag always responds with the same IDS until a successful authentication, it can be traced. This is not a breach of security, as the displacement of unauthenticated tags is not common.

Forward security requires that a compromised session key or secret should only affect its session, and not endanger earlier sessions [3]. The ULERAP supports this property because capturing the values n_1, n_2, k_2, and k_3 in a session is not sufficient to decrypt the past sessions, as these terms are updated every session and need to be used all together.

5.5. Passive attacks

We test our design against all of the referenced attacks. None of the successful attacks on triangular functions succeed, as the ULERAP is an ARX-c system. None of the secrets can be deduced by bit insertion or bit flipping. Our technique used to attack Gossamer is successful because the protocol uses only double rotations for obtaining the messages. We use partial AES S-boxes to add constants to the rotations, turning the scheme into an ARX-c system, which are well defended in [30].

The rotational analysis of all ones in Section 2.2.3 used to obtain equations does not work, because the analysis in Section 5.2 shows that the ULERAP's messages cannot be reduced to one unknown equation. For an all zeros attack, the probability of the 3 terms k_2, k_3, and n_1 being zeros is $1 / 2^{288}$; still cryptanalysis is not possible. An active attacker may try to impose all zeros, but k_0, k_1 from the AES S-box and eMT do not allow further zeros.

5.6. Active attacks

The passive ULERAP tags are not equipped to resist sophisticated active attacks. The gain obtained by actively attacking a single tag is highly unfavorable compared to the danger of being detected. Meanwhile, the silent

recording of all communication for later analysis at a secure hideout is more dangerous. A single decoded exchange may lead to the exposure of the information on all tags. Therefore, passive attacks are more beneficial to an attacker. Despite the reported risks and threats of using RFID technology, it is very unlikely that an attacker would actively interfere in the presence of a legitimate reader without being noticed. If all of the RFID abuse reports were true, the tags would not have been so popular in supply chains. Nevertheless, traditional perimeter-measures are always good to counter such attacks [3], as explained in Section 5.6.2.

5.6.1. Denial of service, replay, exhaustion, and disclosure attacks

In the first phase of the authentication, neither the reader nor the tag knows the opposite. Therefore, just a 'hello' and a simple 'IDS' opens the avenue to the denial of service, replay, and exhaustion attacks of [6,37,38]. However, these attacks are easily detectable and cannot escalate into the mutual authentication phase, as the tag checks B' = B. Any mismatch here stops the exchange. A fresh nonce n_1 is used in every session and the IDS, n_2, k_2, and k_3 are recalculated, preventing a replay attack. Disclosure attacks try to reach the ID value. The ULERAP prevents the disclosure of the ID by encrypting it in step 4 (Figure 3). The increased security of the encryption of messages A, B, and the avoidance of repeating the same terms in C are explained in Section 5.2.

5.6.2. Desynchronization attack and IDS collision

Desynchronization is possible mainly because previous protocols allow the tag update first and then ignore any acknowledgement. The authors in [6] and [13] exploited this weakness in their attack on Gossamer. Adding only a synchronization bit s as in [13] is not enough either. In our protocol, the reader updates first and signals the tag to update its secrets by message D. This helps to resist tag desynchronization, but allows reader desynchronization. For example, selectively blocking the reader's message D (Figure 3) in 2 consecutive exchanges tricks the reader into updating its IDS value twice, but leaves the tag with the old, irrelevant IDS value. This permanently blocks any future authentication. Although possible, this type of desynchronization can be detected and prevented. An increased number of desynchronized tags in an area is easily detected. The perimeter security equipment mentioned in Section 3.1 would reveal the presence of an active attack. The attack can be prevented if the database server keeps 2 counters as a sequence number of the old and new secret values; for example, one for IDS_{next} and one for IDS_{old} (Figure 3). Never letting the difference between the sequence numbers of IDS_{next} and IDS_{old} be greater than one, i.e. never allowing 2 consecutive IDS_{old} authentications, is the trick.

The occurrence of IDS collision (same IDS for 2 tags) is a possibility to be considered [6]. To avoid IDS collision, if IDS_{next} happens to be a presently used IDS, instead of sending message D, the reader is instructed to start a new round. Since the server knows the collision, the old IDS is expected in the retry.

5.7. Performance analysis

The clock cycles of the computations needed for the decryption of a challenge, generation of a nonce, and the encryption of a reply deteriorate the number of tags read per second. Moreover, a complex PRNG function increases the die area, and hence the UHF tag's cost. Because of these facts, a faster and a smaller deterministic random number generator is more acceptable. Below we try to prove that the ULERAP is a valid candidate.

