A Set of Efficient Privacy Protection Enforcing Lightweight Authentication Protocols for Low-cost RFID Tags

Pierre-François Bonnefoi, Pierre Dusart and Damien Sauveron
XLIM (UMR CNRS 7252 / Université de Limoges), Département Mathématiques Informatique. Limoges, France
Email: bonnefoi@unilim.fr, pierre.dusart@unilim.fr, damien.sauveron@unilim.fr

Raja Naeem Akram and Konstantinos Markantonakis
Information Security Group Smart Card Centre, Royal Holloway, University of London. Egham, United Kingdom
Email: R.Akram@rhul.ac.uk, K.Markantonakis@rhul.ac.uk

***Abstract*—Providing security and privacy protection in Radio Frequency IDentification (RFID) tags is a challenging task due to their highly constrained resources. Because tags cannot support strong cryptography, security must rely on low-computational solutions. Privacy protection is often an additional requirement for acceptance of the technology by end-users. Thus, dedicated lightweight algorithms and protocols need to be designed.**

In this paper, we propose a set of extremely lightweight (optionally mutual) authentication protocols between a tag and a database (sharing a secret value) which protect the identity of the tag. However, if, for some practical reasons, the identity of the tag needs to be revealed to the reader, an additional optional last step can be added at the end of our protocols to satisfy this requirement. Our proposals are inspired by the protocol of Chien *et al.* [1] but they are more efficient, since they operate in $O(1)$ at database side, whilst covering security flaws and enforcing privacy protection. We examine the adequacy of our solutions regarding the chosen security requirements and their strengths against some well-known and more exotic attacks.

I. Introduction

The Radio Frequency IDentification (RFID) tags are small integrated-chip circuits which usually do not have an internal battery (power is supplied by a reader through an inductive coupling). Thus processing time and communication distance are limited (e.g. a tag could only communicate up to a few meters) but sufficient for many applications requiring a remote wireless identification (ID) to identify products, animals, persons, etc. These applications [1, 2] must be designed carefully in order to ensure security and privacy protection in different areas like warehouse or library management, fast shopping checkouts, pets identification, anti-counterfeiting, physical access control (e.g. to buildings or public transport systems), logical access control (e.g. to online application or content), etc.

As depicted in figure 1, an RFID system is usually composed of a database hosted on a server which is connected to several readers (for an access control system for public transport system, there are readers at each entrance). Each of these readers can read one or several tags at the same time according its capabilities and the database has to handle authentications in an efficient way.

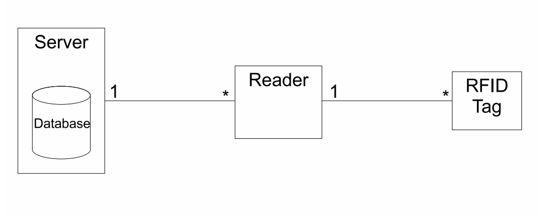

Figure 1. Simplified overview of an RFID infrastructure

For a wide market penetration and for economical reasons, tags are manufactured with low computational and memory capabilities. Although they are not designed to be truly tamper-resistant, they provide sufficient levels of protection that require expensive reverse engineering methods for an attacker to disclose sensitive information like internal keys. Therefore the main attack vector is the communication channel. For instance, an adversary can abuse it in order to obtain some secret information by performing various attacks like physical layer analysis [3], eavesdropping [4, 5]. He can also perform more sophisticated attacks to disrupt communication (e.g. jamming the channel to disable exchanges between the tag and the rest of the infrastructure) [6, 7].

A well-know issue of RFID systems, is related to privacy protection, since each tag is usually associated to a unique identity that must be protected against disclosure to unauthorized entities.

A. Contributions

The salient contributions of this paper are to:

1) Specify a set of protocols which aim to efficiently enforce privacy protection at reader level for low-cost tags during an (optionally mutual) authentication process. They are inspired of the protocol proposed by Chien *et al.* [1] but they are more efficient, since the database can operate in

$O(1)$, whilst covering security flaws [8, 9] and enforcing privacy protection. The main goal of proposing several protocols is to enable the RFID system issuer to choose one that better meets any identified requirements.

2) Provide security and performance analysis to demonstrate that the proposed protocols support both the functional and security requirements defined in this paper.

B. Structure of the paper

Section II presents the functional requirements of the targeted RFID systems. Then, a threat analysis is performed and some assumptions are given in section III. Section IV describes the security requirements for our protocols in order to protect at reader level the identity of tags whilst keeping the computational complexity low. The protocols are then detailed in section V. Section VI provides a security analysis of protocols to give a rational that security requirements are satisfied. Finally, section VII provides performance analysis. Section VIII is dedicated to related work and section IX provides concluding remarks.

II. Functional Requirements

As aforementioned in the previous section, our proposal is targeting RFID systems where an end-to-end authentication is performed by the database in order to ensure privacy protection at reader level. A database is thus a central element and both infrastructure and protocol need to support it in an efficient way. For instance, unambiguous and fast information retrieval from the database is an essential feature of RFID systems that the protocol should have to carefully protect. Subsequently we present the minimal functional requirements for targeted RFID systems.

A. Efficiency

1) Uniqueness of Tag's ID: In RFID systems that our protocols are addressing each tag should have a unique Secure ID (SID) shared with the database. At enrollment of the tag, an entry is added in a hash table of database: SID being the value to store and $hkSID$ the related key to access this value.

2) Authentication Processing Time: In RFID systems, authentication must often be real-time. Indeed the database must be able to handle multiple authentication tries in the same time (from several tags through one or several readers). For instance in a shop, it is not acceptable that authentication of products takes several seconds, since products are scanned by multiple cashiers at the same time and customers may complain if the overall process is not efficient enough. Even if our authentication protocols are not designed to be used in "time-critical" environments, the authentication processing time should have to be limited in order to guarantee efficiency.

a) Fast Information Retrieval and Scalability: To ensure faster information retrieval than Chien *et al.*'s protocol [1] and to support scalability when the population of tags is growing on the database side, it is required that $hkSID$ should be stored on the tag.

b) Minimize Exchanged Messages: To ensure efficiency of communications, our protocols must minimize the number of messages and the size of the data exchanged on the RF interface between reader and the tag. Indeed communication standards on this latter interface are slower than those between the reader and database. However the number of messages should be minimized without jeopardizing security.

B. Lightweight

For a wide adoption, our proposals address RFID systems operating with low-cost tags. Thus our protocols will have to be in the "lightweight class" of the well established Chien's classification of RFID authentication protocols [10]: i.e. protocol that can only use Random Number Generator (RNG) and simple functions like a Cyclic Redundancy Code (CRC) checksum, but no hash function.

