Ultralightweight Cryptography for passive RFID systems

Umar Mujahid, M.Najam-ul-islam, Jameel Ahmed

Department of Electrical Engineering, Bahria University Islamabad

Abstract

RFID (Radio Frequency Identification) is one of the most growing technologies among the pervasive systems. Non line of sight capability makes RFID systems much faster than its other contending systems such as barcodes and magnetic taps etc. But there are some allied security apprehensions with RFID systems. RFID security has been acquired a lot of attention in last few years as evinced by the large number of publications (over 2000).

In this paper, a brief survey of eminent ultralightweight authentication protocols has been presented & then a four-layer security model, which comprises of various passive and active attacks, has been proposed. Cryptanalysis of these protocols has also been performed under the implications of the proposed security model.

Introduction:

Radio Frequency Identification (RFID) is broad concept of closed loop wireless networking between Main node (Reader) and small nodes (Tags) providing automatic identification of nodes present in the vicinity of main node. In RFID systems, there are mainly three characters: Reader, Tag and database server. Tags are sort of transponders, which contain a small amount of memory (for identity of the attached object and other relevant function) and on board circuitry including transceiver, the readers are just like scanners, which read the contents of the tags and then match these contents with entries at the database for identification. We normally assume the link between reader and back end database is secure as there is no power computation issue, so we can incorporate various security relevant solutions. Link between tag and reader needs more attention as this is wireless link and adversaries can have easy access to this link. As, we also have very limited resources at the tag end, so to make RFID system practically feasible we have to reduce the cost of the tag and then within these limited resources we also have to address these security issues. By keeping in view of all these limitations, a new field of cryptography known as ultralightweight cryptography had been introduced back in 2006. This field specifically had been introduced for low cost RFID tags to make them applicable and comparable with its contending systems. For low cost passive RFID tags, we can use only 5-10 K gates and among which 250-3000 gates are devoted for security (Cryptography)[1].

The main objective of this sort of cryptography is to provide the secure mutual authentication between reader and tag in a cost effective way. Because of this cost effectiveness this type of cryptography is known as ultralightweight cryptography and associated protocols are known as ultralightweight mutual authentication protocols (UMAP). These protocols consist of simple bit wise operations like XOR, OR, AND etc, as other cryptographic functions like one-way hash functions MD5 and SHA-256, respectively require 8K and 11 K logical gates, which makes them practically unfeasible. In this paper, we will first discuss the major protocols from UMAP family, and then run these protocols through four-layer security model. This security model will assess the authenticity of the protocols; by applying various cryptanalysis tests/ attacks. The paper is organized as follows: In section I, we introduce the UMAP protocols, and then in section II attributes of the proposed security model have been discussed. In section III, cryptanalysis on the basis of security model has been presented. Finally, performance analysis of discussed protocols has been presented to evaluate the protocols.

Section I

Ultralightweight mutual authentication Protocols:

Mutual authentication protocols provide corroboration to both tag and reader that they are communicating with valid reader/tag.

Chein [3] presented classifications of authentication protocols based on cryptographic functions that can be used at Tag's end.

Full-fledged: This is the most powerful class of mutual authentication in which we can incorporate traditional cryptographical solutions such as symmetric encryption, one-way Hash functions and even public key cryptography.

Simple: This class is weaker as compared to full-fledged class because we can only use pseudorandom number generator and one-way hash functions.

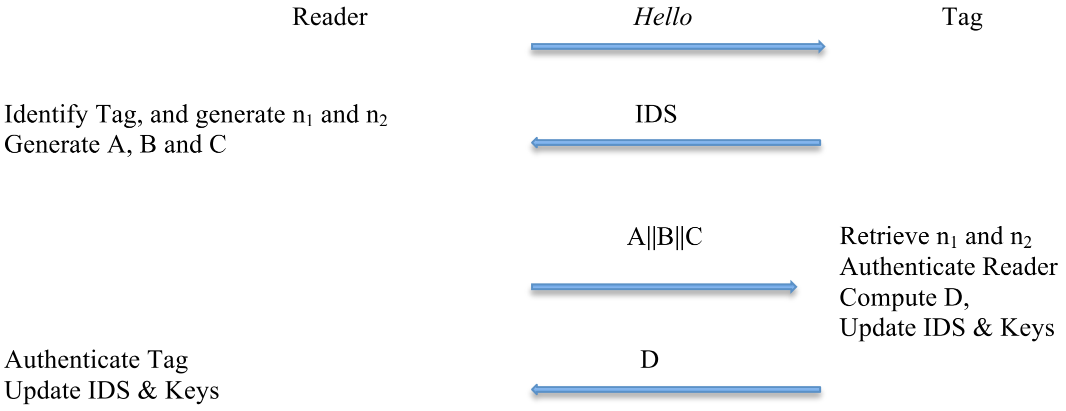
Lightweight: This class is even weaker than simple authentication protocols; in this class we can use lightweight pseudorandom number generators and some simple functions such as Cyclic Redundancy Check (CRC) but no hash functions at tag side.

Ultralightweight: This is the weakest class; we cannot incorporate even pseudorandom number generators at tag end. We can only use simple bitwise XOR, OR, AND etc. logical functions. So, randomness can only be generated from readers. Rest of the research paper will be focused on the applications and working of this category.

Recently, there has been proposed several ultralightweight RFID authentication protocols. The basic operation of the protocols involves exchange of pseudonyms such as IDS (Identity pseudonym) and keys between reader and tags. The original identity conceals within the message comprises of logical operations between pseudonyms and original values. Normally, a random number is transmitted by reader towards tag because of power computation issues at tag's end. This random number provides or we may say enhances the diffusion property of the protocol. Then after each successful authentication session both reader and tag update their pseudonyms using comparable equations at both ends. To avoid the Desynchronization attacks some protocols provide the room for storage of old pseudonyms. Protocols using this approach are: LMAP [1] (2006), EMAP,[2] (2006), SASI [3] (2007), GOSSAMER [5] (2009), David-Prasad [8] (2009) and RAPP [7] (2012). They all are relatively new and designed empirically, and most of them are wrecked, as we will discuss in later section. Some assumptions have been made for our research, which will be applicable for all protocols to be discussed; firstly the length of the all keys, Pseudonyms and other identifiers is 96 bits as per EPC global standard [17]. Secondly, we will consider the channel between reader and backend database a secure one and our research will be focused to make the channel between reader and tag as secure as possible.