Word length L is 96 bits, which is compatible with all encoding schemes accepted by the EPC [8], but the messages can be processed in m-bit blocks. Gossamer analyzes 4 different word lengths (m= 8, 16, 32, 96) and

proposes a possible architecture and the logical memory map conforming to Gen-2 RFID specifications. The guidelines of the implementation in Gossamer for m = 16 set a fair comparison ground. As will become clear, m = 16 is also necessary for conforming to clock-cycle constraints. Since the eMT uses the same simple operations in a sequential manner, the same arithmetic logic unit (ALU) of Gossamer can be used, which facilitates fair comparison. Two registers and the same signaling are enough to select inputs from the stored values for the eMT function. The control signal selects the operation that will be used in the ALU. It should be noted that the Mixbits of the Gossamer protocol requires 3 registers.

Table 8 shows the estimation of chip areas, which uses the gate equivalents (GEs) of [51] and the amount required for each logic gate [52]. A total GE for each operator is obtained after multiplying by m, i.e. 16. Although the ULERAP has additional OR, AND, and NOT operations, it uses 2 registers. Gossamer requires 3 registers because Mixbits is passed by 2 parameters and an additional register is required to hold temporary values. The ULERAP's total GE of 680 gates is smaller than Gossamer and well below the critical value of a few thousand gates. As every 1000 GE adds US$0.01 to the cost [50] and the power consumption is proportional to the number of gates, the ULERAP is well in the low-cost, ultra-lightweight category.

Table 8. Gate equivalents of ULERAP and Gossamer (16-bit architecture).

Operator	No. used	Logic	GE [53]	ULERAP	Gossamer
Register	2	Flip flop	5.33	170.56	255.84
Adder	1	6 gates	1.78	170.88	170.88
Shifter	1	Flip flop	5.33	85.28	85.28
AND	1	Gate	1.33	21.28	-
OR	1	Gate	1.33	21.28	-
XOR	1	Gate	2.67	42.72	42.72
NOT	1	Gate	0.67	10.72	-
Total				522.72	554.72
Control	-	Gates	30%	156.82	166.42
Grand Total				679.54	721.14

Other performance features are given in Table 9. The operation types of both schemes are in the simple bitwise category, except that the ULERAP is sequential and Gossamer is not. One important feature is the memory requirement on the tag. Memory contains the IDS, ID, constants, and the keys k_0, k_1, k_2, and k_3. The constants of the eMT and k_0, k_1 can be programmed into the reserved area of the ROM, as in [3]. The remaining IDS, k_2, and k_3 values (3L) are kept in the user memory. Nowadays, 512 bits (>5 L) of user RAM are common in commercial UHF Gen-2 tags, leaving an additional 2 L of space for storing temporary values. While both protocols pass the same amount of messages, the ULERAP uses more space on the server at the expense of more security. For a server with gigabytes of disk space, this is no longer a handicap.

Table 9. Other performance features of ULERAP and Gossamer (16-bit architecture).

Performance feature	Gossamer	ULERAP
Operation types	$\oplus, +$, Rot, Mixbits	$\oplus, +, \wedge, \vee$, Rot, eMT
Memory size on the tag	7 L	9.5 L
Memory size on the database	6 L	13 L
Total messages passed	6 L	6 L

Next, the number of clock cycles has to be checked against the limit of 1800 in [53] or 2000 in [54] for the ultra-lightweight category. The tag's work at 100 kHz and a reader communicates with the tags using

an interleaved protocol [53,55]. The ULERAP and Gossamer both satisfy the 1800 clock cycles with 16-bit architecture design (m = 16). In Table 10, our design's tag finishes computations in 1458 clocks, which is less than the 1800 clocks and hence conforms to the ultra-lightweight category of [5,54]. Meanwhile, Mixbits is called twice in Gossamer, which costs 4 × 32 × 96 / m computations per call. For m = 8, Gossamer fails and barely satisfies the 1800 limit for m = 16. The ULERAP consumes less time than Gossamer and the eMT costs only 255 clocks.

Table 10. Comparison of gate count-clock cycle products based on [53].

Performance parameter	Gossamer	SHA-1	AES [55]	SHA-256	ULERAP
Comp. cost (clock cycles)	1752	1274	1032	1128	1458
Chip area (GE)	721	8120	3400	10,868	680
Area × delay (complexity)	1,263,192	10,344,880	3,508,800	12,259,104	991,440

The chip area × delay value (area-delay product) is a measure of complexity [54]. Table 10 shows that our scheme has lower complexity than Gossamer's hashing and encryption implementations. A special AES design consumes around 3400 GE and uses 1032 clock cycles to finish a 128-bit encryption [54]. A protocol where a tag performs one encryption for authenticating the reader, one encryption for preparing its authenticator, and a random number generation (for the eMT equivalence), typically consumes 1032 × 2 × 3400 + 255 (eMT) = 7,017,855 gate-clocks. Hence, the complexity of any AES-based proposal is more than 7.07 times that of our proposal.