C. Resilience

Our protocols will have to protect the infrastructure against Denial-of-Service (DoS) attacks. A stateless-designed protocol, i.e. a protocol in which communication partners do not need to retain session information to ensure a good execution of the run, would help to support this requirement.

III. Threats Analysis and Assumptions

The adversary model considered in this paper is stronger than the Dolev-Yao model [11] since our attacker can perform timing attacks.

A. Threats

The aim of an attacker is to extract sensitive information from the protocol or from the communicating entities in order to break the privacy protection or simply to bypass the authentication process. The identified threats are listed bellow:

- (T1) An attacker can impersonate one entity: a tag, a reader or the database.
- (T2) An attacker can eavesdrop the communication.
- (T3) An attacker can inject chosen data.
- (T4) An attacker can disrupt the communication channel.
- (T5) An attacker can retrieve secret information from the implementation leakages.
- (T6) An attacker can exploit unexpected states in the design of the protocol.

B. Assumptions

The assumptions for the secure and efficient operation of our protocol are:

- (A1) The RFID system provider (database owner) is trusted.
- (A2) The server which hosts the database is secure and protected.
- (A3) The infrastructure between the server and the readers is trusted both in terms of information protection (e.g. thank to a Secure Access Module embedded in the readers that establishes a secure connection with the database) but also in terms of availability.

- (A4) The tag is tamper-resistant and not tamper-proof.
- (A5) The tag has a good Random Number Generator (RNG).
- (A6) The tag supports a function named g which is in the aforementioned lightweight-class and one-way (i.e. it is not possible to retrieve the input thanks to the output). We assume that this function is not vulnerable against timing and algebraic attacks. It can be considered as a Pseudo Random Number Generator (PRNG). For this function, there are several candidates: LAMED [12], AKARI-X [13], the ring-FCSR Stream Cipher [14, 15], etc. Since these functions operate on different input and output sets, we will propose several variants for g, called g_1, g_2 and g_3.

IV. Security Requirements

To ensure adequate security and privacy protection of the SID and $hkSID$ used in the authentication protocols, we draw the following security requirements. To support them, our protocols use a key (K) shared by the tags and the database.

A. Authentication

The first requirement is to be able to perform a tag authentication to the database. However, to support more use cases, we add the requirement that the tag is also able to authenticate the database.

B. Privacy protection

The second security requirement of our protocols is to avoid traceability at reader level in order to enforce privacy protection. During the authentication process with the database, the communications must not reveal neither the SID nor $hkSID$ (i.e. they should not be exchanged in plaintext). Communications must also not reveal if the tag was ever seen before (i.e. traces of authentication runs for a same tag must be different at each run). However for specific applications, our protocols will provide the possibility to reveal the tag identity to the reader during an optional final step: e.g. when the reader is fully trusted and some specific operations must be performed. It must be noted that our protocols do not ensure anonymity at the database level since the RFID system provider is considered as a trusted party (see assumption A1).

V. Protocols

For each of our protocols we provide two variants. The first version only achieves a tag authentication to the database. The second version achieves a mutual authentication of the tag and the database. Since the three protocols follow the same scheme, we will first present its building blocks and the basic operations. The main difference between our three protocols is the sets of input/output over the g function.

A. Notation

Table I introduces the notation used to describe our protocols.

Table I
Notation used in protocols description.

DB	:	Denotes a database
R	:	Denotes a reader
T	:	Denotes a tag
SID	:	Denotes a Secure ID of a tag
$hkSID$	:	Denotes key of Secure ID of a tag in hash table of database
K	:	Denotes the key shared by tags and database
R_1	:	Random number generated by the reader
R_2	:	Random number generated by the tag
$X_{[i..j]}$	:	Denotes the sequence of bits of X from i^{th}-bit to j^{th}-bit; $i > j$
$A \to B$	:	Message sent by an entity A to an entity B
$X \oplus Y$	:	Bitwise XOR of data items X, Y
$X \wedge Y$	:	Bitwise AND of data items X, Y
$X \ll Y$	:	Bitwise left shift on X of Y bits
$X \gg Y$	:	Bitwise right shift on X of Y bits
$X\|\|Y$	:	Concatenation of data items X, Y
$X = Y$	:	Assign value Y to X
$X \overset{?}{=} Y$	:	Compare equality of X with Y
Query	:	Message to start the authentication procedure
$Auth_{DB}$	:	Authentication ticket from the database
$Auth_T$	:	Authentication ticket from the tag
OK	:	Message to acknowledge success of authentication
NOK	:	Message to acknowledge failure of authentication

B. Definitions

Before providing the details of our proposed protocols, several functions must be defined.

The $g_1(X)$ is the PRNG function:

$$g_1 : \begin{array}{ccl} \mathbb{F}_2^l & \to & \mathbb{F}_2^l \\ X & \mapsto & g_1(X) \text{ where the resulting value is random.} \end{array}$$

The $g_2(X,Y)$ is the PRNG function:

$$g_2 : \begin{array}{ccl} \mathbb{F}_2^l \times \mathbb{F}_2^l & \to & \mathbb{F}_2^l \\ X, Y & \mapsto & g_2(X,Y) \text{ where the resulting value is random.} \end{array}$$

The $g_3(X)$ is the PRNG function:

$$g_3 : \begin{array}{ccl} \mathbb{F}_2^l & \to & \mathbb{F}_2^{2l} \\ X & \mapsto & g_3(X) \text{ where the resulting value is random.} \end{array}$$

$Left(X)$ and $Right(X)$ are two substring functions which respectively return the half-most and the half-least significant bits of X.

$$Left : \begin{array}{ccl} \mathbb{F}_2^{2m} & \to & \mathbb{F}_2^m \\ X & \mapsto & Left(X) = X_{[2m-1..m]} \end{array}$$

$$Right : \begin{array}{ccl} \mathbb{F}_2^{2m} & \to & \mathbb{F}_2^m \\ X & \mapsto & Right(X) = X_{[m-1..0]} \end{array}$$

C. The Tag Enrollment Phase

We assume SID, $hkSID$ and $K \in \mathbb{F}_2^l$. The following steps must be performed by the issuer prior to use the protocols.

