Yet Another Ultralightweight Authentication Protocol That Is Broken

Gildas Avoine and Xavier Carpent

Université catholique de Louvain
B-1348 Louvain-la-Neuve
Belgium

Abstract. Eghdamian and Samsudin published at ICIEIS 2011 an ultralightweight mutual authentication protocol that requires few bitwise operations. The simplicity of the design makes the protocol very suitable to low-cost RFID tags. However, we demonstrate in this paper that the long-term key shared by the reader and the tag can be recovered by an adversary with a few eavesdropped sessions only.

Additionally, we provide the backbone of some attacks on a series of similar recent protocols, and highlight important common weaknesses in the design of ultralightweight protocols.

Keywords: Authentication, Ultralightweight protocol, RFID.

1 Introduction

The market pressure to lower the price of tags is such that it has become a major topic of research to design an RFID protocol requiring very few gates and little computational power on the tag side. Several families of protocols have been proposed, such as the influential HB family (see [5] for a thorough presentation of the HB family), and other "human authentication" protocols. In [11], Peris-Lopez, Hernandez-Castro, Estevez-Tapiador, and Ribagorda introduced a mutual protocol, called LMAP, which is the first of what came to be known as the "ultralightweight protocols family". Many proposals followed (see [2] for a comprehensive introduction to this protocol family), but almost all of them have been broken. These protocols rely on very simple building blocks, such as bitwise operations ($\oplus$,$\vee$, $\wedge$), modular addition ($+$), or data-dependent rotations ($\mathrm{Rot}(x, y)$). They often do not require the tag to generate randomness, and require tags to update their state every successful authentication.

Recently, Eghdamian and Samsudin proposed a new protocol in that family, claiming more security than its predecessors. We show in this paper how a passive attack can recover the 96-bit secret of a tag, using only 20 authentication sessions on average.

We also show similar attacks on RPAP (by Ning, Liu and Yang [10]), PUMAP (by Bassil, El-Beaino, Itani, Kayssi and Chehab [4]), and DIDRFID and SIDFRID (by Lee [9]). We finally point out traceability attacks on RAPP (by Tian, Chen and Li [13]), and Improved LMAP+ (by Gurubani, Thakkar and Patel [8]).

J.-H. Hoepman and I. Verbauwhede (Eds.): RFIDSec 2012, LNCS 7739, pp. 20–30, 2013.

The highlighted attacks show once more that most of the protocols of this class can be broken with little effort.

The paper is divided as follows. In Sect. 2, we present Eghdamian and Samsudin's protocol. Our attack on it is thoroughly described in Sect. 3. In Sect. 4, we briefly describe a series of other ultralightweight protocols and miscellaneous attacks on them. We highlight some common weaknesses in the design of ultralightweight protocols. We finally conclude in Sect. 6.

2 Eghdamian and Samsudin's Protocol

The protocol designed by Eghdamian and Samsudin [7] consists of four messages, represented on Fig. 1. First of all, the reader sends an hello message, then the

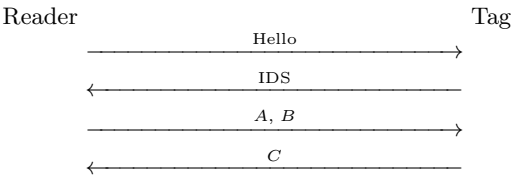

Fig. 1. Eghdamian and Samsudin's Protocol

tag sends its IDS. This IDS allows the reader to identify the tag and find the corresponding key K. If this identification step fails for some reason (error on the channel, tag not synchronized, false IDS), the reader sends a second request, to which the tag responds the old value of IDS. After the identification stage, the reader sends A and B, and the tag C. The content of A, B, and C is as follows:

$$A = K \oplus N \tag{1}$$

$$B = \mathrm{Rot}(K, N) \wedge \mathrm{Rot}(N, K) \wedge \mathrm{Rot}(N, N) \tag{2}$$

