Ultra-lightweight Mutual Authentication Protocols : Weaknesses and Countermeasures

Zeeshan Bilal
Information Security Group
Royal Holloway University of London
Egham, Surrey, TW20 0EX
Zeeshan.Bilal.2010@live.rhul.ac.uk

Keith Martin
Information Security Group
Royal Holloway University of London
Egham, Surrey, TW20 0EX
Keith.Martin@rhul.ac.uk

***Abstract*—This paper reviews weaknesses highlighted in existing proposals for a family of mutual authentication protocols belonging to the ultra-lightweight class, which are designed for low-cost RFID systems. This family is suitable for systems where authenticating parties already share secrets, which are updated in each authentication round to counter tracking of the tag. We propose a new ultra-lightweight authentication protocol that builds on the strengths of existing schemes yet incorporates countermeasures to overcome previous weaknesses. Significantly our protocol uses lower resources than previous proposals.**

***Keywords*-Ultra-lightweight; Mutual Authentication Protocol; RFID**

I. Introduction

Radio Frequency Identification (RFID) systems consist of three main components: *server*, *reader* and *tag*. The communication channel between server and reader is assumed to be secure while the channel between reader and tag is insecure. The wide deployment of RFID systems is being constrained due to many security and privacy issues [1] concerning the channel between reader and tag.

To secure this channel researchers have proposed various cryptographic solutions, including mutual authentication protocols between the two communicating parties. Based on the computational cost and operations supported by the tags, these authentication protocols are divided into four classes: *full-fledged*, *simple*, *lightweight* and *ultra-lightweight* [2]. The ultra-lightweight class is proposed for the low-cost RFID systems that are most widely deployed and most likely to replace bar-codes. The main limiting factor in these tags is the various resource constraints. Since cost has to be kept low, these tags cannot afford a state-of-the-art CPU, large memory, or to support large bandwidth. Generally, low-cost RFID tags consist of a few thousand gates, a simple Arithmetic and Logic Unit (ALU) performing simple operations, and no active power source.

A. Ultra-lightweight Mutual Authentication Protocols

Ultra-lightweight mutual authentication protocols (UMAPs) are designed to provide mutual authentication between a reader and a tag. These schemes are proposed for low-cost RFID tags having a total of 5,000-10,000 logic gates. Out of this total, less than 3,000 logic gates can be used to implement the authentication protocol [2]. Many UMAPs have been proposed, but all existing proposals have significant flaws, as we now outline.

Initial proposals were based only on the use of triangular functions (T-Functions) [3] including XOR, AND, OR and $addition\ modulo\ 2^L$. These schemes require fewer logic gates (as low as 3000) and are considered very efficient [4]–[6]. However, T-functions have very poor diffusion and use of AND and OR produces biased results. Weaknesses in these proposals were highlighted soon after publication [7]–[10].

SASI [2] is the first UMAP to use a lightweight non-triangular function $RotBits$ (left rotation of bits) with triangular functions. This protocol was initially acclaimed but later weaknesses were highlighted in its design which resulted in de-synchronization and full-disclosure attacks [11]–[13]. A full-disclosure attack on $SASI$ in [11] used the properties of $RotBits$ function. The main assumption of the attack was that if $RotBits$ does not rotate the values, $SASI$ can be treated as a scheme with T-functions only. As a result, the $Gossamer$ [14] UMAP was proposed which introduced a new non-triangular lightweight function known as $MixBits$. However, the weakness which caused de-synchronization in $SASI$ [12], [13] was not addressed in $Gossamer$, resulting in further de-synchronization attacks [15]–[17]. This weakness arises since the reader generates new random numbers in each authentication round and both reader and tag use these random numbers to update their values. An adversary can thus use this property to its advantage by de-synchronizing the authentication process [18].