- Generate the SID value and the $hkSID$ key (ideally values and keys are uniformly distributed, i.e. they are not consecutive) with the property that there is no collision on SID and $hkSID$ (no identical value as SID and no identical keys as $hkSID$).
  For instance, a random number in $\mathbb{F}_2^l$ is generated by the server, the database checks that this value is not present

in the values of hash table of the database in $O(n)$. The $hkSID$ key can be computed with a PRNG function like g_1 applied on the generated SID since its output being unique, there will be no collision with an already existing key if there is no collision on SID.

- Send the entry $hkSID$ and SID of the hash table along with the related secret key K to the tag.
- Put the tag in a ready to use state (i.e. a state in which it is no longer possible to update $hkSID$ and SID).
- Add in the hash table of the database this entry $hkSID$, SID along with additional information about the entity to which the tag is delivered.

Now the tag is considered fully operational.

D. The Authentication Process of The Main Scheme

We assume R_1, $R_2 \in \mathbb{F}_2^{l/2}$. For simplicity reasons, we also choose the following $2m = l$ (for m introduced in substring functions). Subsequently, there are two main phases presented: the tag authentication phase and the database authentication phase. If the mutual authentication is not required the database authentication phase can be skipped. Finally a tag identity disclosure phase is optionally proposed for application-specific purposes.

1) Tag Authentication to Database Phase: The protocol messages for the tag authentication to the database are listed in table II and described below. In figure 2, this phase encompasses steps 1 and 2.

Table II
TAG AUTHENTICATION TO DATABASE.

1. $R \rightarrow T$	:	Query$\|\|R_1$
2. $T \rightarrow R$	:	$R_2\|\|hkSID'\|\|Auth_T$
3. $R \rightarrow DB$	:	$R_1\|\|R_2\|\|hkSID'\|\|Auth_T$
4. $DB \rightarrow R$	:	OK/NOK

a) Message 1: The reader generates a random number, R_1, and it sends it with the Query. On receipt of this message, the tag generates a random number, R_2, and computes SM, a session mask with $SessionMask(R_1, R_2, K)$. This mask will be used to conceal the $hkSID$ during exchanges and to compute $Auth$, a mutual authentication ticket.

b) Message 2: In order to be authenticated by the database, the tag computes $hkSID'$ and $Auth$ respectively with $Mask(hkSID, SM)$ and $AuthTicket(SID, SM)$. Since $Auth$ is a mutual authentication ticket, only the $Auth_T$, defined here as $Left(Auth)$ is sent along with R_2, $hkSID'$.

c) Message 3: The reader receives message and forward these values to database along with the previous random number, R_1. On receipt of this message, the database computes SM with $SessionMask(R_1, R_2, K)$ and $hkSID$ with $UnMask(hkSID', SM)$. Then it verifies that the $hkSID$ is a valid key in the hash table and extracts the related SID. To authenticate the tag, it computes its own authentication ticket $Auth = AuthTicket(SID, SM)$ and compares its left part, i.e. $Left(Auth)$, with $Auth_T$.

d) Message 4: If the $hkSID$ was not a valid key or if $Left(Auth)$ is different of $Auth_T$, the database sends NOK to the reader and then the authentication of tag has failed. Else, it sends OK to the reader to indicate a successful authentication.

At this stage, the tag authentication has been performed and if a mutual authentication is not needed, the next phase can be skipped. Our protocol suite being flexible, the optional deliberate disclosure phase presented below can be achieved if SID must be revealed to the reader for a specific application-purpose.

Note that our protocol suite is flexible since if SID must be revealed to the reader for application-purpose, the deliberate disclosure phase can be achieved even when next phase have been skipped.

2) Database Authentication to Tag Phase: Protocol messages for the subsequent database authentication to tag are listed in table III and described as below. It only occurs if a tag authentication phase was successful. In figure 2, this phase is related to step 3.

Table III
DATABASE AUTHENTICATION TO TAG.

1. $D \rightarrow R$	:	$Auth_{DB}$
2. $R \rightarrow T$	:	$Auth_{DB}$
3. $T \rightarrow R$	:	OK/NOK

a) Message 1: The database computes $Auth_{DB} = Right(Auth)$ and sends it to the reader.

b) Message 2: The reader receives the message and forwards it to the tag. On receipt of this message, the tag compares $Auth_{DB}$ with $Right(Auth)$.

c) Message 3: If $Right(Auth)$ is different of $Auth_{DB}$, the tag sends NOK to the reader and then the authentication of database has failed; else, it sends OK to the reader to inform it of a successful authentication.

3) Optional Deliberate Disclosure of SID to Reader Phase: The protocol messages for the subsequent optional disclosure of tag's SID to reader are listed in table IV and described as below. It only occurs if, at least, a tag authentication phase was successful. In figure 2, this phase encompasses step 4.

Table IV
DELIBERATE DISCLOSURE OF SID TO READER.

1. $R \rightarrow DB$	:	OK
2. $DB \rightarrow R$	:	SID

a) Message 1: The reader informs database that tag has authenticated it and it needs the SID by sending OK.

b) Message 2: The database provides it the requested SID of the tag.

E. Our Set of Protocols

In this section, three protocols are described. They have the same security requirements and the same communication costs. They differ on the computational cost, storage cost as