$$C = \mathrm{Rot}(K + \mathrm{Rot}(N, N), \mathrm{Rot}(K, K) \vee N) \tag{3}$$

where $\mathrm{Rot}(X, Y)$ means that X is rotated of $\mathcal{H}(Y)$ bits to the left, where $\mathcal{H}(Y)$ denotes the Hamming weight of Y. The symbol N represents a random value. After a successful authentication, the tag updates its key and session identifier as follows:

$$K^{next} = \mathrm{Rot}(N + \mathrm{Rot}(K, K), \mathrm{Rot}(N, N) \wedge K) \tag{4}$$

$$\mathrm{IDS}^{next} = K \wedge \mathrm{Rot}(N, K \vee N) \tag{5}$$

Let L denote the length of all the variables (recommended to be 96 in [7]):

$$|K| = |N| = |A| = |B| = |C| = |IDS| = L.$$

3 Attack on Eghdamian and Samsudin's Protocol

We introduce in this section a key-recovery attack that allows an adversary to recover the key K shared by the reader and the tag. The attack requires a passive adversary to eavesdrop one authentication session where a property on the Hamming weight of N is ensured, as detailed below. If the adversary is active and knows the current IDS of her target, she can perform her attack without the presence of the targeted tag.

3.1 Discovering the Hamming Weight of N

The first step of the attack aims to recover $\mathcal{H}(N)$. Below B_i denotes the bit at index i of B, with B_0 being the least significant bit of B. From Eq. (2), we know that:

$$\forall i, 0 \leq i < L, (B_i = 1) \Rightarrow (K_{i-\mathcal{H}(N) \bmod L} = N_{i-\mathcal{H}(N) \bmod L} = 1).$$

Using Eq (1), we deduce:

$$\forall i, 0 \leq i < L, (B_i = 1) \Rightarrow (A_{i-\mathcal{H}(N) \bmod L} = 0). \tag{6}$$

Consequently, a candidate r for $\mathcal{H}(N)$ is discarded if Eq (6) is not satisfied. If only one candidate r among the n possible ones remains, then $\mathcal{H}(N) = r$. Experimentally, we observed that this case occurs with a probability close to 0.9 when $L = 96$. When more that one candidate remain, the adversary can keep the few candidates and discard the wrong ones later in the attack, or she can simply eavesdrop another authentication session in order to be luckier and obtain a single candidate.
We consider from now on that the adversary knows $\mathcal{H}(N)$.

3.2 Recovering Half of the Secret Bits

The adversary assumes that $\mathcal{H}(K) = \mathcal{H}(N)$. This assumption will be denoted H_1 in the following. Whenever H_1 is true, Eq (2) yields:

$$B = \mathrm{Rot}(K, N) \wedge \mathrm{Rot}(N, N),$$

and so:

$$\mathrm{Rot}^{-1}(B, N) = K \wedge N. \tag{7}$$

where Rot^{-1} means the right-rotation. We will denote below:

$$\widetilde{B} := \mathrm{Rot}^{-1}(B, N).$$

From Eq (1), we know that $A_i = 0$ implies that either $K_i = N_i = 0$ or $K_i = N_i = 1$. Consequently:

$$\forall i, 0 \leq i < L, (A_i = 0) \Rightarrow (K_i = \widetilde{B}_i).$$

This technique allows the adversary to recover half of the secret bits on average. Given that $\mathcal{H}$ follows a binomial distribution, Vandermonde's identity allows to demonstrate that the assumption H_1 actually occurs with probability $\binom{2L}{L}/2^{2L}$. When $L = 96$, this value is close to 0.058, which implies that the adversary should eavesdrop about 18 authentication sessions on average in order to observe one where the property $\mathcal{H}(N) = \mathcal{H}(K)$ is satisfied.