The area-delay product for hashing functions is very high, even for 32-bit architectures. The complexity values clearly indicate that encryption and hashing schemes are not in the ultra-lightweight category. However, tag manufacturers can wait until they are viable, during which ultra-lightweight cryptography is available.

5.8. Possible attacks on our protocol

Gen-2 tags are not robust against physical tampering or side channel attacks like electromagnetic or power analysis methods that may reveal stored secrets on the tag. Advanced electronic security techniques are needed to resist these attacks.

For special cases where the x and y values of operation Rot(x, y) are all ones or zeros (Section 2.2.3), an active rotational cryptanalysis attack may be launched to obtain the terms of A, B, C, or D. Although not very likely, messages A, B, or C can be reduced to equations with 2 unknowns.

6. Conclusion

This paper outlines a security analysis of previous ultra-lightweight authentication protocols. Passive and active attacks are considered. The failure of the previous protocols is demonstrated to be the result of repeated weaknesses in protecting the secret ID number. A protocol, the ULERAP, is proposed, which attains the indispensable property of confusion and nonlinearity by including parts of AES S-boxes into rotation operations and by a random number of rotations. Security is further increased by reconstructing a proven random number generator as a deterministic function eMT. The eMT's success in providing nonzero, unbiased, random variables is clearly depicted in the messages generated. The ARX-c system rules are introduced and obeyed in the proposal. A 'first reader-updates' algorithm and tag update-signaling are used. To resist desynchronization and disclosure attacks, keeping sequence numbers on the server side for updated values is recommended.

Stronger test results prove that the proposed protocol offers better resistance against known attacks than the previous vulnerable protocols. In addition, the proposed scheme has 7 times lower complexity than AES-based protocols, which is today's widely accepted symmetric cryptography algorithm. Unfortunately, incorporating AES in low-cost tags is not yet possible. Therefore, a cheaper, implementable algorithm in the ultra-lightweight category is needed. With its simple bitwise operations and small complexity, the ULERAP is a viable option until the AES becomes available in low-cost tags.

Future work involves the physical implementation of a tag circuit and the design of a stateful server database that keeps track of authentication steps. While efforts towards compact encryption designs suitable for passive tags are intensified, our protocol is a viable option to implement for better security.

References

[1] C.M. Robert, "Radio frequency identification", Computers and Security, Vol. 25, pp. 18–26, 2006.

[2] R. Das, P. Havrop, "RFID forecasts, players and opportunities 2011–2021", 2010. Available at http://www.idtechex.com/research/reports/rfid_forecasts_players_ and_opportunities_2011_2021_000250.asp (Last accessed: 10 October 2012).

[3] P.P. Lopez, J.C.H. Castro, J.M.E. Tapiador, A. Ribagorda, "An ultra light authentication protocol resistant to passive attacks under the Gen-2 specification", Journal of Information Science and Engineering, Vol. 25, pp. 33–57, 2009.

[4] S.E. Sarma, S.A. Weis, D.W. Engels, "RFID systems and security and privacy implications", Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems, pp. 454–470, 2002.

[5] H.Y. Chien, "SASI: A new ultra lightweight RFID authentication protocol providing strong authentication and strong integrity", IEEE Transactions on Dependable and Secure Computing, Vol. 4, pp. 337–340, 2007.

[6] Z. Bilal, A. Masood, F. Kausar, "Security analysis of ultra-lightweight cryptographic protocol for low-cost RFID tags: Gossamer protocol", International Conference on Network-Based Information Systems, pp. 260–267, 2009.

[7] ISO/IEC 18000-6:2010. Available at http://www.iso.org/iso/iso_catalogue/ catalogue_tc/catalogue_detail.htm?csnumber=46149 (Last accessed: 21 October 2012).

[8] Gen-2, 2008. Class-1 Generation 2 UHF Air Interface Protocol Standard, Ver. 1.2.0. Available at http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2 (Last accessed: 31 October 2012).

[9] P.P. Lopez, J.C.H. Castro, J.M.E. Tapiador, A. Ribagorda, "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol", Information Security Applications, pp. 56–68, 2008.

[10] T. Van Deursen, S. Radomirovic, "Attacks on RFID protocols", Cryptology ePrint Archive, Report 2008/310, 2008.