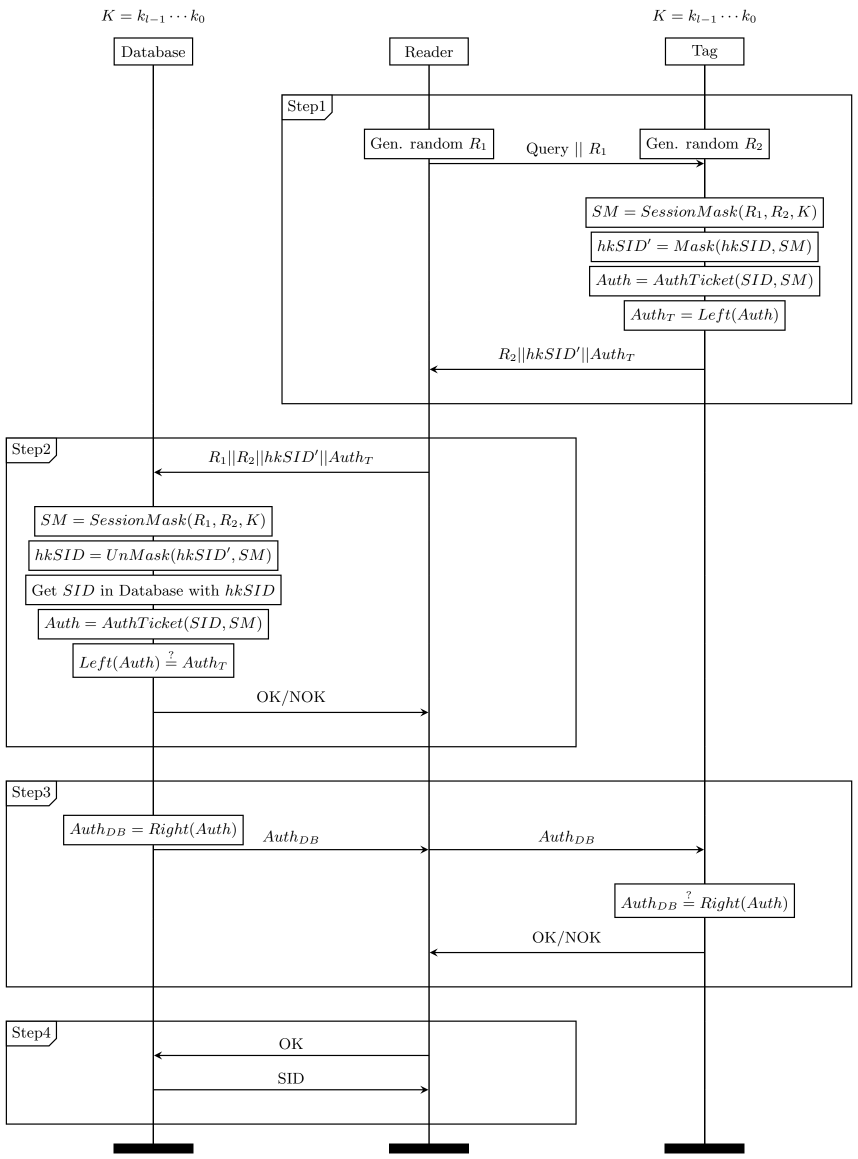
$K = k_{l-1} \cdots k_0$
Database
Reader
Tag
$K = k_{l-1} \cdots k_0$
Step1
Gen. random R_1
Query || R_1
Gen. random R_2
$SM = SessionMask(R_1, R_2, K)$
$hkSID' = Mask(hkSID, SM)$
$Auth = AuthTicket(SID, SM)$
$Auth_T = Left(Auth)$
$R_2||hkSID'||Auth_T$
Step2
$R_1||R_2||hkSID'||Auth_T$
$SM = SessionMask(R_1, R_2, K)$
$hkSID = UnMask(hkSID', SM)$
Get SID in Database with $hkSID$
$Auth = AuthTicket(SID, SM)$
$Left(Auth) \overset{?}{=} Auth_T$
OK/NOK
Step3
$Auth_{DB} = Right(Auth)$
$Auth_{DB}$
$Auth_{DB}$
$Auth_{DB} \overset{?}{=} Right(Auth)$
OK/NOK
Step4
OK
SID

Figure 2. Main scheme of the proposed authentication protocols.

illustrated in sections VII-A and VII-B, and on the g function used as illustrated below.

1) Protocol 1 using $g_1(X)$: In protocol 1, the g PRNG function is $g_1(X)$ and other presented functions are defined as follow:

- $SessionMask(R_1, R_2, K) = g_1((R_1||R_2) \oplus K)$
- $Mask(hkSID, SM) = hkSID \oplus SM$
- $AuthTicket(SID, SM) = g_1(SID \oplus SM)$
- $UnMask(hkSID', SM) = hkSID' \oplus SM$.

2) Protocol 2 using $g_2(X, Y)$: In protocol 2, the g PRNG function is $g_2(X, Y)$ and other presented functions are defined as follow:

- $SessionMask(R_1, R_2, K) = g_2(R_1||R_2, K)$
- $Mask(hkSID, SM) = hkSID \oplus SM$
- $AuthTicket(SID, SM) = g_2(SID, SM)$
- $UnMask(hkSID', SM) = hkSID' \oplus SM$.

3) Protocol 3 using $g_3(X)$: In protocol 3, the g PRNG function is $g_3(X)$ and other presented functions are defined as follow:

- $SessionMask(R_1, R_2, K) = g_3((R_1||R_2) \oplus K)$
- $Mask(hkSID, SM) = hkSID \oplus Left(SM)$
- $AuthTicket(SID, SM) = SID \oplus Right(SM)$
- $UnMask(hkSID', SM) = hkSID' \oplus Left(SM)$.

VI. Security Analysis

In this section, we verify that our protocols meet the security properties identified section IV and that they are resistant to the attacks and threats identified section III-A.

A. Properties

1) Untraceability: The property of untraceability could be expressed through the following rule [16]: it is not possible to determine if two traces of the protocol are related to the same tag to to two different tags.

For our protocols, the exchanged variables R_1, R_2 are random. The tag can be identified by SID or $hkSID$. However, these values are never exchanged in plaintext. Since SM is computed with the $SessionMask$, a function designed to output values looking like random for an adversary, and this value is used to mask $hkSID$, $hkSID'$ seems to be random. Since the $AuthTicket$ function is also designed to exploit the randomness of SM to hide SID value in the $Auth$, the $Auth_T$ also seems to be random. Thus no identity-related information can derived from this value.

An adversary has a probability to identify twice **a same** tag (if each tag in the database has the same probability, $\mathcal{P}$, to be used) of

$$\mathcal{P} \leq \frac{(\text{number of recorded sessions})}{(2^l * \text{number of SIDs in database})}$$

(where $l/2$ is the length of R_i) since the same R_1 and R_2 must occur in among the recorded sessions and should give the same $hkSID'$ and $Auth_T$.

However the probability, $\mathcal{P}$, to differentiate among traces (recorded sessions) that is not **the same** tag is

$$\mathcal{P} \leq (\text{number of recorded sessions})/2^l$$

(if at least two traces with same R_1, R_2 contain different $hkSID'$, from the above definition an attacker can determine that this is not the same tag for each trace).

These probabilities are valid if the adversary only eavesdrops the communications. Indeed if an active attacker is involved in a subsequent transaction and at the same time he has previously recorded sessions involving the tag, he will attempt to identify the tag (through SID or $hkSID$) by using a fake reader that will always use the same value for R_1. To identify $hkSID$, he must find a same $hkSID'$. However, as the attacker cannot choose R_2, SM looks like a random value and it is very difficult to successfully identify $hkSID$. To identify SID, he must find a same $Auth_T$. However, since a same $Auth_T$ does not implied to have a same $Auth$ ($Auth_T$ is only the left part of $Auth$; several couple (SID, SM) can give the same left part when $AuthTicket$ function is applied to them), the probability, $\mathcal{P}$, to identify another time a tag is:

$$\mathcal{P} \leq (\text{number of recorded sessions})/2^{l/2}$$

where $l/2$ is the size of R_2.