LMAP:

Lightweight Mutual Authentication protocol (LMAP) [1] was the first proposal in the UMAP family presented in 2006. The protocol is divided into four main stages: Tag identification, Mutual authentication, index-pseudonym updating and key updating. Tag stores one constant and five variable values each of 96 bits, in which ID will remain constant while IDS and four other keys K_1, K_2, K_3, K_4 are variables that will be updated in a well synchronized manner after each successful authentication protocol run.



[Pseudonym Updating both Tag and Reader]

$$IDS^{n+1} = (IDS^n + (n_2{}^n + K_4{}^n))\oplus ID$$
$$K_1{}^{n+1} = K_1{}^n \oplus n_2{}^n \oplus (K_3{}^n + ID)$$
$$K_2{}^{n+1} = K_2{}^n \oplus n_2{}^n \oplus (K_4{}^n + ID)$$
$$K_3{}^{n+1} = K_3{}^n \oplus n_1{}^n \oplus (K_1{}^n \oplus ID),\ K_4{}^{n+1} = K_4{}^n \oplus n_1{}^n + (K_2{}^n \oplus ID)$$

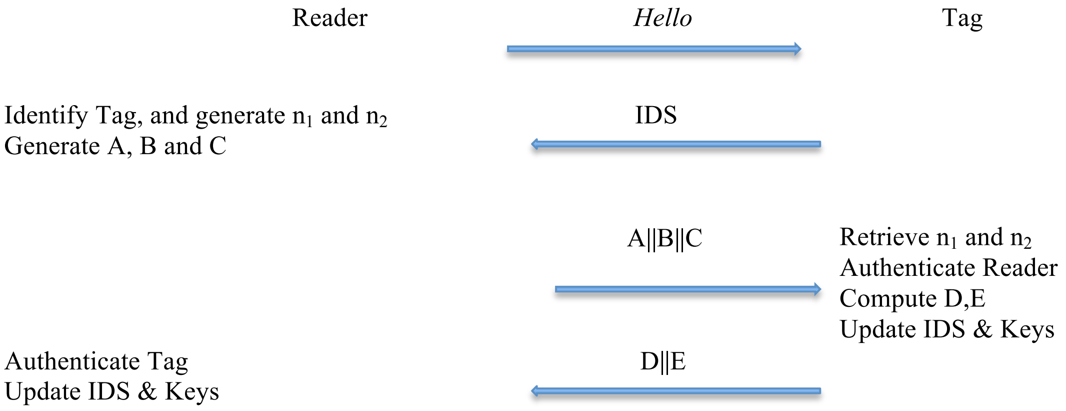
$$A = IDS \oplus K_1 \oplus n_1$$
$$B = (IDS \vee K_2) + n_1$$
$$C = IDS + K_3 + n_2$$
$$D = (IDS + ID) \oplus n_1 \oplus n_2$$

Fig.1 LMAP Protocol

In LMAP reader initiates the protocol by transmitting the Hello message towards tag, and tag will then reply with its IDS. Reader compares the received IDS with its database and if it matches with its entry then reader will generate two random number n_1 and n_2 and conceals these numbers within the messages A, B and C which have already been shown in the above table. Reader concatenates and transmits these combinational messages towards tag. The tag will then retrieve the random numbers $n_1$and n_2 from the messages and calculates B using the same synchronized equation, compares this B with received B if a match occurs it means tag is communicating with a valid reader and then tag will update its pseudonyms (IDS, K_1, K_2, K_3, K_4). Now, tag transmits D message towards reader from which reader authenticates tag and then after successful tag authentication reader will also update its Pseudonyms (IDS, K_1, K_2, K_3, K_4). The authors estimated that for implementation of protocol requires only 1000 logic gates, which fulfils the requirements for a protocol to be considered as ultralightweight. But protocol doesn't prosper in averting even basic traceability and information leakage attacks.

EMAP:

Efficient mutual authentication protocol (EMAP)[2] was another protocol from UMAP family. Here a new Parity function F_p was introduced, which is introduced, as vector built from the parity bits and the rest was quite similar to LMAP. Reader initiates the protocol by transmitting a 'Hello' message towards tag and tag responds with its current IDS. Reader matches the received IDS with its database if a match occurs then reader will generate two random numbers n_1 & n_2 and conceals these random numbers within messages A, B and C. Reader transmits these messages towards tag then tag retrieves these random numbers from A and C. Tag will calculate local value of B and compares it with received B, if successful match occurs tag will first update its pseudonyms and then tag generates D and E. After receiving D and E reader will also calculate local values of D and E and compares them with received ones, if a match occurs reader will update its Pseudonyms in the same fashion as tag which have been described as follows:



[Pseudonym Updating both Tag and Reader]

$$IDS^{n+1} = (IDS^n \oplus n_2^n \oplus K_1^n$$
$$K_1{}^{n+1} = K_1{}^n \oplus n_2{}^n \oplus ([ID]_{0:47} || F_P(K_4^n) || F_P(K_3^n))$$
$$K_2{}^{n+1} = K_2{}^n \oplus n_2{}^n \oplus (F_P(K_1^n) || F_P(K_4^n) || [ID]_{48:95})$$
$$K_3{}^{n+1} = K_3{}^n \oplus n_1{}^n \oplus (K_1{}^n \oplus ID),$$

$$A = IDS \oplus K_1 \oplus n_1$$
$$B = (IDS \vee K_2) \oplus n_1$$
$$C = IDS \oplus K_3 \oplus n_2$$
$$D = (IDS \wedge K_4) \oplus n_2$$

$$K_4{}^{n+1} = K_4{}^{n} \oplus n_1{}^{n} \oplus (F_P(K_3^n)\,||\,F_P(K_1^n)\,||\,[ID]_{48:95})$$