[11] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber, "Modeling attacks on physical unclonable functions", Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 237–249, 2010.

[12] U. Rührmair, J. Sölter, F. Sehnke, "On the foundations of physical unclonable functions", IACR Cryptology ePrint Archive, pp. 277, 2009.

[13] K.H. Yeh, N.W. Lo, "Improvement of two lightweight RFID authentication protocols", Information Assurance and Security Letters, Vol. 1, pp. 6–11, 2010.

[14] I. Vajda, L. Buttyán, "Lightweight authentication protocols for low-cost RFID tags", Proceedings of the 7th IFIP WG 11.2 Workshop on Security in Ubiquitous Computing, 2003.

[15] A. Juels, "Minimalist cryptography for low-cost RFID tags", Proceedings of the 4th International Conference on Security in Communication Networks, pp. 149–164, 2005.

[16] P.P. Lopez, J.C.H. Castro, J.M.E. Tapiador, A. Ribagorda, "M2AP: a minimalist mutual-authentication protocol for low-cost RFID tags", Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing, pp. 912–923, 2006.

[17] P.P. Lopez, J.C.H. Castro, J.M.E. Tapiador, A. Ribagorda, "LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags", Workshop on RFID Security, 2006.

[18] P.P. Lopez, J.C.H. Castro, J.M.E. Tapiador, A. Ribagorda, "EMAP: an efficient mutual authentication protocol for low-cost RFID tags", OTM Federated Conference and Workshop: IS Workshop, pp. 352–361, 2006.

[19] T. Li, R. Deng, "Vulnerability analysis of EMAP - an efficient RFID mutual authentication protocol", Proceedings of the 2nd International Conference on Availability, Reliability and Security, pp. 238–245, 2007.

[20] T. Li, G. Wang, "Security analysis of two ultra-lightweight RFID authentication protocols", Proceedings of the IFIP TC-11 22nd International Information Security Conference, Vol. 232, pp. 109–120, 2007.

[21] C. Hung-Yu, H. Chen-Wei, "Security of ultra-lightweight RFID authentication protocols and its improvements", ACM SIGOPS Operating Systems Review, Vol. 41, pp. 83–86, 2007.

[22] M. Barasz, B. Boros, P. Ligeti, K. Loja, D. Nagy, "Breaking LMAP", Conference on RFID Security, pp. 69–78, 2007.

[23] M. Barasz, B. Boros, P. Ligeti, K. Loja, D. Nagy, "Passive attack against the M2AP mutual authentication protocol for RFID tags", Proceedings of the 1st International EURASIP Workshop on RFID Technology, 2007.

[24] B. Alomair, L. Lazos, R. Poovendran, "Passive attacks on a class of authentication protocols for RFID", Proceedings of the 10th International Conference on Information Security and Cryptology, pp. 102–115, 2007.

[25] R.C.W. Phan, "Cryptanalysis of a new ultralightweight RFID authentication protocol – SASI", IEEE Transactions on Dependable and Secure Computing, Vol. 6, pp. 316–320, 2009.

[26] J.C.H. Castro, J.M.E. Tapiador, P.P. Lopez, J.J. Quisquater, "Cryptanalysis of the SASI ultralightweight RFID authentication protocol", IEEE Transactions on Dependable and Secure Computing, Submitted 2008.

[27] H. Sun, W. Ting, K. Wang, "On the security of Chien's ultralightweight RFID authentication protocol", IACR Cryptology ePrint Archive, pp. 83, 2008.

[28] P. D'Arco, A. De Santis, "On ultralightweight RFID authentication protocols", Transactions on Dependable and Secure Computing, Vol. 8, pp. 548–563, 2011.

[29] T. Cao, E. Bertino, H. Lei, "Security analysis of the SASI protocol", Transactions on Dependable and Secure Computing, Vol. 6, pp. 73–77, 2009.

[30] D. Khovratovich, I. Nikolic, "Rotational cryptanalysis of ARX", 17th International Conference on Fast Software Encryption, pp. 333–346, 2010.

[31] J.C.H. Castro, P.P. Lopez, R.C.W., Phan, J.M.E. Tapiador, "Cryptanalysis of the David-Prasad RFID ultra-lightweight authentication protocol", Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues, pp. 22–34, 2010.

[32] S.H. Wang, G.L. Wang, "Analysis of passive attack on RFID authentication protocol ULAP", Networks and Communications, Vol. 36, pp. 17–19, 2010.

[33] RFID Security & Privacy Lounge, 2012. Available at http://www.avoine.net/rfid (Last accessed: 21 October 2012).