David et al. [19] and Tagra et al. [17] proposed countermeasures to prevent de-synchronization attacks. However, these countermeasures require additional valuable resources at the tag end. Hernandez-Castro et al. [20] presented a passive attack on the scheme in [19] which can recover a tag's secret using linear cryptanalysis. Lee et al. [21] also presented a scheme which requires additional memory and communication overheads and has some privacy issues [22]. Yeh et al. [16] suggested reducing the storage overheads on the tag's memory, however, it is very easy to force de-synchronization [18]. Moreover, a passive adversary can carry out a traceability and a full-disclosure attack on the Yeh et al. protocol [23]. Similarly a protocol suggested by Eghdamian et al. [24] is cryptanalyzed by Avoine in [25]. In most UMAPs, the main vulnerability

exploited by an adversary is the stateless nature of a tag. The attacker runs many incomplete protocols and gathers information from each in order to disclose secret values.

Most of the existing proposals for UMAPs thus have flaws. The existing countermeasures to overcome these flaws given in [17], [26] have notable overheads. In this paper, we try to rectify this by proposing a new ultra-lightweight protocol that builds on the strengths of previous designs while overcoming their weaknesses using fewer resources. Our UMAP is explained in detail in Section II. We then carry out a security and performance analysis in Section III.

II. Proposed Protocol

We now explain our proposed UMAP.

A. Assumptions

We first make the following assumptions that must hold prior to running our protocol.

- Each tag shares secrets (specifically a key and static identity) with the server.
- The server holds a database which records details about a particular tag, including its shared secrets (key and static identity).
- This database is indexed by a dynamic and publicly known index-pseudonym unique to each tag.
- The reader is an intermediary which relays the messages from the tag (prover) to the server (verifier).
- The reader, querying the tag, is connected to the server and is legitimate (the communication channel between the reader and server is secured).

B. Adversarial Model

We consider that our scheme is vulnerable to both passive and active attackers. The abilities and limitations of our potential adversary are as follows:

- The adversary is capable of listening to both forward and backwards channels (the reader to the tag and vice versa).
- We assume that our adversary has two options: either to jam the conversation between the legitimate server and tag (active) or to eavesdrop (passive). However we also assume that our adversary cannot function in full duplex mode i.e. she cannot transmit and receive on the same frequency slot, at the same time.
- The adversary cannot take over an ongoing authentication round because when the tag detects a collision of readers, it stops responding (we assume the use of reader collision algorithm, see [27]).
- Defenses against relay attack (man-in-the-middle), physical capture and tampering are not in the scope of this paper.

The notation in our scheme is shown in Table I.

Table I
Notation

Notation	Description
T	The tag participating in an authentication round.
R	The reader participating in an authentication round.
S	The server holding the database and authenticating a tag.
Adv	Both passive as well as active adversary.
$Index^i$	A dynamic index-pseudonym uniquely associated to each tag in i^{th} authentication round.
KS^i	A dynamic secret key shared between the tag and server in i^{th} authentication round.
ID	Tag's static and unique identity.
L	Length of the secret key and static identity.
r^i	Random number generated by server in i^{th} authentication round.
$+$	Addition modulo 2^L as all values are assumed to be L-bit long.
$A \rightarrow B : M$	A sends to B, public message M.
HW(X)	Hamming weight of bit string X.
$\lambda(X)$	Integer value of L-bit string X reduced modulo L.
$LRot(X, \mu)$	Left rotation of argument X by μ.
$f(X, Y)$	A secure lightweight pseudo random function (PRF) which takes two inputs X, Y and outputs a pseudo-random value where $f(X, Y) \neq f(Y, X)$ (such as $MixBits$ specified in [14]).

C. Goals

A UMAP should achieve the following goals considering the variety of potential threats (details are given in [1]):

- **Mutual Authentication:** Our scheme should provide mutual (entity) authentication.
- **Tag Content Privacy:** The secret static identity of the tag should not be transmitted in the clear as it is linked to the contents of the item it is attached with.
- **Availability:** Authenticating parties should stay synchronized and always be available to communicate.
- **Tag Anonymity:** The adversary should not be able to track a target tag by listening to the channel.
- **Forward Security:** If a tag is compromised at any stage, the adversary should not be able to compromise any future communication.
- **Performance:** Since UMAPs are designed for low-cost RFID systems:
  - storage space should be as low as possible,
  - cryptographic functions should be extremely lightweight in nature and efficient to compute,
  - the amount of data communicated should be kept as low as possible.