2) Mutual authentication: The server identifies the tag with the $hkSID$ key and gets the related SID in the hash table of the database. Next the server authenticates tag by comparing its computation $Auth$ (Left substring) with $Auth_T$, the tag authentication ticket.

The tag authenticates the server by verifying the server computation $Auth$ (Right substring) with $Auth_{DB}$, the tag authentication ticket. Since only the server and the tag have key K and SID as shared secrets, our protocols provide mutual authentication.

It can be noticed that, after a successful mutual authentication, SID can be considered as a secret shared by the database and the tag and it can thus be used for security operations (like signing the subsequent exchanges if message authentication and integrity are required).

B. Attacks Analysis

In this section we analyze the resistance of our protocols against several attacks achieving threats presented in section III.

1) Random Authentication Attack: For this attack, an opponent can attempt a random authentication to impersonate a tag. He sends random data from a fake tag and hopes that it will be luckily authenticated by the database. In our protocols, the database authenticates the tag if there is an SID for $hkSID$ and that use of this SID in computation leads to $Left(Auth) = Auth_T$. The probability of a successful random authentication is that of the $hkSID$ is valid multiply by that of $Auth_T$ is valid. $hkSID$ being valid if there is a SID for it and $Auth_T$ being related to $Auth$ which is computed by the g PRNG function applied on SID and SM (a random number-like), the probability, $\mathcal{P}$, is:

$$\mathcal{P} \leq (\text{number of SIDs in database})/2^{l+l/2}$$

(where l is the size of $hkSID'$ and $l/2$ is the size of $Auth_T$).

2) Replay Attack: Since an opponent can eavesdrop and record the communications between the tag and the reader to impersonate a tag, he can try to replay authentication with recorded data answered by a tag. However as the value provided by the reader is random, the probability to use the recorded data another time is low.

For a passive attacker, the probability, $\mathcal{P}$, to reuse data of recorded sessions (with different R_1) to successfully authenticate a tag is:

$$\mathcal{P} \leq (\text{number of recorded sessions})/2^{l/2}$$

(where $l/2$ is the size of R_1) since for each R_1 recorded, the attacker has a true answer.

For an active attacker, like in the attack done by Tom Van Deursen *et al.* [8, 9] on Chien *et al.*'s protocol [1], the probability is the same as performing a random authentication attack. Indeed an algebraic difference on R_1 cannot be exploited since the g functions used in $SessionMask$ functions defined in our protocols are resistant to algebraic attacks (see assumption A6).

3) Man-in-the-middle and Relay Attacks: By design, the proposed protocols are protected against the man-in-the-middle attack but not against relay attacks. To be protected against this latter attack, our protocols, like those presented in the related work section, should implement additional countermeasures like distance-bounding protocols [17]–[19] that try to detect the delay induced by the relay between the leech and the ghost, tag localization [20], etc. However, the performance penalties incurred due to these countermeasures may not be suitable for low-cost tags.

4) DoS and Desynchronization Attacks: By design, the proposed protocols being stateless, they are protected against DoS and desynchronization attacks. Indeed, they do not establish dependencies between each authentication. Each transaction being considered as a new one, no synchronization is required between the database and the tag. By design, since there is no update in the tag side, it is impossible to put it in a dead-end state.

For the security of the database common solutions (like tracking queries of readers and blocking them in the the case of malevolent behavior) could be used to avoid a reader to overload the database query engine. However, the infrastructure between readers and database being assumed trusted (see assumption A3), study of these solutions is out of the scope of this paper.

5) Chosen Values Attack: This attack is based on the fact that a protocol may be vulnerable to some specific values. Thank to the use of g function and random numbers in the computation of the session mask SM, there are no specific values forbidden in our protocols.

6) Timing Attack: An attacker can perform a well-known attack in world of smart card: the timing attack. The timing attack is a side channel attack presented by Kocher [21] in which the attacker attempts to compromise a cryptosystem by analyzing the time taken to execute operations (like cryptographic algorithms). Every logical operation in a tag takes time to execute and the time can differ based on the input. With precise measurements of the time for each operation, an attacker can work backwards to the input. This attack is very efficient when a secret is used as a loop bound.

This attack is not applicable on our protocols since the only function likely to be attacked is g and it is assumed (see assumption A6) being resistant against this attack (which is realistic for candidates mentioned in section III-B). However, this attack may be effective on some implementations of the protocols [22, 23] presented in the related work section.

7) Database Impersonation Attack: The tag authenticates the database if $Right(Auth) = Auth_{DB}$. This equality depends on the behaviour of g function whose outputs are randoms. The probability, $\mathcal{P}$, is that of $Auth_{DB}$ is valid, i.e.:

$$\mathcal{P} \leq 1/2^{l/2}$$

(where $l/2$ is the size of $Auth_{DB}$).

It is obvious that this attack cannot be considered as practicable if l is long enough.

C. Physical Attacks

Even if we assumed (see assumption A4) that our tag provides sufficient security against a key recovery from the memory, some extensions can be added to increase the security. Some groups of tags can be made and named T_i. Each tag of the group can be personalized with the same key K_{T_i} at the enrollment phase. Thus the tag has to store this group identifier T_i which can be represented on n bits. During the first step of the authentication, T_i can be sent in plaintext along with the rest of the message in order that the database used the related key, K_{T_i} for its computation.

Obviously to preserve the privacy, tags from different groups should be given to user in a random way and n should be kept smaller than l. The advantage of this solution is that an attacker will not break the whole RFID system: if he succeeds to extract key, K_{T_i}, in memory of a tag, only tags of the group T_i will be compromised.

In addition, it may be possible to add this key update phase in our protocol after a successful authentication based on database decision (for instance database is aware that K_{T_i} of the group T_i has been compromised and it wish to change key). Note that updating key, means to change the tag of group (which is not a problem, since tags have been given in a random way to their owners) from T_i to T_j. Protocol messages for a key update are listed in table V and described as below.

Table V
KEY UPDATE FOR A TAG BELONGING TO A GROUP T_i AND MOVED IN T_j.