[34] R. Bassil, W. El-Beaino, W. Itani, A. Kayssi, A. Chehab, "PUMAP: a PUF-based ultra-lightweight mutual-authentication RFID protocol," International Journal of RFID Security and Cryptography, Vol. 1, pp. 58–66, 2012.

[35] M. Safkhani, N. Bagheri, M Naderi, "Security analysis of a PUF based RFID authentication protocol", IACR Cryptology ePrint Archive, pp. 704, 2011.

[36] G. Avoine, X. Carpent, "Yet another ultralightweight authentication protocol that is broken", IACR Cryptology ePrint Archive, pp. 691, 2011.

[37] N. Rama, R. Suganya, "SSL-MAP: a more secure Gossamer-based mutual authentication protocol for passive RFID tags", International Journal of Computer Science and Engineering, Vol. 2, pp. 363–367, 2010.

[38] E.G. Ahmed, E. Shaaban, M. Hashem, "Lightweight mutual authentication protocol for low cost RFID tags", International Journal of Network Security & Its Applications, Vol. 2, pp. 27–35, 2010.

[39] "EPC Global Class1 Gen2 RFID Specifications", 2005. Available at http://www.alientechnology.com/docs/AT_wp_EPCGlobal_WEB.pdf (Last accessed: 27 February 2012).

[40] Y.C. Lee, "Two ultralightweight authentication protocols for low-cost RFID tags", Applied Mathematics and Information Sciences, Special Issue, pp. 425–431, 2012.

[41] Y.C. Lee, Y.C. Hsieh, P.S. You, T.C. Chen, "A new ultralightweight protocol with mutual authentication", WASE International Conference on Information Engineering, Vol. 2, pp. 58–61, 2009.

[42] P.P. Lopez, J.C.H. Castro, J.M.E. Tapiador, J.C.A. Van der Lubbe, "Security flaws in a recent ultralightweight RFID protocol", Workshop on RFID Security, Cryptology and Information Security Series, pp. 83–93, 2010.

[43] M. Matsumoto, T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", ACM Transactions on Modeling and Computer Simulation, Vol. 8, pp. 3–30, 1998.

[44] F. Panneton, P. L'Ecuyer, M. Matsumoto, "Improved long-period generators based on linear recurrences modulo 2", ACM Transactions on Mathematical Software, Vol. 32/1, pp. 1–16, 2006.

[45] J. Daemen, V. Rijmen, "AES proposal: Rijndael", 1999. Available at http://csrc.nist.gov/archive/aes/index.html (Last accessed: 31 October 2012).

[46] W. Stallings, "Cryptography and Network Security", Pearson Education, 5th Ed., pp. 96–97, 2011.

[47] David Sexton's battery, 2005. Available at http://www.oocities.org/da5id65536/ (Last accessed: 31 October 2012).

[48] J.C.H. Castro, J.E. Tapiador, A. Ribagorda, B. Ramos, "Wheedham: an automatically designed block cipher by means of genetic programming", IEEE Congress on Evolutionary Computation, pp. 192–199, 2006.

[49] E. Aras, M.D. Yücel, "Performance evaluation of safer K-64 and S-boxes of the safer family", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 9, pp. 161–175, 2001.

[50] P.P. Lopez, P.T. Lim, T. Li, "Providing stronger authentication at a low-cost to RFID tags operating under the EPCglobal framework", IEEE/IFIP International Conference on Embedded and Ubiquitous Computing Conference, Vol. 2, pp. 159–167, 2008.

[51] A. Moradi, A. Poschmann, "Lightweight cryptography and DPA countermeasures: a survey", Proceedings of the 14th International Conference on Financial Cryptography and Data Security, pp. 68–79, 2010.

[52] C. Paar, A. Poschmann, M.J.B. Robshaw, "New designs in lightweight symmetric encryption", In: RFID Security: Techniques, Protocols and System-on-Chip Design, Springer, pp. 349–371, 2009.

[53] M. Feldhofer, S. Dominikus, J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm", Cryptographic Hardware and Embedded Systems, Vol. 3156, pp. 357–370, 2004.

[54] M. Feldhofer, J. Wolkerstorfer, "Hardware implementation of symmetric algorithms for RFID security", In: RFID Security: Techniques, Protocols and System-on-Chip Design, Springer, pp. 373–415, 2009.

[55] P.P. Lopez, J.C.H. Castro, J.E. Tapiador, A. Ribagorda, "An efficient authentication protocol for RFID systems resistant to active attacks", Conference on Emerging Direction in Embedded and Ubiquitous Computing, pp.781–794, 2007.