3.3 Recovering More Secret Bits

The adversary can increase the number of revealed bits of the secret key by exploiting the IDS following the session where H_1 is satisfied. Indeed, we know from Eq (5) that:

$$\mathrm{IDS}^{next} = K \wedge \mathrm{Rot}(N, K \vee N).$$

We conclude that

$$\forall i, 0 \leq i < L, \left(\mathrm{IDS}_i^{next} = 1\right) \Rightarrow \left(K_i = 1\right). \tag{8}$$

3.4 Recovering Still More Secret Bits

Once some bits of K and N are known, the adversary can exploit them to recover more bits of K. For that, we can first trivially notice that:

$$K \vee N = (K \wedge N) \vee (K \oplus N). \tag{9}$$

When H_1 holds, we deduce, by inserting Eq (1) and Eq (7) in Eq (9):

$$K \vee N = A \vee \widetilde{B}. \tag{10}$$

Therefore, Eq (5) can be rewritten using Eq (10) as:

$$\mathrm{IDS}^{next} = K \wedge \mathrm{Rot}(N, A \vee \widetilde{B}). \tag{11}$$

If the adversary already knows i such that $K_i = 1$ then using Eq (1) and Eq (11), we deduce:

$$K_{i-\mathcal{H}(A \vee \widetilde{B})} = A_{i-\mathcal{H}(A \vee \widetilde{B})} \oplus \mathrm{IDS}_i^{next}. \tag{12}$$

Likewise, if the adversary already knows i such that $K_{i-\mathcal{H}(A \vee \widetilde{B})} \oplus A_{i-\mathcal{H}(A \vee \widetilde{B})} = 1$ then using Eq (1) and Eq (11), we deduce:

$$K_i = \mathrm{IDS}_i^{next}. \tag{13}$$

These two last steps can further be iterated a few times, until no more information can be gathered. At that point, most of the bits of K are known. We have observed experimentally that an average of 73 bits of K are discovered.

3.5 Recovering the Remaining Secret Bits with a Passive Adversary

If the adversary is passive, she can recover the remaining secret bits performing a reasonable exhaustive search on the 23 unknown bits (on average). Candidates can be tested on C and B. If no suitable candidate is found in the exhaustive search, then the hypothesis $\mathcal{H}(K) = \mathcal{H}(N)$ was wrong, and another authentication attempt must be eavesdropped on.

3.6 Recovering the Remaining Secret Bits with an Active Adversary

An active adversary can block the message C in order to cancel the update on the reader side, and thus force the tag to use the same IDS and K in the following session. This allows her to collect A, B, C messages for the same K, but different N, and therefore guess all the bits of K, with no exhaustive search required.

4 Attacks on Other Protocols

In this section, several privacy and key-recovery attacks on a series of recent similar protocols are introduced. The protocols are not fully described but, instead, the key-points in their design that open the door for an attack are highlighted.

4.1 Ning, Liu and Yang's Protocol

RPAP was proposed by Ning, Liu and Yang in [10]. The main novelty is that the secret between the reader and a tag is partitioned into three sub-secrets, and the way the partition is done depends on a parameter d chosen and sent by the reader. The secret S is partitioned such that:

$$\begin{aligned} S_1 &= [S]_{L-d:L-1} \\ S_2 &= [S]_{d:L-d-1} \\ S_3 &= [S]_{0:d-1}, \end{aligned}$$

with $[x]_{a:b}$ denoting the number comprised of bits of x from a to b. The sub-secrets are 0-padded on the most significant bits when appropriate. Note that no information was given in [10] regarding how the reader should choose d, other than ranging from 1 to $L/2$.

A first important weakness is the way the message D (sent by the tag) is designed:

$$D = (S'_1 \vee S'_2) \oplus S_3,$$

with S'_1 and S'_2 defined as:

$$\begin{aligned} S'_1 &= \mathrm{Rot}(S_1, r_1, d) \\ S'_2 &= \mathrm{Rot}(S_2, r_2, d) \end{aligned}$$

A good estimator for each bit of S_3 is $\overline{D}$ (with probability of 3/4). In a handful of runs, an eavesdropper can thus easily recover the lower half of S.