1. $DB \rightarrow R$	:	$(K_{T_j} \oplus SID)\|\|T_j$
2. $R \rightarrow T$	:	$(K_{T_j} \oplus SID)\|\|T_j$

a) Message 1: After a successful authentication, the database decides to change the key K_{T_i} of the tag belonging to the group T_i to another group, lets say T_j. T_j can be a new group and in this case on database side, the database has to add it by generating a key K_{T_j}. In any case, T_j has just been created or if T_j already existing, the database sent to the reader $(K_{T_j} \oplus SID)$ and the new group identifier T_j.

b) Message 2: The reader forwards this message to the tag. On receipt of this message, to avoid a desynchronization attack, the tag first unmasks the key with its SID, puts it in an additional memory space of length l bits and put the group identifier T_j in an additional memory space of length n bits. Then a flag should be set atomically[1] to inform the tag that it must now use the space containing T_j and K_{T_j} instead of T_i and K_{T_i}. Spaces containing T_i and K_{T_i} will serve for a future key update. The flag is just here to indicate which is the current space to read it.

However, if this key update mechanism enforce the protocols, it requires a bit more of $l + n$ supplementary bits in

[1]An atomic update guarantee that either the new or the previous value will be in the updated memory area at the end of update but not another value – it is specially useful to protect update against error occuring during the operation.

the tag memory. Contrary to other protocols being prone to desynchronization attack, our solution should be more resistant due to the atomical update mechanism that we explicitly propose and that is only for 1 flag. Indeed, if the flag is not updated due to a desynchronization attack, the tag is still able to communicate with the database since it is still using the old group identifier T_i and the old related key K_{T_i}.

VII. Performance Analysis

In this section, the costs of our proposals are analyzed to prove that they can be successfully implemented on low-cost tags, which is a key point to attain a great market penetration, since they minimize the computational, storage and communication costs whilst being efficient and secure (as shown in the previous section).

A. Computational Cost Analysis

The lightweight property is ensured since on tag's side since our protocols only require that tag uses a RNG to create the challenge R_2, a PRNG function, g, and few common functions (XOR, Substring part). If we analyse each function individually, a PRNG is lightweight (but not ultralightweight) and the $Left$ and $Right$ functions are also lightweight since they can be implemented with masked bitwise AND and shift (for instance with $X \in \mathbb{F}_2^8$, $Right(X)$ and $Left(X)$ can be respectively computed with $(X \wedge 0\text{x}0F)_{[3..0]}$ and $(X \wedge 0\text{x}F0 \gg 4)_{[3..0]}$, i.e. for each only the 4 least significant bits are kept).

The total cost of our protocols on tag's side depends of the cost of the g function and the number of calls to it (it is called twice for protocols using g_1 and g_2 and only once for that using g_3).

B. Storage Cost Analysis

The tag must permanently store: the tag's identity, SID, the key of SID in hash table of database, $hkSID$, and a shared secret key, K, each of them of the length of l bits. To achieve computations and temporary storage of R_1, tag also handles several variables of the length of l and $l/2$ bits.

Depending of the chosen proposed protocol we evaluate the total memory consumption from $5l + l/2$ to $6l + l/2$ bits. We should add to this value the memory consumption required by the g function.

However for comparison purposes of our protocols to other lightweight RFID protocols we only take into account the $3l$ bits used in persistent memory like all authors did in their papers [22, 24, 25].

C. Communication Cost Analysis

The communication cost is important since if the communication cost is heavy, the protocol is slow to execute and an user must wait to have her authentication response.

The size of exchanged messages are:

- for reader-to-tag communication, the query of let say 40 bits, a challenge R_1 of $l/2$-bits, an authentication response $Auth_{DB}$ of $l/2$-bits, i.e. a total of $40 + l$ bits;
- for tag-to-reader communication, almost the same (R_2 and $Auth_T$) but $hkSID'$ of l bits and the final response (OK/NOK) of 1 bit must be added and the query must be removed, i.e. a total of $1 + 2l$ bits.

D. Implementation Advices

To reduce the cost of the tags used in our protocols, the RNG can be replaced by a PRNG which could be the g function already used in implementation.

To improve performance on back-end, for a mutual authentication, a developer is advised to gather OK/NOK and $Auth_{DB}$ in only one message between step 2 and step 3.

VIII. Related Work

To compare our proposals with related work, we have studied a number of lightweight authentication protocols and for most of them, several attacks were possible, even some which are, at the best of our knowledge, unpublished but that we cannot expand here due to space constraints, like: a) an algebraic attack is possible on [26], an "improved" version of the protocol of Chien *et al.* [1] that inspired us; b) a random authentication attack is possible on [27] which is a 2-factor authentication also inspired by Chien *et al.* [1] but patched against algebraic attack; c) a chosen value attack is possible on Gossamer [25] if an attacker uses a defective or rogue reader whose he can control the RNG, he can recover ID and thus trace the tags; d) some desynchronization attacks against [26, 28]; e) timing attack may be possible against Chien *et al.* [22] and Peris *et al.* [25] that both use a secret data as parameter of a "rotate" function whose implementation may leak information on the secret data; f) Chien *et al.* [22] and Peris *et al.* [25] are both susceptible by design to a collision on the tag pseudonyms computed, even though probability of collision is low if the size l is long enough; g) to be protected against some desynchonization problems during update operation, Song *et al.* [24] and Chien *et al.* [22] must be atomic which is expensive in term of memory and not mentioned in the original paper – so this overhead has not been added into table VI but we have considered them as being sensible to desynchonization. Among all of them, we have only kept 3 from the family of lightweight mutual authentication protocols presented by Song *et al.* [24], Chien *et al.* [22] and Peris *et al.* [25] (Gossamer[2]) since communication or storage costs were close of ours.

One of the key differences of our protocols is the use of $hkSID$ which is loaded on the tag at the enrollment step to maintain the computational cost at the database side during tag authentication at $O(1)$. The only related work claiming to be in $O(1)$ is the last protocol of Chien *et al.* [22]. However, any reader who will briefly analyse the protocol will see that the claim is wrong and the protocol at database side is still in $O(n)$ like other papers.

[2]Contrary to Peris *et al.*, in table VI, the Gossamer protocol is indicated as ultralightweight to lightweight and not ultralightweight only like in [23] because the $MixBit$ function uses the add function which we consider expensive compare to $\oplus$.

Last but not least, our protocols enable tags to have a lifetime not limited by numbers of write in non volatile memory (that store SID, $khSID$ and K) since there is none during authentication. Even though the feature to allow key update is activated in our protocol, the number of writes in EEPROM's cells is significantly less than those of other protocols where there is at least an update of a non volatile memory's cell (EEPROM, Flash or FRAM) at each authentication run. The endurance of each cell of memory being limited to 10^5 writing for EEPROM to 10^{12} for FRAM [29], our protocol is more efficient for applications requiring very frequent authentications (e.g. life of EEPROM is less than 1 month if authentication occurs every 30 seconds).