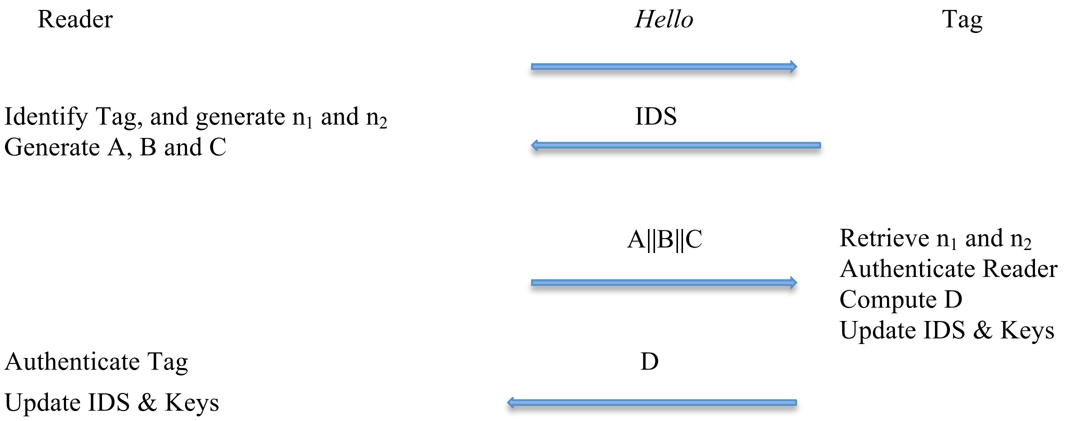
$$E = ((IDS \wedge n_1) \vee n_2) \oplus \mathrm{ID} \oplus K_1 \oplus K_2 \oplus K_3 \oplus K_4$$

Fig.2 EMAP Protocol

EMAP requires 500 logic gates for implementation, which is much lighter than any other mutual authentication protocol. But again recent cryptanalysis on EMAP has found a lot of security threats and vulnerabilities in the protocol, which made it highly unsuitable for practical systems. These attacks and threats will be discussed in the next section.

SASI:

Chein presented a new ultralightweight mutual authentication protocol SASI [3] (Strong authentication and integrity) in 2007. This protocol has similar operational structure as proposed in LMAP and EMAP, but here a new function Rot (Left cyclic Rotation) has been introduced in SASI, which was quite different from Triangular functions (XOR, OR etc.) extensively used in previous protocols, as these triangular functions have congenital poor diffusion properties. The use of non-triangular function makes this protocol a unique one as compared to its contending protocols. The basic working of SASI protocol is as follows:



Reader

Hello

Tag

Identify Tag, and generate n_1 and n_2
Generate A, B and C

IDS

A||B||C

Retrieve n_1 and n_2
Authenticate Reader
Compute D
Update IDS & Keys

Authenticate Tag
Update IDS & Keys

D

Pseudonym updating and key updating:
$IDS = (IDS + ID) \oplus (n_2 \oplus K_1^*);\ K_1 = K_1^*; K_2 = K_2^*$

$$A = IDS \oplus K_1 \oplus n_1$$
$$B = (IDS \vee K_1) \oplus n_2$$
$$K_1^* = Rot(K_1 \oplus n_2, K_1)$$

$$K_2^* = Rot(K_2 \oplus n_1, K_2)$$

$$C = (K_1 \oplus K_2^*) + (K_1^* \oplus K_2)$$

$$D = (K_2^* + ID) \oplus ((K_1 \oplus K_2) \vee K_1^*)$$

Fig.3 SASI Protocol

In SASI reader initiates the protocol by sending a '*Hello'* message towards tag. Tag then responds with its current IDS. Reader matches IDS with its database if received IDS is different then reader matches this with old IDS (To avoid Desynchronization attack); on a successful match reader generates and transmits A, B & C towards tag. To enhance diffusion properties of the communication, reader generates pseudo random numbers and conceals them with messages (A, B &C), which are as follows:

$A = IDS \oplus K_1 \oplus n_1$
$B = (IDS \vee K_2) \oplus n_2$
$C = (K_1 \oplus K_2^*) + (K_1^* \oplus K_2)$

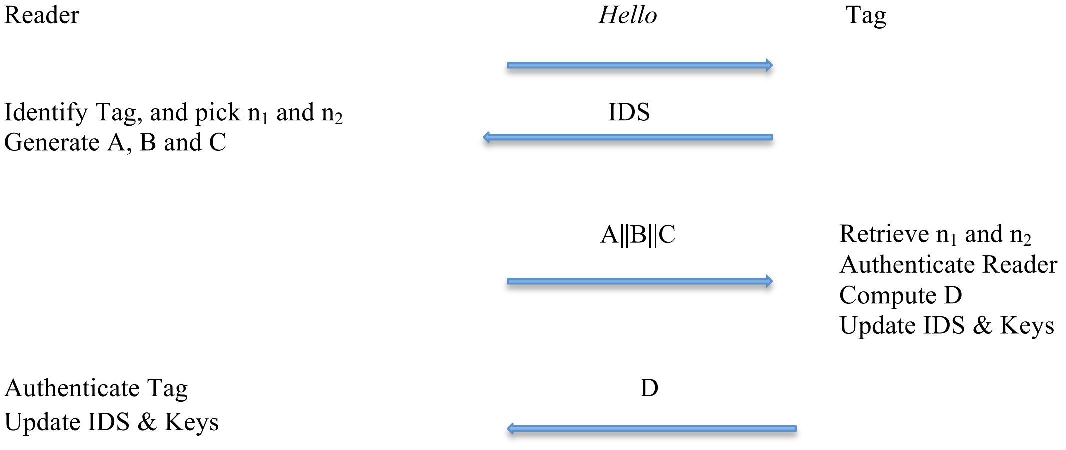
On receiving of $A||B||C$, tag extracts n_1 from A and n_2 from B. Then by using these new random numbers tag generates $K_1^*\& K_2^*$ using above mentioned equations. Tag calculates the local value of C and compares it with received C, if a match occurs then it means tag is communicating with genuine reader. Tag then updates its pseudonyms and generates D, so reader can also authenticate tag. After receiving D reader will verify the received D and updates its pseudonyms. Again here update process is similar except back ups of pseudonyms to prevent against Desynchronization attacks, but still Desynchronization is possible with repeatedly interrupting the message D. This will be discussed in detail in next section.