There are other weaknesses in the design of the protocol that can help an attacker discover most of the secret S in few protocol runs. For instance, after recovering partially S, an adversary knows $(S'_1 \vee S'_2)$ from D. Therefore, when $[D \oplus S_3]_i = 0$, then $[S'_1]_i = [S'_2]_i = 0$ (where $[x]_i$ denotes the i-th bit of x), and when $[D \oplus S_3]_i = 1$, then $[S'_1]_i = [S'_2]_i = 1$ with probability 2/3. This gives further information on $S_1 \oplus r_1$ and $S_2 \oplus r_2$, which, in conjunction with other weaknesses gives information on S_1 and S_2.

One such other weakness lies in the message A (sent by the reader) which is built as:

$$A = (IDS_T \vee S_1) \oplus r_1.$$

Since IDS_T is public (it plays the same role as IDS in [7]), the adversary can easily get half of the bits of r_1, and the other half of $S_1 \oplus r_1$ on average at each run. The construction of B (sent by the reader) is also weak:

$$B = IDS_T \oplus (S_2 + r_2).$$

Here, IDS_T is essentially useless since public, and the adversary gets $S_2 + r_2$ trivially.

While all these issues are not important on their own (except the first one), they are very dangerous when considered together, and allow an eavesdropper to recover most bits of S in a few runs.

4.2 Bassil, El-Beaino, Itani, Kayssi and Chehab's Protocol

Bassil, El-Beaino, Itani, Kayssi and Chehab proposed in [4] a new authentication for RFID using PUF's (physically unclonable functions), called PUMAP. Regardless of the use of PUF's, the protocol uses constructions that are similar to other ultralightweight authentication protocols.

PUMAP follows the same scheme as Eghdamian and Samsudin's Protocol (see Fig. 1). The reader sends messages A,B and C to the tag, which are defined as follows:

$$\begin{aligned} A &= SVT \oplus SVR \oplus n_1 \\ B &= \mathrm{Rot}(SVR + n_2, SVT) \\ C &= \mathrm{Rot}(SVT \oplus SVR \oplus n_1, n_2), \end{aligned}$$

where $\mathrm{Rot}(X, Y)$ here means that X is rotated by $(Y \bmod L)$ bits to the left, and SVT and SVR are essentially the analogues of respectively IDS and K in [7]. The former is thus public, the latter secret. The other values are nonces.

The first attack we suggest is an active desynchronization one. Note that C is simply $\mathrm{Rot}(A, n_2)$. This means that an adversary has a probability of $1/L$ of forging a valid (i.e., one accepted by the tag) triplet (A, B, C) if she just sends a triplet (X, Y, X) with X and Y being arbitrary values. When receiving one such triplet, a tag updates SVT and SVR, and desynchronizes with the system.

An adversary just has to keep sending forged triplets until one is accepted. She needs to do this L times on average.

The second attack allows an eavesdropper to guess the next SVR at each run, and thus to trace and/or impersonate a tag. The messages sent by the tag after receiving A, B and C are defined as[1]:

$$\begin{aligned} D &= \mathrm{Rot}(\mathrm{Rot}(n_1 + (n_2 \oplus SVT) + SVR, n_2), n_1) \\ E &= \mathrm{Rot}(SVT^{next} \oplus n_2, n_1) \\ F &= \mathrm{Rot}(SVR^{next} \oplus n_1, n_2) \end{aligned}$$

Note that there are only L possibilities for the rotation in E. An eavesdropper getting SVT^{next} on the next session (or skimming the tag) thus has L candidates for n_2. Using B and then A, she gets the corresponding candidates for SVR and n_1. These candidate triplets (n_2, SVR, n_1) can then be tested against D. Once a correct set of values has been found, SVR^{next} can be obtained from F and n_1.