D. Design Features

Our protocol has the following design features, intended to overcome flaws as outlined in Section I-A:

1) **Combination of Functions:** The protocol uses a combination of lightweight non-triangular functions and triangular functions. We use $LRot(X, \lambda(Y))$ as left rotation of bit string X by $\lambda(Y)$ positions, where $\lambda(Y)$ is computed by first converting the L-bit string Y into an integer and then reducing it modulo L. Some of the existing schemes have used $Rot(X, HW(Y))$ as left rotation of bit string X by

the hamming weight of bit string Y. Since HW(Y) does not follow a uniform distribution, this weakens the security property given by the rotation function, whereas $\lambda(Y)$ follows a uniform distribution.

2) **Use of Random Nonce:** In our protocol, the server generates a random nonce for data freshness. In existing schemes random nonces change on every communication attempt even in event of a failed authentication. Avoine et al. [18] mention this as a potential vulnerability which can lead to de-synchronization attacks. Our scheme overcomes this vulnerability by recording each random nonce in a database for a particular tag's $Index$. It then uses the random value to calculate internal secret values for updating tuple (KS, $Index$). The server generates a new random nonce only after a successful authentication. This resists de-synchronization attacks and provide tag anonymity and forward security.

3) **Provision for Re-synchronization:** In event of a failed authentication attempt either due to communication error or intentional interference by an adversary, both server and tag may become de-synchronized. Our scheme re-synchronizes as the tag does not update its values in a failed authentication attempt and the server keeps a copy of older values. In some existing schemes [2], [14], [17], [19], [24], [26], [28], re-synchronization is attempted using older values of $Index$ and shared secrets stored in the tag's memory. This not only causes additional overheads on the tag's valuable memory but also leads to a potential weakness which allows the server to ask for older values of $Index$ of the tag if the updated $Index$ is not recognized. This weakness leads to denial-of-service attacks as mentioned in [15], [18]. In our scheme, if the server asks for older values, this will be an indication of a replay attack carried out by an impersonating server.

4) **Cost, Performance and Security Trade-offs:** Our scheme provides a trade-off between cost, performance and level of security. It uses lightweight functions which can be easily incorporated in the simple ALU of low-cost RFID tags. Our protocol consumes a small amount of storage on these tags and completes the protocol using two messages. The schemes mentioned in [17], [19], [24], [26], [28], [29] require additional messages and memory requirements in order to overcome existing weaknesses. Moreover, many of these schemes are still vulnerable and have been analyzed to highlight weaknesses in the design [18], [20], [23], [25], [30].

E. The Protocol

We now propose a new UMAP that provides the security goals mentioned in Section II-C and has the design features identified in Section II-D.

1) Identification Stage: A compatible T in the vicinity of a compatible R is identified as follows:

- **Step 1.** $R \rightarrow T : Hello$
- **Step 2.** $T \rightarrow R : Index^i$
- **Step 3.** $R \rightarrow S : Index^i$
- **Step 4.** S now searches for this $Index^i$ in its database. If it matches an existing entry, S proceeds to the next stage, otherwise it does not respond to T.

2) Server Authentication and Update Stage: On successful identification, S is authenticated as follows:

- **Step 1.** S uses $Index^i$ sent by T to extract KS^i associated with this particular T.
- **Step 2.** S now generates a random value r^i and calculates the internal secret values n_1^i,n_2^i using tuple (KS^i, r^i) as follows:

$$\begin{aligned} n_1^i &= f(KS^i, r^i), \\ n_2^i &= f(r^i, KS^i). \end{aligned} \tag{1}$$

- **Step 3.** S now generates public message A^i using tuple $(n1^i, n2^i, Index^i, KS^i, ID)$ as follows:

$$A^i = LRot(LRot(n2^i + Index^i + KS^i + ID, n1^i) + n1^i, n2^i). \tag{2}$$

- **Step 4.** $S \rightarrow R : A^i \| r^i$
- **Step 5.** $R \rightarrow T : A^i \| r^i$
- **Step 6.** T calculates internal secrets $n1^i$ and $n2^i$ as in (1) and uses these to calculate a local copy $A^{i'}$ of A^i using (2).
- **Step 7.** T now checks:
  if $A^{i'} = A^i$ **then**
  S is authenticated; proceed to next stage;
  else
  Protocol is abandoned;
  end if
- **Step 8.** S after sending $A^i \| r^i$ also updates its tuple $(Index^i, KS^i)$ as follows:

$$\begin{aligned} Index^{i+1} &= LRot(LRot(n1^i + Index^i, n1^i) + n2^i, n2^i), \\ KS^{i+1} &= LRot(LRot(n2^i + KS^i, n1^i) + n1^i, n2^i). \end{aligned} \tag{3}$$