The Gossamer Protocol:

In 2009, Peris-Lopez et.al proposed a new ultralightweight mutual authentication protocol: GOSSAMER [5] . The basic working of the protocol was again similar to other previously proposed protocols but in Gossamer, they incorporated two new functions double Rotation and MixBits. The internal structure of these functions consists of same traditional triangular functions (Shifting &Addition) but have more robust diffusion properties as compare to uncluttered triangular function. MixBits is a function based on genetic programming and extremely lightweight in nature, as there are only bitwise right shifts (>>) and additions are employed. To calculate, $Z = MixBits(X, Y)$ pseudo code is as follows:

```
$Z = X;$
$for(i = 0; i < 32; i++)$
$\{Z = (Z \gg 1) + Z + Z + Y\ ;\ \}$
```

From above, working of the algorithm can be seen as initially equate Z as per the value of the X, then give one right cyclic bitwise shift in Z (string). Add Z with the shifted version of Z, and then add this with Z+Y. This will give us the MixBits composite string. The Gossamer protocol is as follows:



Reader | *Hello* | Tag

Identify Tag, and pick n_1 and n_2
Generate A, B and C | IDS

A||B||C | Retrieve n_1 and n_2
Authenticate Reader
Compute D
Update IDS & Keys

Authenticate Tag
Update IDS & Keys | D

Pseudonym updating and key updating:

$IDS = (IDS + ID) \oplus (n_2 \oplus K_1^*);\ K_1 = K_1^*; K_2 = K_2^*$

$$A = Rot(Rot(IDS + K_1 + \pi + n_1, K_2) + K_1, K_1)$$

$$B = Rot(Rot(IDS + K_2 + \pi + n_2, K_1) + K_2, K_2)$$

$$C = Rot(Rot(n_3 + K_1^* + \pi + n_1', n_3) + K_3^* \oplus n_1', n_2) \oplus n_1'$$

$$n_3 = MixBits(n_1, n_2)$$

$$K_1^* = Rot(Rot(n_2 + K_1 + \pi + n_3, n_2) + K_2 \oplus n_3, n_1) \oplus n_3$$

$$K_2^* = Rot(Rot(n_2 + K_1 + \pi + n_3, n_1) + K_1 + n_3, n_2) + n_3$$

$$n_1' = MixBits(n_3, n_2)$$

$$D = Rot(Rot(n_1 + K_2^* + ID + n_1', n_2) + K_1^* + n_1', n_3) + n_1'$$

$$n_2' = MixBits(n_1', n_3)$$

$$IDS^{next} = Rot(Rot(n_1' + K_1' + IDS + n_2', n_1') + K_2^* \oplus n_2', n_3) \oplus n_2'$$

Fig.4 GOASSMER Protocol

The protocol works in the same fashion as we have already discussed in the other protocols. Reader initiates the protocol by transmitting a message signal "Hello", tag responds with its current updated IDS. On receiving this IDS, reader compares this with its database; if a match occurs then it further sends the concatenated message A||B||C, which are defined as:

$$A = Rot(Rot(IDS + K_1 + \pi + n_1, K_2) + K_1, K_1)$$

$$B = Rot(Rot(IDS + K_2 + \pi + n_2, K_1) + K_2, K_2)$$

$$C = Rot(Rot(n_3 + K_1^* + \pi + n_1', n_3) + K_2^* \oplus n_1', n_2) \oplus n_1'$$

Here π is the 96-bit constant value, and K_1^* and K_2^* are as follows:

$$K_1^* = Rot(Rot(n_2 + K_1 + \pi + n_3, n_2) + K_2 \oplus n_3, n_1) \oplus n_3$$

$$K_2^* = Rot(Rot(n_2 + K_1 + \pi + n_3, n_1) + K_1 + n_3, n_2) + n_3$$

If IDS doesn't match with the entries of database, then reader will send hello message again towards tag to resend its old IDS. After successful matching and receiving of concatenated messages, tag computes n_1', n_3 & K_1' for calculation of C. It then compares the calculated C with received C; if a match occurs then tag will perform three tasks. Firstly, it computes D message, and then transmit the message towards reader. Thirdly it also updates its Pseudonyms as reader has been successfully authenticated in the previous step.

$$D = Rot(Rot(n_1 + K_2^* + ID + n_1', n_2) + K_1^* + n_1', n_3) + n_1'$$

Reader will also calculate the local version of D &compare it with received D; on successful matching reader will also update its Pseudonyms for future correspondence.

This protocol is more sophisticated then other protocols of UMAP family, as there is no full disclosure attack available, which can break Gossamer.

David-Prasad Protocol:

In September 2009, David and Prasad [8] proposed a new ultralightweight mutual authentication protocol for passive low cost RFID tags. The basic working principle of the protocol is similar to other contending protocols of UMAP family, SASI and Gossamer. The main aim of the protocol was to provide the security within limited resources (Hardware and power computation). It also includes the storage of previous value of IDS to counter measure against Desynchronization attacks. In David-Prasad protocol, before inquiring tags; reader have to get a one-day certificate from CA (Certificate authority) after authenticating himself. Reader initiates the protocol by transmitting the message

"*Hello*" towards tag. Tag then responds with its current updated IDS, reader matches this IDS with its database; if a match occurs it produces two nonces (n_1, n_2), computes and then transmits messages A, B and D, which are as follows:

$A = (IDS \wedge K_1 \wedge K_2) \oplus n_1$

$B = (IDS' \wedge K_2 \wedge K_1) \oplus n_2$

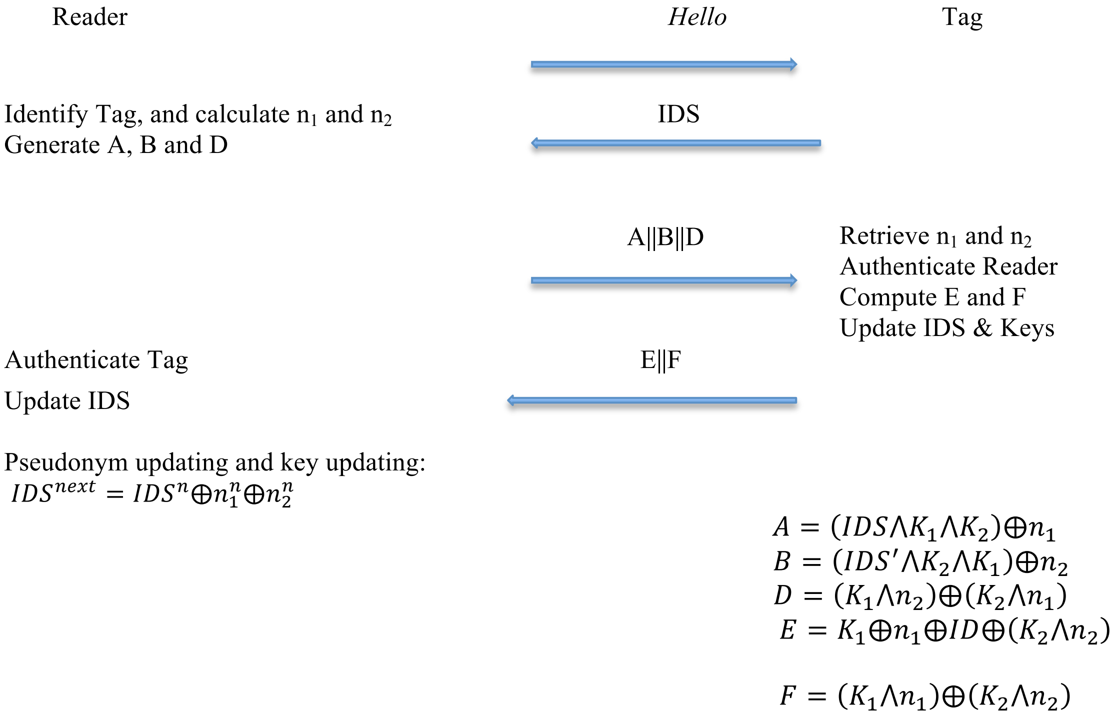
$D = (K_1 \wedge n_2) \oplus (K_2 \wedge n_1)$

Tag then extracts nonces (n_1, n_2) and computes a local value of D. It then compares locally generated D with received one, on successful matching tag updates it pseudonyms, computes and transmits E and F towards reader.

$E = K_1 \oplus n_1 \oplus ID \oplus (K_2 \wedge n_2)$

$F = (K_1 \wedge n_1) \oplus (K_2 \wedge n_2)$

Reader also generates the local values of E and F, compares these values with received ones, after successful matching it will update its pseudonyms and terminate the protocol.



Reader
Hello
Tag
Identify Tag, and calculate n_1 and n_2
Generate A, B and D
IDS
A||B||D
Retrieve n_1 and n_2
Authenticate Reader
Compute E and F
Update IDS & Keys
Authenticate Tag
Update IDS
E||F
Pseudonym updating and key updating:
$IDS^{next} = IDS^{n} \oplus n_1^{n} \oplus n_2^{n}$
$A = (IDS \wedge K_1 \wedge K_2) \oplus n_1$
$B = (IDS' \wedge K_2 \wedge K_1) \oplus n_2$
$D = (K_1 \wedge n_2) \oplus (K_2 \wedge n_1)$
$E = K_1 \oplus n_1 \oplus ID \oplus (K_2 \wedge n_2)$
$F = (K_1 \wedge n_1) \oplus (K_2 \wedge n_2)$

Fig.5 *David-Prasad Protocol*

RAPP Protocol:

Yun Tian, Gongliang et.al proposed a new ultralightweight RFID mutual authentication protocol with permutation (RAPP) [7] in 2012. RAPP introduces a new function permutation; which have been incorporated with XOR operation in all equations. The usage of permutation in RAPP avoids the usage of unbalanced AND & OR operations. RAPP uses only three operations; Bitwise XOR, left rotation, and permutation. Permutation operation is ultralightweight in nature as it involves only bitwise shifting of bit. The rudimentary working of permutation involves the generation of new string based on shifting the bits position of second string with respect to the entry at first string. It means if first entry in first string is 0 then first bit of second string will be shifted to the last position in third string or vice versa. Let say, A=1011101 & B=0111010 then Per (A, B)=0110011.

In RAPP protocol, tag stores four values (Strings) IDS, K_1, K_2, & K_3 (each is of 96-bit long). Reader also stores the same variables, but to avoid Desynchronization attacks in addition to current pseudonyms it also stores the old values of these pseudonyms. Reader initiates the protocol while sending a Hello message towards tag. Upon receiving the reader's probe, tag transmits its current IDS to the reader. After receiving IDS, reader uses it as an index to search a corresponding record in the database. If IDS is old one then reader uses Old values of pseudonyms to calculate A and B message integrated with 96-bit random number n_1, otherwise vice versa. After calculating A and B, reader then transmits these messages toward tag. Where A and B are as follows:

$$A = Per(K_2, K_1) \oplus n_1$$

$$B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1)$$

Tag extracts n_1 message from A and calculate the local version of B. If local value of B and received B are same then tag transmits C message towards reader.

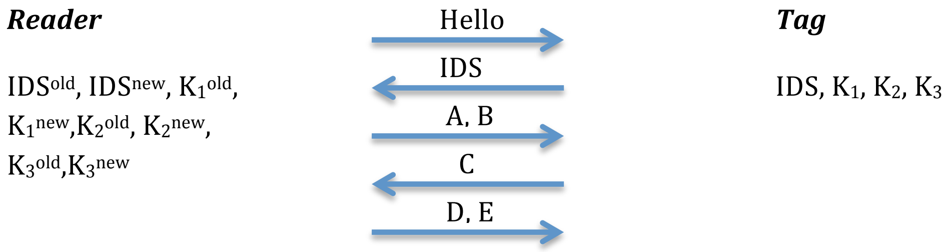
$$C = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$$

When reader receive C message, it will compare it with local C, again if a match occurs then it will generate another L-bit pseudonym n_2. Both n_1 and n_2 will be used for key update. The reader calculates D and E messages and transmits them towards tag. Then reader also updates its pseudonyms for future correspondence with the particular tag.

$$D = Per(K_3, K_2) \oplus n_2$$

$$E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$$

Tag extracts n_2 from D and computes local value of E. If locally calculated value of E is same as received one then tag also updates its pseudonyms and terminate the link. The protocol is as follows:



Reader
Hello
Tag
IDS
IDS^{old}, IDS^{new}, K_1^{old}, K_1^{new}, K_2^{old}, K_2^{new}, K_3^{old}, K_3^{new}
IDS, K_1, K_2, K_3
A, B
C
D, E

Pseudonyms updating:

$$IDS^{new} = Per\left(IDS^{old}, n_1 \oplus n_2\right) \oplus K_1^{old} \oplus K_2^{old} \oplus K_3^{old}$$
$$K_1^{new} = Per(K_1^{old}, n_1) \oplus K_2^{old}$$
$$K_2^{new} = Per(K_2^{old}, n_1) \oplus K_1^{old}$$

$$K_3^{new} = Per(K_3^{old}, n_1 \oplus n_2) \oplus IDS^{old}$$

Fig.6 RAPP Protocol

Section II

Security Model for cryptanalysis of the UMAP Protocols:

The security of the protocol can be analyzed in two major aspects: the functionality of the protocols and confrontation against attacks. The functionality of the protocols comprises of mutual authentication, data confidentiality, data integration, tag anonymity and untraceibility. While Desynchronization, tango and replay fall in the attack category.