4.3 DIDRFID and SIDRFID

In [9], Lee presents two new ultralightweight authentication protocols, DIDRFID and SIDRFID. We present a full key-recovery attack on each of them. Rotations are used in both protocols, and use the Hamming weight of the second argument, much like the ones in [7].

The equations relevant for the attack in DIDRFID are the following:

$$\begin{aligned} A &= K \oplus R \\ DIDT^{next} &= \mathrm{Rot}(R, R \vee K) \oplus \mathrm{Rot}(K, R \wedge K) \\ K^{next} &= \mathrm{Rot}(R, R \wedge K) \oplus \mathrm{Rot}(K, R \vee K), \end{aligned}$$

where $DIDT$ is the equivalent of IDS in [7], K is the secret key, and R is a nonce. We thus have that

$$DIDT^{next} \oplus K^{next} = \mathrm{Rot}(A, R \vee K) \oplus \mathrm{Rot}(A, R \wedge K).$$

There are thus L^2 possibilities for K^{next}, which can be tested on the next session. Moreover, given the biased nature of the rotations, and given that the rotations are using Hamming weights, an eavesdropper usually needs much less than L^2 guesses. An eavesdropper thus gets the whole key of a tag by simply listening to one protocol run.

We will not detail SIDRFID, because the protocol uses a master key in the tag. This solution is dangerous because an adversary, after compromising a single tag, obtains this master key. She can then impersonate any tag in the system after eavesdropping one single protocol run with her victim.

[1] Note that there is an unmatched bracket for D in [4], but both attacks work regardless.

4.4 RAPP

Tian, Chen and Li introduce in [13] a new building block for ultralightweight protocols, as well as a new protocol using it, called RAPP. The new operator is called the permutation Per. We do not cover its definition here and refer the interested reader to the original paper. However, one bad feature of this construction, as pointed by the authors, is that it is Hamming weight-invariant (much like the rotations). We provide a traceability attack that highlight the weakness of this new operator and the design of RAPP.

The relevant equations are:

$$\begin{aligned} A &= \mathrm{Per}(K_2, K_1) \oplus n_1 \\ C &= \mathrm{Per}(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID \\ K_1^{next} &= \mathrm{Per}(K_1, n_1) \oplus K_2 \\ K_2^{next} &= \mathrm{Per}(K_2, n_2) \oplus K_1, \end{aligned}$$

where n_1 is a nonce. We point out the following fact:

$$\mathcal{H}(x \oplus y) = \mathcal{H}(x) + \mathcal{H}(y) - 2\mathcal{H}(x \wedge y),$$

for any x, y. As a corollary, we have that

$$\mathcal{H}(x \oplus y) \equiv \mathcal{H}(x) \oplus \mathcal{H}(y) \pmod 2.$$

This result has the following implications in RAPP:

$$\begin{aligned} \mathcal{H}(K_1^{next}) &\equiv \mathcal{H}(K_1) \oplus \mathcal{H}(K_2) \pmod 2 \\ \mathcal{H}(K_2^{next}) &\equiv \mathcal{H}(K_2) \oplus \mathcal{H}(K_1) \pmod 2. \end{aligned}$$

This implies that, after the very first run of the protocol, we have that $\mathcal{H}(K_1) \equiv \mathcal{H}(K_2) \equiv 0 \pmod 2$. Furthermore,

$$\begin{aligned} \mathcal{H}(A) &\equiv \mathcal{H}(K_2) \oplus \mathcal{H}(n_1) \pmod 2 \\ \mathcal{H}(C) &\equiv \mathcal{H}(n_1) \oplus \mathcal{H}(K_1) \oplus \mathcal{H}(ID) \pmod 2. \end{aligned}$$

An eavesdropper therefore gets easily that $\mathcal{H}(ID) \equiv \mathcal{H}(A) \oplus \mathcal{H}(C) \pmod 2$. This allows her to trace a tag.