- **Step 9.** In addition, S keeps a copy of tuple $(Index^i, KS^i, r^i)$ in its memory.

3) Tag Authentication and Update Stage: Once S is authenticated, T is now authenticated as follows:

- **Step 1.** T generates the public message B^i using tuple $(n1^i, n2^i, Index^i, KS^i, ID)$ as follows:

$$B^i = LRot(LRot(n1^i + Index^i + KS^i + ID, n2^i) + n2^i, n1^i). \tag{4}$$

- **Step 2.** $T \rightarrow R : B^i$
- **Step 3.** $R \rightarrow S : B^i$
- **Step 4.** S calculates a local copy $B^{i'}$ of B^i using (4).
- **Step 5.** S now checks:
  if $B^{i'} = B^i$ **then**
  T is authenticated;
  else

Protocol is abandoned;
end if

- **Step 6.** T after sending B^i updates its values of tuple $(Index^i, KS^i)$ only after authenticating S using (3).

The message B^i can only be verified by a legitimate S. Successful mutual authentication concludes the protocol and S grants access to T. Both T and S have updated their values as shown in (3). S, after successfully authenticating T, deletes the old values of the tuple $(Index^i, KS^i, r^i)$ in its database to avoid tag impersonation.

III. Security and Performance Analysis

We now conduct a security analysis to show how our UMAP meets the goals of Section II-C, as well as a performance analysis which demonstrates that our scheme uses fewer resources than schemes given in [17], [26].

A. Mutual Authentication

We first show that our scheme provides mutual authentication by demonstrating that only a valid pair of S and T (in possession of KS) can generate public messages A and B, respectively, that will be accepted by the other party. The freshness of these public messages is ensured by the use of a random nonce in every authentication round.

1) **Authentication of the server:** S is authenticated by checking the authenticity of public message A. This message is generated using shared secrets known only to legitimate authenticating parties. Therefore, only a legitimate T can check the legitimacy of the message. The correctness of public message A thus determines the authenticity of S.
2) **Authentication of the tag:** Once T authenticates S successfully, it transmits its shared secrets in the form of a public message B. S can check the legitimacy and correctness of this message and hence authenticates T.

We consider whether an Adv without shared secrets can generate the public messages. To do so, Adv has to take over the authentication round after disrupting message $A||r$ and replaying it later by impersonating a genuine S, or Adv has to eavesdrop $Index$ and message $A||r$ and then take over the authentication round after disrupting and eavesdropping message B to replay it for T's impersonation. However, this is infeasible due to these reasons: 1) Adv cannot take over an ongoing authentication round (see Section II-B), 2) Adv cannot disrupt and eavesdrop at the same time (see Section II-B), 3) Adv has to perform a relay attack (see Section II-B). So, server and tag impersonation attacks are not feasible.

B. Tag Content Privacy

Each T has a unique static identity ID and is linked to the content of the particular tagged item. We want to transmit this ID confidentially so that an Adv is unable to read, copy or track it. In our scheme, S and T share a secret dynamic KS. Our scheme uses this KS to calculate two internal secret values $n1$ and $n2$ using a secure PRF f. We then use the tuple of ($n1$,$n2$,KS) to generate public messages which are used for transmitting the secret ID confidentially. Recall from Section II-E that each of the two public messages has the following form:

$$P = LRot(LRot(s2 + p + K + S, s1) + s1, s2). \quad (5)$$

where P and p are public values, $s1$, $s2$ are dynamic secret values, K is a shared secret key and S is a static secret (ID of T). The goal of the Adv is to disclose S. The complexity of recovering S is as follows:

1) The outer rotation from (5) is undone with complexity $O(\log_2 s2)$:

$$\begin{aligned} Q &= LRot^{-1}(P, s2), \\ &= LRot(s2 + p + K + S, s1) + s1. \end{aligned} \quad (6)$$

2) It requires a complexity $O(2^{s1} \times \log_2 s2)$ to subtract all possible values of $s1$ from R.H.S of (6):

$$\begin{aligned} R &= Q - s1, \\ &= LRot(s2 + p + K + S, s1). \end{aligned} \quad (7)$$

3) Further inner rotation is undone from (7) from all corresponding $2^{s1} \times \log_2 s2$ values (this doubles the complexity as $O(2 \times 2^{s1} \times \log_2 s2)$):

$$\begin{aligned} T &= LRot^{-1}(R, s1), \\ &= s2 + p + K + S. \end{aligned} \quad (8)$$

4) We now subtract public value p from (8) (this doubles the overall complexity as $O(2 \times 2 \times 2^{s1} \times \log_2 s2) = O(2^2 \times 2^{s1} \times \log_2 s2) \approx O(2^{s1} \times \log_2 s2)$:

$$\begin{aligned} U &= T - p, \\ &= s2 + K + S. \end{aligned} \quad (9)$$

5) Subtracting corresponding values of $s2$ from (9) requires an overall complexity of $O(2^{s1} \times \log_2 s2 \times \frac{2^{s2}}{\log_2 s2}) = O(2^{s1} \times 2^{s2})$:

$$\begin{aligned} V &= U - s2, \\ &= K + S. \end{aligned} \quad (10)$$

Concluding we shall have a total of 2^{3K} (considering $s1$, $s2$ and K are of same length) possible values of S. Since $s1$, $s2$ and K change in every authentication round (and $s1$,$s2$ are output of a secure PRF), our protocol provides privacy to T content.

C. Availability

In our scheme, both S and T update their shared secret KS and $Index$ after every successful authentication round in synchronization with each other. This synchronization is based on the receipt and authenticity of public messages A and B. Since update only takes place after a successful authentication and public messages A and B can only be generated by legitimate parties, we consider the following threats which can break the synchronization:

1) **Adversary disrupts message $A||r$:** Since T does not receive message $A||r$ sent by S, it will not update its values and keep the tuple $(Index^i, KS^i)$ in its memory. Though S updates to new tuple $(Index^{i+1}, KS^{i+1})$, it still has an entry for old tuple $(Index^i, KS^i, r^i)$ in its database. In this case, S identifies T with $Index^i$ which is still not updated and both remain synchronized.
2) **Adversary disrupts message B:** Since S does not receive message B, it has both old and new values as $((Index^i,\ KS^i, r^i)$,$(Index^{i+1},\ KS^{i+1}))$ stored in its database. Whereas T, on sending message B, has already updated its tuple to $(Index^{i+1}, KS^{i+1})$. This avoids de-synchronization as T is identified by S using $Index^{i+1}$.
3) **Adversary tampers with $A||r$ or B:** If an Adv tampers with the public messages A or random number r, a genuine T shall calculate a different value of $A^{'}$ which indicates that the message has been altered. Similarly, a genuine S can check the integrity of public message B.

D. Tag Anonymity

Two of the main privacy concerns in RFID systems are tracking and content privacy [1]. In our scheme, the $Index$ and public messages (A,B) change in every authentication round. This avoids tracking the location of a T.

E. Forward Security

In our UMAP, S generates a random value to calculate internal secrets using f. These internal secrets are used to update the $Index$ and KS after every successful authentication round. Therefore, if a T is compromised, it does not reveal any of its past and future communications.

F. Performance Analysis

We now briefly carry out a comparative analysis of performance parameters compared with the UMAPs given in [17], [26] which are the only existing ones that appear to meet the security goals detailed in Section II-C.