Mutual Authentication: Mutual authentication is basic and essential operation of the protocol, in which the tag authenticate the reader and reader authenticates tag. This bidirectional operation validates that either reader or tag are communicating with a genuine one or not.

Data confidentiality: Data confidentiality is another integral parameter; which depicts the confidentiality of the transmitted data between tag and reader.

Data Integrity: In data integrity, if an adversary alters the information; which was transmitted between tag and reader, then to maintain data integrity the protocol should detect the error.

Tag anonymity & Untraceibility: This is also very important parameter, as if an adversary successfully identifies a particular tag; then the particular tag can be traced out easily. It means its mobility can be under observation; which is prevalent security menace.

On the basis of some renowned attacks [4],[6],[8],[9],[11],[12],[13], [14],[15], [16] we have proposed a security model; a protocol can be considered a reliable one if it satisfies all the layers of the model. Security model is as follows:

Serial no.	Security Analysis/Attacks	Adversaries capabilities
1	Desynchronization attacks	i) Man in middle ii) Communication blocking
2	Traceability attacks	i) Man in middle ii) Communication blocking
3	Full Disclosure attacks	i) Eavesdropping
4	General Adhoc attacks	i) Eavesdropping ii) Man in middle iii) Denial of service

Proposed security model contains four-layers, each layer analyze the security vulnerabilities in the protocols by applying the defined mathematical and logical operations.

1. ***Desynchronization***: In this layer, the cryptanalyzers try to break synchronization between the reader and tag. This can be achieved by if an adversary successfully able to tune the genuine reader and tag on different pseudonyms values. We will discuss some practical Desynchronization attacks on the various UMAP protocols in next section.

2. ***Traceability attacks:*** In this layer attackers try to identify the particular tag, so its movement can be recorded. This will be only possible if attacker successfully able to block the pseudonym updating step; so, tag will unable to randomize its IDS.
3. ***Full Disclosure attacks:*** This is the most powerful attack among others as by applying this category, we can disclose all the secrets bearing a protocol. Tango attack is most prominent attack from this category; which needs only a few eavesdrop session to execute its results. Other frame works in full disclosure category are Recursive Linear Cryptanalysis, Differential linear cryptanalysis and Norwegian attacks.
4. ***General adhoc attacks:*** This category basically finds weaknesses in mathematical equations of the protocols to disclose the secrets. We will discuss some probabilistic models to find the secrets of the protocols in next section.

Section III

Security analysis of the protocols:

In this section we will perform security analysis of the various protocols based on proposed security model to validate their practical suitability. As, to make the thing clear in concise manner, we will discuss Desynchronization for all protocols but full disclosure attack and general adhoc attacks only for David-Prasad and SASI Protocols. Because if the protocol fails to satisfy any one of the layers then it will lose it's candidacy for being a Standard UMAP protocol.

a) Security analysis of LMAP & EMAP:

1. Desynchronization attack on LMAP& EMAP:

Desynchronization attack is easily applicable in LMAP, as it doesn't provide the option in the reader for storage of previous IDS value. So, as in LMAP reader initiates the protocol by transmitting a Hello message towards Tag. Tag responds with its Current ***IDS,*** on receiving of IDS reader calculates A, B and C and transmits towards Tag.

$A = IDS \oplus K_1 \oplus n_1$

$B = (IDS \vee K_2) + n_1$

$C = IDS + K_3 + n_2$

As these messages are from a valid reader, so tag generates a message D using n_1 and n_2. But now attacker interrupts the link and block D.

$D = (IDS + ID) \oplus n_1 \oplus n_2$

As a result, tag will update its Pseudonyms but reader will not and it will remain tune up with its previous pseudonyms. Next time when reader transmits Hello message towards this particular tag, then it will respond with such IDS which is quite different from its database. Hence a genuine reader will not communicate with its own tag.

Other security analysis tests of the model can also be applied to protocol; but as it is even unable to resist against a weaker Desynchronization attack, so it cannot be considered as authenticated candidate for practical usage. Same Desynchronization attack is applicable to EMAP as well; In EMAP if we block D and E messages then reader will not able to update its pseudonyms but tag will do. So, this attack in the same manner is applicable to both protocols.

b) Security analysis of SASI protocol:

1. Desynchronization attack [12]:

Lets assume, Reader initiates the protocol and tag responds with IDS. On receiving of IDS from valid tag; reader calculates and transmits A, B and C. Attacker also sniffs these messages and IDS; attacker now perform two operations, make a alias of these messages and block D message. Now as reader didn't receive D message so, it will not able to update its pseudonyms but tag will do. So, tag is tuned on new pseudonyms IDS_2, $K1_1$, $K2_2$.

Next, we allow reader and tag to run the protocol without intervening them. After successful completion of the protocol both database and tag are tuned up on identical values of pseudonyms (IDS_3, $K1_3$, $K2_3$).

Finally, attacker initiates the protocol while pretending itself a valid reader. On receiving of IDS_3, attacker sends an error signal towards tag and asks for IDS_1. Tag immediately responds with IDS_1 and attacker transmits tag pre-captured messages A, B and C (Recorded in previous step). Obviously tag assumes (attacker) a valid reader (as these messages are captured from valid reader's conversation) and transmits D message towards attacker. Now, tag's new pseudonyms are IDS_2, $K1_2$, $K2_2$;which are entirely different from the values stored in database (Which are IDS_3, $K1_3$, $K2_3$).