4.5 Improved LMAP+

In [8], Gurubani, Thakkar and Patel propose an improved version of LMAP+, itself an extension of LMAP [11]. The improved LMAP+ is supposed to guarantee untraceability, but we show that this is not the case.

The messages in Improved LMAP+ are the following:

$$\begin{aligned} A &= (PID \oplus K_1) + r \\ B &= PID + K_2 + r \\ C &= PID \oplus (K_3 + r) \\ PID^{next} &= (PID \oplus r) + K_1 + K_2 + K_3 \\ K_1^{next} &= (K_1 \oplus r) + PID^{next} + K_2 \\ K_2^{next} &= (K_2 \oplus r) + PID^{next} + K_3 \\ K_3^{next} &= (K_3 \oplus r) + PID^{next} + K_1, \end{aligned}$$

where PID plays the same role as IDS in [7] and r is a nonce. A very natural thing to do when analyzing messages containing both XOR's and modular additions such as these is to look at the least significant bit (LSB) position (transforming the sums in XOR's). This allows to note that:

$$PID^{(n+2)} = r^{(n)} \oplus r^{(n+1)},$$

where the notation at the exponent is used to denote the value of that variable at a given protocol run. An eavesdropper can thus get the LSB of each nonce by making a single hypothesis on an initial value. The LSB of the keys can then be obtained using:

$$\begin{aligned} \mathrm{lsb}(K_1) &= \mathrm{lsb}(A \oplus PID \oplus r) \\ \mathrm{lsb}(K_2) &= \mathrm{lsb}(B \oplus PID \oplus r) \\ \mathrm{lsb}(K_3) &= \mathrm{lsb}(C \oplus PID \oplus r) \\ \mathrm{lsb}(K_1^{next}) &= \mathrm{lsb}(K_1 \oplus r \oplus PID^{next} \oplus K_2) \\ \mathrm{lsb}(K_2^{next}) &= \mathrm{lsb}(K_2 \oplus r \oplus PID^{next} \oplus K_3) \\ \mathrm{lsb}(K_3^{next}) &= \mathrm{lsb}(K_3 \oplus r \oplus PID^{next} \oplus K_1), \end{aligned}$$

which allows an eavesdropper to trace a tag. Although this has not been verified, we believe a full recovery attack could also be done using the same technique as the attack on LMAP by Bárász, Boros, Ligeti, Lója and Nagy [3], that is, further guess the bit just after the LSB, than the one after that, and so on.

5 Discussion on Weaknesses

From the weaknesses exploited in this paper, we can highlight some weak constructions.

The use of biased operations such as OR ($\vee$) and AND ($\wedge$) has often led to vulnerabilities (see [1,3] for instance, as well as the attacks presented in this paper). Although they bring non-linearity, and seem good when combined to other types of operations, an attacker may exploit the bias when used on their own, or weakly "shielded".

The combined use of modular additions ($+$) and XOR ($\oplus$) seems good, but it has been proved to be weak in some cases (see the attack on LMAP [3] for instance). One major point is that the modular addition is a XOR in the least significant bit, and the leakage of this bit is enough for performing a privacy attack. Moreover, if an adversary knows the least significant bit of the operands, the second bit can usually be guessed as well and so on. It has also been shown that when the operands are biased or partially known, information can be gathered on their sum the same way it can be done with their XOR ([2]).

Data-dependent rotations have been allegedly first used for RFID protocols in SASI ([6]), and are since then often part of the building blocks used in ultralightweight protocols (either using the modular or the Hamming weight version). It has been shown repeatedly that although they bring non-linearity at a cheap cost, they are dangerous if carelessly used. The output only has L possible outcomes, which makes guessing and trying an easy task.

Operations affecting the Hamming weight (such as OR and XOR) or other external measures are sometimes problematic. On the contrary, some operations such as rotations and permutations from [13] are Hamming weight-preserving, which allows an adversary to guess some information on the operands, allowing traceability for instance.