Note that some results in the table VI are different of those presented in related papers since we take into account all exchanged messages (like the Query, random sent by reader and other information) to give a fair view.

Table VI
COMPARISON OF COSTS WITH OTHER PROTOCOLS.

	Protocols			
	Song *et al.* [24]	Chien *et al.* [22]	Gossamer [23]	Our Proposals
Communication $R \rightarrow T$ (bits)	$40 + 2l$	$40 + l + l/2$	$40 + 3l$	$40 + l$
Communication $T \rightarrow R$ (bits)	$2l$	$1 + 2l + l/2$	$2l$	$1 + 2l$
Total on RF interface (bits)	$41 + 4l$	$41 + 4l$	$40 + 5l$	$41 + 3l$
Memory size on Tag (bits)	l	$3l$	$7l$	$3l$
Memory size on database (bits)	$4l$	$5l$	$4l$	$2l$
Computational cost on database	$O(n)$	$O(n)$	$O(n)$	$O(1)$
Classification of computational cost on tag	Lightweight	Lightweight	Ultralightweight to Lightweight	Lightweight
Lifetime of tag in number of authentications	10^5 or 10^{12}	10^5 or 10^{12}	10^5 or 10^{12}	∞

Table VII provides a comparison of the resistance of our protocols and related work against aforementioned attacks.

Table VII
SECURITY COMPARISON WITH OTHER PROTOCOLS.

	Protocols			
	Song *et al.* [24]	Chien *et al.* [22]	Gossamer [23]	Our Proposals
Untraceability	No [8, 9]	Yes	Yes	Yes
Mutual authentication	No*	Yes	Yes	Yes
Algebraic Attack Resistant	No [9]	Yes	Yes	Yes
Desynch. & DoS Attacks Resistant	No [8, 9, 30]	No	No [31, 32]	N/A (Yes)
Database Entry Collision Resistant	N/A (Yes)	No	No [33]	N/A (Yes)
Chosen Value Attack Resistant	$\triangledown$	Yes	ImpDep	Yes
Timing Attack Resistant	$\triangledown$	ImpDep	ImpDep	N/A (Yes)
Random Authentication Attack Resistant	$\triangledown$	Yes	Yes	Yes
Database Impersonation Attack Resistant	No [8, 9, 34]	Yes	Yes	Yes
Man-in-the-middle Attack Resistant	No [8, 9]	Yes	$\triangledown$	Yes
Replay Attack Resistant	No [8, 9]	Yes	No [33]	Yes

$\triangledown$: Considered as irrelevant to resist to this attack since another attack exists with a higher probability of success

N/A: Not Applicable (thus the protocol is resistant) – No*: No since an attack exists – ImpDep: Implementation Dependent

IX. CONCLUSIONS

In this paper we have shown that our protocols fulfill both the functional and security requirements for low-cost tags. Our proposals are practicable with the real world's constraints since the computational cost at database side being $O(1)$ against $O(n)$ for the related work, transaction time is improved.

In addition, our protocols are flexible and can be modified according to different goals:

- for tag authentication to the database, only steps 1 and 2 are sufficient. Indeed some applications do not require mutual authentication (for instance, in stock management, database just needs to know that a tag has been read and then it could carry some internal operations like removal from the current stock).
- for a mutual authentication between tag and database without revealing information about tag to reader, steps 1, 2 and 3 are needed.
- for the disclosure of the tag's identity to the reader, step 4 can be added. The disclosure of this identity could be made according to a predefined policy: a) by the database which reveals the SID to the reader, or b) by the tag (with the OK/NOK message) which reveals itself its SID.

Finally, the protocol suite presented in this paper is flexible and more secure and efficient than related work whilst ensuring privacy-protection.

REFERENCES

[1] H.-Y. Chien and C.-W. Huang, "A Lightweight RFID Protocol Using Substring," in *Embedded and Ubiquitous Computing*, ser. Lecture Notes in Computer Science, T.-W. Kuo, E. Sha, M. Guo, L. T. Yang, and Z. Shao, Eds. Taipei, Taiwan: Springer Berlin Heidelberg, December 2007, vol. 4808, pp. 422–431. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77092-3_37

[2] Y.-Z. Li, Y.-B. Cho, N.-K. Um, and S.-H. Lee, "Security and Privacy on Authentication Protocol for Low-Cost RFID," in *Computational Intelligence and Security*, ser. Lecture Notes in Computer Science, Y. Wang, Y.-m. Cheung, and H. Liu, Eds. Guangzhou, China: Springer Berlin Heidelberg, November 2007, vol. 4456, pp. 788–794. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74377-4_82

[3] B. Danev, T. S. Heydt-Benjamin, and S. Čapkun, "Physical-layer Identification of RFID Devices," in *Proceedings of the 18th Conference on USENIX Security Symposium*, ser. SSYM'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 199–214. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855768.1855781

[4] G. P. Hancke, "Practical Eavesdropping and Skimming Attacks on High-frequency RFID Tokens," *J. Comput. Secur.*, vol. 19, no. 2, pp. 259–288, Apr. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1971859.1971863

[5] Y. Oren, D. Schirman, and A. Wool, "Range extension attacks on contactless smart cards," in *Computer Security ESORICS 2013*, ser. Lecture Notes in Computer Science, J. Crampton, S. Jajodia, and K. Mayes, Eds. Egham, UK: Springer Berlin Heidelberg, September 2013, vol. 8134, pp. 646–663. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40203-6_36

[6] ——, "RFID Jamming and Attacks on Israeli e-Voting," in *Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2012 European Conference on*, June 2012, pp. 1–7.

[7] A. Juels, R. L. Rivest, and M. Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ser. CCS '03. New York, NY, USA: ACM, 2003, pp. 103–111. [Online]. Available: http://doi.acm.org/10.1145/948109.948126

[8] T. van Deursen and S. Radomirovi, "Algebraic Attacks on RFID Protocols," in *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks*, ser. Lecture Notes in Computer Science, O. Markowitch, A. Bilas, J.-H. Hoepman, C. Mitchell, and J.-J. Quisquater, Eds. Brussels, Belgium: Springer Berlin Heidelberg, September 2009, vol. 5746, pp. 38–51. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03944-7_4

[9] ——, "Attacks on RFID Protocols," Cryptology ePrint Archive, Report 2008/310, 2008, http://eprint.iacr.org/.