2. Adhoc/ Probabilistic Attacks[12]:

Let say, reader and tag have completed a successful protocol run but attacker eavesdrops the messages A, B and C during communication. Now, tag and reader's new pseudonyms are IDS_2, $K1_2$ and $K2_2$.

After this attacker initiates protocol with valid tag, by claiming himself a valid reader. On receiving of IDS_2 from tag, attacker asks for IDS_1 (old values) for correspondence. Now, attacker flips the LSB (kth bit) in A, due to which kth value in C message automatically got flipped. On receiving of these altered messages (but in a significant and justified way) tag assumes attacker a valid one, as tag has calculated C from already altered n_1. So, it will transmit D message towards attacker and updates its Pseudonyms (IDS_3,$K1_3K2_3$). Now, next time if a genuine reader wants to communicate with this meticulous tag; it will not find its entry in the database.

c) Security Analysis of David-Prasad Protocol:

Desynchronization attack [8] is again possible on David-Prasad in the same manner, as here you need to block the messages E and F in first run. As a result, reader will not update its pseudonyms in database but tag will do. Attacker sniffs all the important messages (IDS, A, B and D) transmitted during communication.

Next time, attacker allows reader and tag to run the protocol on successful completion of the protocol; both reader and tag updates their pseudonyms accordingly. After this attacker pretends to be genuine reader and initiates the protocol with pre-captured messages. Now, again we will encounter with Desynchronization state. This shows that David-Prasad also doesn't satisfy even the first layer of the model; but to understand full disclosure attack and traceability attack, lets have a look these cryptanalysis for David-Prasad.

1) Adhoc/Probabilistic attacks[8]:
As, we know that XOR & AND operations give unalike results with 75% probability ratio. We can see this thing from the following truth table:

a	b	$a\oplus b$	$a\wedge b$
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Now, by considering the above-mentioned veracity if we perform internal XOR operation of different proposed equations of the security protocols; we can extract some concealed information with certain probability. So, by keeping in view this concept if we take XOR between E and F (David-Prasad messages from Tag) we can find the ID (Secret) of tag with 75% probability of correctness. The operation is as follows:

$$E\oplus F = (K_1\oplus n_1\oplus ID)\oplus(K_2\wedge n_2)\oplus(K_1\wedge n_1)\oplus(K_2\wedge n_2)$$

$$= (K_1\oplus n_1)\oplus ID\oplus(K_2\wedge n_2)$$

As, $(K_1\oplus n_1)\oplus(K_1\wedge n_1)$ always give 1 with 75% probability (because of identical results). So, by using this fact we can easily extract ID of the concerned Tag.

2) *Passive Tango Cryptanalysis [8]:*

Tango attack is among one of the most powerful cryptanalysis, which can recover the secret keys and even ID of the tag. The attack has been divided into two main phases; Selection of Good Approximations & Combination of good approximations.

a) Selection of Good Approximations:
Triangular operations are well known to have very deprived diffusion properties; but in UMAP protocols these operations have been widely used. Now, firstly attacker will have to identify some good approximations (GA) using multiple simple combinations of the exchanged messages (A, B, D, E &F). The GA is based on the closer hamming distance between target and approximations, and compare the number of one's for two consecutive sessions with a threshold value. Here we have mentioned some GA for each of the three secret values on the basis of hamming distances (10000 tests).

Target	Good Approximations (GA)	Hamming distance
K_1	GA-K$_1$= D, F, (A⊕D), $\overline{(A\oplus F)}, \overline{(B\oplus D)}, (B\oplus F), (A\oplus B\oplus D), (A\oplus B\oplus F)$	34 ± 1.9, 36.1±3.3, 37.2±3.4, 61.3±3.7, 61.8±4.3, 37.7±2.6, 37.6±5.8, 35.5±3.2
K_2	GA-K$_2$= D, F, $\overline{(A\oplus D)}$, (A⊕F), (B⊕D), $(\overline{B\oplus F})$, $(A\oplus B\oplus D), (A\oplus B\oplus F)$	35.1±3.8, 35.6±3.1, 61.6±2.2, 37.7±4.6, 36.9±4.2, 60.8±4.5, 36.8±2.4, 36.3±3.03
ID	GA-ID= $\overline{(E\oplus F)}$, $(A\oplus B\oplus E)$, $(A\oplus D\oplus E)$, $(A\oplus E\oplus F)$,$(B\oplus D\oplus E)$, $(D\oplus E\oplus F)$,$\overline{(A\oplus B\oplus D\oplus E)}$, $(A\oplus D\oplus E\oplus F)$, $(\overline{B\oplus D\oplus E\oplus F})$	67.7±5.4, 24.5±3.6, 35.8±4.9, 22.2±1.7, 34±3.7, 31.1±3.5, 61.1±4.3, 35.8±6.14, 62.4±2.7

Fig.7 Good Approximations (Tango attack)

b) Combinations of good approximations:

To understand the combination of GA concept lets have an example for 8 bits (just to understand concept as in practical n=96bits). Suppose the following variables:

ID= [0,0,0,0,0,0,1,1]

Session i	GA (Good Approximations)	Results
A=[1,0,0,1,0,1,0,1]	$\overline{(E\oplus F)}$	0,1,0,0,0,1,1,1
B=[1,1,0,1,0,1,1,1]	$A\oplus B\oplus E$	0,0,1,1,1,1,1,1
D=[1,0,1,0,1,0,1,1]	$A\oplus D\oplus E$	0,1,0,0,1,0,1,1
E=[0,1,1,1,0,1,0,1]	$A\oplus E\oplus F$	0,0,1,0,1,1,0,1
F=[1,1,0,0,1,1,0,1]	$B\oplus D\oplus E$	0,0,0,0,0,0,0,1
	$D\oplus E\oplus F$	0,0,0,1,0,0,1,1
	$\overline{A\oplus B\oplus D\oplus E}$	0,1,1,0,1,0,1,1
	$A\oplus D\oplus E\oplus F$	1,0,0,0,0,1,1,0
	$\overline{B\oplus D\oplus E\oplus F}$	1,1,0,0,1,1,0,0
Session i+1		
A=[1,1,1,0,1,1,0,0]	$\overline{(E\oplus F)}$	1,0,0,1,0,0,1,0
B=[0,0,1,1,1,1,0,1]	$A\oplus B\oplus E$	0,0,1,0,0,1,1,0
D=[1,0,0,0,1,0,0,1]	$A\oplus D\oplus E$	1,0,0,1,0,0,1,0
E=[1,1,1,1,0,1,1,1]	$A\oplus E\oplus F$	1,0,0,0,0,0,0,1
F=[1,0,0,1,1,0,1,0]	$B\oplus D\oplus E$	0,1,0,0,0,0,1,1
	$D\oplus E\oplus F$	1,1,1,0,0,1,0,0
	$\overline{A\oplus B\oplus D\oplus E}$	0,1,0,1,0,0,0,0
	$A\oplus D\oplus E\oplus F$	0,0,0,0,1,0,0,0
	$\overline{B\oplus D\oplus E\oplus F}$	1,1,0,1,1,0,0,1