Using public messages in the construction of others has sometimes little to no cryptographic use. This is particularly the case for IDS. Since this information is public, the adversary has access to it and can reverse the operations (provided these are reversible).

Symmetry, although appealing, sometimes allows simplifications in the protocol messages and eases the task of an attacker. Notable examples include the attack of Peris-Lopez, Hernandez-Castro, Estevez-Tapiador and Van der Lubbe on Lee, Hsieh, You and Chen's protocol ([12]) and the attack on DIDRFID presented in this paper.

6 Conclusion

We have shown in this paper that Eghdamian and Samsudin's ultralightweight protocol is not secure, since a passive adversary can recover the key of a tag in an average of 20 authentication sessions. Although this number depends on L, the attack remains very efficient, even for bigger values of L than the recommended 96.

We also show key-recovery attacks on RPAP [10], PUMAP [4], DIDRFID [9] and SIDFRID [9], as well as traceability attacks on RAPP [13], and Improved LMAP+ [8].

These attacks are an additional example of the lack of security of ultralightweight protocols, and they question the relevance of this approach to design authentication protocols for RFID.

References

1. Avoine, G., Carpent, X., Martin, B.: Strong Authentication and Strong Integrity (SASI) Is Not That Strong. In: Ors Yalcin, S.B. (ed.) RFIDSec 2010. LNCS, vol. 6370, pp. 50–64. Springer, Heidelberg (2010)
2. Avoine, G., Carpent, X., Martin, B.: Privacy-friendly synchronized ultralightweight authentication protocols in the storm. Journal of Network and Computer Applications 35(2), 826–843 (2012)
3. Bárász, M., Boros, B., Ligeti, P., Lója, K., Nagy, D.: Breaking LMAP. In: Conference on RFID Security, Malaga, Spain (July 2007)
4. Bassil, R., El-Beaino, W., Itani, W., Kayssi, A., Chehab, A.: PUMAP: A PUF-based ultra-lightweight mutual-authentication RFID protocol. International Journal of RFID Security and Cryptography 1(1), 58–66 (2012)
5. Bosley, C., Haralambiev, K., Nicolosi, A.: HB^N: An HB-like protocol secure against man-in-the-middle attacks. Cryptology ePrint Archive, Report 2011/350 (2011)
6. 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)
7. Eghdamian, A., Samsudin, A.: A Secure Protocol for Ultralightweight Radio Frequency Identification (RFID) Tags. In: Abd Manaf, A., Zeki, A., Zamani, M., Chuprat, S., El-Qawasmeh, E. (eds.) ICIEIS 2011. CCIS, vol. 251, pp. 200–213. Springer, Heidelberg (2011)
8. Gurubani, J.B., Thakkar, H., Patel, D.R.: Improvements over Extended LMAP+: RFID Authentication Protocol. In: Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H. (eds.) IFIPTM 2012. IFIP AICT, vol. 374, pp. 225–231. Springer, Heidelberg (2012)
9. Lee, Y.-C.: Two ultralightweight authentication protocols for low-cost RFID tags. Applied Mathematics and Information Sciences 6(2S), 425–431 (2012)
10. Ning, H., Liu, H., Yang, C.: Ultralightweight RFID authentication protocol based on random partitions of pseudorandom identifier and pre-shared secret value. Chinese Journal of Electronics 20(4), 701–707 (2011)
11. 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: Workshop on RFID Security – RFIDSec 2006, Graz, Austria (July 2006); Ecrypt
12. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., van der Lubbe, J.C.A.: Security Flaws in a Recent Ultralightweight RFID Protocol. In: Workshop on RFID Security – RFIDSec Asia 2010. Cryptology and Information Security, vol. 4, pp. 83–93. IOS Press, Singapore (2010)
13. Tian, Y., Chen, G., Li, J.: A new ultralightweight RFID authentication protocol with permutation. IEEE Communications Letters 16(5), 702–705 (2012)