[10] H.-Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 4, pp. 337–340, Oct 2007.

[11] D. Dolev and A. C. Yao, "On the security of public key protocols," *Information Theory, IEEE Transactions on*, vol. 29, no. 2, pp. 198–208, Mar 1983.

[12] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "LAMED - A PRNG for EPC Class-1 Generation-2 RFID Specification," *Comput. Stand. Interfaces*, vol. 31, no. 1, pp. 88–97, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2007.11.013

[13] H. Martin, E. San Millan, L. Entrena, P. Lopez, and J. Castro, "AKARI-X: A pseudorandom number generator for secure lightweight systems," in *On-Line Testing Symposium (IOLTS), 2011 IEEE 17th International*, July 2011, pp. 228–233.

[14] F. Arnault, T. P. Berger, and B. Pousse, "A matrix approach for FCSR automata," *Cryptography and Communications*, vol. 3, no. 2, pp. 109–139, 2011. [Online]. Available: http://dx.doi.org/10.1007/s12095-010-0041-z

[15] F. Arnault, T. Berger, C. Lauradoux, M. Minier, and B. Pousse, "A new approach for fcsrs," in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science, J. Jacobson, MichaelJ., V. Rijmen, and R. Safavi-Naini, Eds. Calgary, Canada: Springer Berlin Heidelberg, August 2009, vol. 5867, pp. 433–448. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-05445-7_27

[16] T. Deursen, S. Mauw, and S. Radomirovi, "Untraceability of RFID Protocols," in *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, ser. Lecture Notes in Computer Science, J. A. Onieva, D. Sauveron, S. Chaumette, D. Gollmann, and K. Markantonakis, Eds. Seville, Spain: Springer Berlin Heidelberg, May 2008, vol. 5019, pp. 1–15. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-79966-5_1

[17] G. P. Hancke and M. G. Kuhn, "An RFID Distance Bounding Protocol," in *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, ser. SECURECOMM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 67–73. [Online]. Available: http://dx.doi.org/10.1109/SECURECOMM.2005.56

[18] G. P. Hancke, "Design of a Secure Distance-bounding Channel for RFID," *J. Netw. Comput. Appl.*, vol. 34, no. 3, pp. 877–887, May 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2010.04.014

[19] S. Karda, M. Kiraz, M. Bingöl, and H. Demirci, "A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions," in *RFID. Security and Privacy*, ser. Lecture Notes in Computer Science, A. Juels and C. Paar, Eds. Amherst, USA: Springer Berlin Heidelberg, June 2012, vol. 7055, pp. 78–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25286-0_6

[20] D. Ma, N. Saxena, T. Xiang, and Y. Zhu, "Location-Aware and Safer Cards: Enhancing RFID Security and Privacy via Location Sensing," *Dependable and Secure Computing, IEEE Transactions on*, vol. 10, no. 2, pp. 57–69, March 2013.

[21] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Advances in Cryptology CRYPTO 96*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed. Springer Berlin Heidelberg, August 1996, vol. 1109, pp. 104–113. [Online]. Available: http://dx.doi.org/10.1007/3-540-68697-5_9

[22] H.-Y. Chien and C.-W. Huang, "A Lightweight Authentication Protocol for Low-Cost RFID," *Journal of Signal Processing Systems*, vol. 59, no. 1, pp. 95–102, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11265-008-0281-8

[23] 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 *Information Security Applications*, ser. Lecture Notes in Computer Science, K.-I. Chung, K. Sohn, and M. Yung, Eds. Jeju Island, Korea: Springer Berlin Heidelberg, September 2009, vol. 5379, pp. 56–68. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00306-6_5

[24] B. Song and C. J. Mitchell, "RFID Authentication Protocol for Low-cost Tags," in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec '08. New York, NY, USA: ACM, 2008, pp. 140–147. [Online]. Available: http://doi.acm.org/10.1145/1352533.1352556

[25] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "EMAP: An Efficient Mutual-Authentication Protocol for Low-Cost RFID Tags," in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, ser. Lecture Notes in Computer Science, R. Meersman, Z. Tari, and P. Herrero, Eds. Montpellier, France: Springer Berlin Heidelberg, November 2006, vol. 4277, pp. 352–361. [Online]. Available: http://dx.doi.org/10.1007/11915034_59

[26] H. Lei, G. Yong, C. Zeng-yu, and L. Na-Na, "An Improved Lightweight RFID Protocol Using Substring," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, Beijing, China, Sept 2009, pp. 1–4.

[27] N. Chikouche, C. Foudil, and M. Benmohammed, "An Authentication Protocol Based on Combined RFID-Biometric System RFID-Biometric System," *International Journal of Advanced Computer Science and Applications*, vol. 3, 2012.

[28] I. Vajda and L. Buttyán, "Lightweight Authentication Protocols for Low-Cost RFID Tags," in *In Second Workshop on Security in Ubiquitous Computing Ubicomp 2003*, Seattle, Washington, USA, October 2003.

[29] S. Kawashima and J. S. Cross, "Feram," in *Embedded Memories for Nano-Scale VLSIs*, ser. Integrated Circuits and Systems, K. Zhang, Ed. Springer US, 2009, pp. 279–328. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-88497-4_8

[30] S. Abughazalah, K. Markantonakis, and K. Mayes, "A vulnerability in the song authentication protocol for low-cost rfid tags," in *Security and Privacy Protection in Information Processing Systems*, ser. IFIP Advances in Information and Communication Technology, L. Janczewski, H. Wolfe, and S. Shenoi, Eds. Springer Berlin Heidelberg, 2013, vol. 405, pp. 102–110. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39218-4_8

[31] D. Tagra, M. Rahman, and S. Sampalli, "Technique for preventing DoS attacks on RFID systems," in *Software, Telecommunications and Computer Networks (SoftCOM), 2010 International Conference on*, Sept 2010, pp. 6–10.

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

[33] Z. Bilal, A. Masood, and F. Kausar, "Security Analysis of Ultra-lightweight Cryptographic Protocol for Low-cost RFID Tags: Gossamer Protocol," in *Network-Based Information Systems, 2009. NBIS '09. International Conference on*, Aug 2009, pp. 260–267.

[34] P. Rizomiliotis, E. Rekleitis, and S. Gritzalis, "Security analysis of the song-mitchell authentication protocol for low-cost RFID tags," *Communications Letters, IEEE*, vol. 13, no. 4, pp. 274–276, April 2009.