1) **Storage Overhead:** S stores the next potential and old values of the tuple $(Index, KS)$. Since S is considered to have less resource constraints, this lifts the burden on T's memory. Moreover, on successful completion of the protocol, S deletes the old entry thus saving storage space. T requires $2L$ bits storage on RAM for tuple $(Index, KS)$ and L bits of ROM to store its ID, which is less compared to other protocols of the same family as shown in Table II.
2) **Computation Overhead:** We have used lightweight functions (Addition modulo, left rotation and lightweight PRF) similar to other members of UMAP family. In our scheme, T has to verify one public message and calculate another message using lightweight functions that can be easily implemented in the ALU (Arithmetic and Logic Unit) processor of T. Therefore it computes two public messages, which are fewer compared to other schemes (which use the same functions i.e., a lightweight PRF to generate internal secrets and then adding, XOR-ing and left rotating these with other secret and public values) as shown in Table II. Moreover, we have also reduced the call to f to two as compared to three in other protocols and do not require XOR.
3) **Communication Overhead:** Our scheme communicates $2L$ bits during authentication round (considering each public message to be L bits) which is less than other schemes in Table II.

Table II
COMPARATIVE ANALYSIS OF DIFFERENT PROTOCOLS

Protocol	Storage	Computation	Communication
The Tagra et al. Protocol [17]	6L	4	4L
The SULMA Protocol [26]	6L	4	4L
Our Protocol (this paper)	2L	2	2L

Chien [2] categorized the RFID tags into four classes depending on the resources, cost and application. Ultralightweight class is considered to be very scarce in its resources. We consider that achieving security goals as mentioned in Section II-C using fewer resources is important in this class. UMAPs are designed using a tradeoff between cost, performance and level of security. Thus our protocol reduces the cost (in terms of storage) and enhances the performance (in terms of computation and communication) without degrading the level of security.

IV. CONCLUSION

This paper proposes a new UMAP designed for use in RFID devices with limited resources. These schemes provide security and privacy properties by updating the secret values and indexes in every authentication round. Synchronization between reader and tag is considered to be of prime importance. We have shown why our protocol overcomes weaknesses in previous UMAP designs and demonstrated that our protocol involves lower overheads.

REFERENCES

[1] A. Juels, "RFID security and privacy: a research survey," in *Journal on Selected Areas in Communications*, vol. 24, no. 2. IEEE, 2006, pp. 381–394.

[2] H.-Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," in *Transactions on Dependable and Secure Computing*, vol. 4, no. 4. IEEE CS, pp. 337–340.

[3] A. Klimov and A. Shamir, "Cryptographic Applications of T-Functions," in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science, vol. 3006. Canada: Springer, 2004, pp. 248–261.

[4] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda, "M^{2}AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags," in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science, vol. 4159. Springer, 2006, pp. 912–923.

[5] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags," in *Workshop on RFID Security*, ser. Lecture Notes in Computer Science. Springer.

[6] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda, "EMAP: An Efficient Mutual-Authentication Protocol for Low-Cost RFID Tags," in *On the Move to Meaningful Internet Systems*, ser. Lecture Notes in Computer Science, vol. 4277. Springer, pp. 352–361.

[7] T. Li and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols," in *International Information Security Conference*, ser. IFIP, vol. 232, South Africa, 2007, pp. 109–120.

[8] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy, "Passive Attack Against the M^{2}AP Mutual Authentication Protocol for RFID Tags," in *EURASIP Workshop on RFID Technology*, Austria, 2007.

[9] ——, "Breaking LMAP," in *Conference on RFID Security*, Malaga, Spain, July 2007.

[10] T. Li and R. H. Deng, "Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol," in *Proceedings of the International Conference on Availability, Reliability and Security*, Austria, 2007, pp. 238–245.

[11] J. C. H. Castro, J. M. Estévez-Tapiador, P. Peris-Lopez, and J.-J. Quisquater, "Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol with Modular Rotations," in *CoRR*, vol. abs/0811.4257, 2008.

[12] T. Cao, E. Bertino, and H. Lei, "Security Analysis of the SASI Protocol," in *Transactions on Dependable Secure Computing*. IEEE.

[13] H.-M. Sun, W.-C. Ting, and K.-H. Wang, "On the security of chien's ultralightweight rfid authentication protocol," in *Transactions on Dependable Secure Computing*. IEEE.

[14] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol," in *Workshop on Information Security Applications*, ser. Lecture Notes in Computer Science, vol. 5379, Korea, 2008, pp. 56–68.

[15] Z. Bilal, A. Masood, and F. Kausar, "Security Analysis of Ultra-lightweight Cryptographic Protocol for Low-cost RFID Tags: Gossamer Protocol," in *NBiS*. IEEE CS, 2009, pp. 260–267.

[16] K.-H. Yeh and N. Lo, "Improvement of Two Lightweight RFID Authentication Protocols," in *Information Assurance and Security Letters*, vol. 1. Dynamic Publishers Inc., 2010, pp. 6–11.

[17] D. Tagra, M. Rahman, and S. Sampalli, "Technique for Preventing DoS Attacks on RFID Systems," in *International Conference on Software Telecommunications and Computer Networks*, Croatia, 2010.

[18] G. Avoine, X. Carpent, and B. Martin, "Privacy-friendly synchronized ultralightweight authentication protocols in the storm," in *Journal of Network and Computer Applications*, vol. 35, no. 2, 2012, pp. 826–843.

[19] M. David and N. R. Prasad, "Providing Strong Security and High Privacy in Low-Cost RFID Networks," in *Security and Privacy in Mobile Information and Communication Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 17, Turin, Italy, 2009, pp. 172–179.

[20] J. C. Hernandez-Castro, P. Peris-Lopez, R. C. Phan, and J. M. Estevez-Tapiador, "Cryptanalysis of the David-Prasad RFID Ultralightweight Authentication Protocol," in *Workshop on RFID Security*, ser. Lecture Notes in Computer Science, vol. 6370, Turkey, 2010, pp. 22–34.

[21] Y.-C. Lee, Y.-C. Hsieh, P.-S. You, and T.-C. Chen, "A New Ultralightweight RFID Protocol with Mutual Authentication," in *WASE International Conference on Information Engineering*, vol. 2, Taiyuan, Shanxi, 2009, pp. 58–61.

[22] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and J. C. A. van der Lubbe, "Security Flaws in a Recent Ultralightweight RFID Protocol," in *Workshop on RFID Security*, ser. Cryptology and Information Security, vol. 4, Singapore, 2010, pp. 83–93.

[23] P. Peris-Lopez, J. C. H. Castro, R. C.-W. Phan, J. M. Estévez-Tapiador, and T. Li, "Quasi-Linear Cryptanalysis of a Secure RFID Ultralightweight Authentication Protocol," in *Information Security and Cryptology - 6th International Conference*, pp. 427–442.

[24] A. Eghdamian and A. Samsudin, "A Secure Protocol for Ultralightweight Radio Frequency Identification (RFID) Tags," in *Informatics Engineering and Information Science*, ser. Communications in Computer and Information Science, vol. 251, Malaysia, 2011, pp. 200–213.

[25] G. Avoine and X. Carpent, "Yet Another Ultralightweight Authentication Protocol that is Broken," in *Workshop on RFID Security*, ser. Lecture Notes in Computer Science, Netherlands, 2012.

[26] M. Kianersi, M. Gardeshi, and M. Arjmand, "SULMA: A Secure Ultra Light-Weight Mutual Authentication Protocol for Lowcost RFID Tags," in *International Journal of UbiComp*, vol. 2. India: AIRCC, 2011, pp. 17–24.

[27] G. E. Standards, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz," in *Specification for RFID Air Interface*, 2008.

[28] Y.-C. Lee, Y.-C. Hsieh, P.-S. You, and T.-C. Chen, "A New Ultralightweight RFID Protocol with Mutual Authentication," in *WASE International Conference on Information Engineering*, Taiyuan, Shanxi, 2009, pp. 58–61.

[29] K.-H. Yeh, N. Lo, and E. Winata, "An Efficient Ultralightweight Authentication Protocol for RFID Systems," in *Workshop on RFID Security*, ser. Cryptology and Information Security, vol. 4, Singapore, 2010.

[30] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estévez-Tapiador, and J. C. A. van der Lubbe, "Security Flaws in a Recent Ultralightweight RFID Protocol," in *CoRR*, vol. abs/0910.2115, 2009.