No of one's in both sessions [7,8,5,5,7,7,10,10]

Fig.8 Tango attack

Threshold value, $\gamma = \left(\frac{1}{2}\right) * N_A * N_S$

Where, N_A=Number of approximations & N_S=Number of sessions

Here in our example; N_A=9 & N_S=2

Now, if we compare the resultant number of no's with threshold, γ we can calculate the actual ID=[0,0,0,0,0,0,1,1]

So, Passive tango attack requires only a few sessions to calculate the secret ID and also it can be applied to calculate secret Keys or other important concealed values.

Same attacks are also possible for RAPP and GOASSMER, but to make this paper concise we have tested four UMAP protocols against Security model.

Performance Analysis of UMAP Protocols:

As stated above, all protocols of UMAP family have been the intention of numerous attacks. And a simple passive attack can retrieve the concealed variables (ID, Keys and random numbers) in a few eavesdropped sessions. Desynchronization attacks have some variations according to protocols but these are applicable to almost all protocols. Finally, we have shown a table, which summarize all the discussed protocols requirements (Memory requirements etc.) and security model satisfaction.

Protocol	Memory size on Tag	Total Messages for Mutual authentications	Operations	Security model satisfaction (Layer wise)			
				1	2	3	4
LMAP	6L*	4L	XOR, AND, OR, modulo-2 addition	Fail	Fail	Fail	Fail
EMAP	6L	5L	XOR, AND, OR	Fail	Fail	Fail	Fail
SASI	7L	4L	XOR, AND, OR, modulo-2 addition, Rot	Fail	Fail	Fail	Fail
GOASSMER	7L	4L	XOR, modulo-2 addition, Rot, MixBits	Fail	Pass	Fail	Pass
David-Prasad	6L	6L	XOR, AND, modulo-2 addition	Fail	Fail	Fail	Fail
RAPP	5L	6L	XOR, Per, Rot	Fail	Fail	Fail	Pass

Table.1 Performance Analysis of UMAP protocols

As we can see from above table that, none of the protocols satisfy all layers of proposed security model. But if we opt any protocol which doesn't successfully pass all security layers then our RFID system's communication will be on risk.

Conclusion:

In this paper, we presented the state of the art in the field of ultralightweight mutual authentication protocols for passive RFID tags. This paper first describes the need for ultralightweight cryptography for ubiquitous systems, and then presents some notorious ultralightweight mutual authentication protocols in sequential fashion. A security model has also been proposed to perform cryptanalysis on discussed protocols to endorse their practical feasibility. To the best of our knowledge, none of the protocols completely satisfy all four layers of proposed security model, because of inherited weak diffusion properties of T functions. These T functions have been extensively used in all UMAP protocols because of cost constraint. So, it may be quite treacherous using only simple bitwise operations to attain RFID authentication under influential adversarial model. The security of such protocols must be proved with care of cryptanalysis. Designing of a secure ultralightweight protocol without strong cryptographic algorithms is still an open problem.

References:

[1] Peris-Lopez, Pedro, et al. "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags." Proc. of 2nd Workshop on RFID Security. 2006.

[2] Peris-Lopez, Pedro, et al. "EMAP: An efficient mutual-authentication protocol for low-cost RFID tags." On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. Springer Berlin Heidelberg, 2006.

[3] Chien, Hung-Yu. "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity." IEEE Transactions on Dependable and Secure Computing, 4.4 (2007): 337-340.

[4] Umar Mujahid, M.Najam-ul-islam, Jameel Ahmed, Usman Mujahid," Cryptanalysis of ultralightweight RFID authentication protocol", IACR Cryptology ePrint Archive 2013 (2013): 385.

[5] Peris-Lopez, Pedro, et al. "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol." Information Security Applications. Springer Berlin Heidelberg, 2009. 56-68.

[6] Avoine, Gildas, Xavier Carpent, and Benjamin Martin. "Privacy-friendly synchronized ultralightweight authentication protocols in the storm." Journal of Network and Computer Applications 35.2 (2012): 826-843.

[7] Tian, Yun, Gongliang Chen, and Jianhua Li. "A new ultralightweight RFID authentication protocol with permutation." , IEEE ,Communications Letters 16.5 (2012): 702-705.

[8] Hernandez-Castro, Julio Cesar, et al. "Cryptanalysis of the David-Prasad RFID ultralightweight authentication protocol." Radio Frequency Identification: Security and Privacy Issues. Springer Berlin Heidelberg, 2010. 22-34.

[9] Li, Ticyan, and Guilin Wang. "Security analysis of two ultra-lightweight RFID authentication protocols." New Approaches for Security, Privacy and Trust in Complex Environments. Springer US, 2007. 109-120.

[10] Phan, RC-W. "Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI." IEEE Transactions on Dependable and Secure Computing, 6.4 (2009): 316-320.

[11] Cao, Tianjie, Elisa Bertino, and Hong Lei. "Security analysis of the SASI protocol." IEEE Transactions on Dependable and Secure Computing, 6.1 (2009): 73-77.

[12] Sun, Hung-Min, Wei-Chih Ting, and King-Hang Wang. "On the security of Chien's ultralightweight RFID authentication protocol." IEEE Transactions on Dependable and Secure Computing, 8.2 (2011): 315-317.

[13] Li, Tieyan, and Robert Deng. "Vulnerability analysis of EMAP-an efficient RFID mutual authentication protocol." The Second International Conference on Availability, Reliability and Security, 2007. ARES 2007.

[14] Ahmadian, Zahra, Mahmoud Salmasizadeh, and Mohammad Reza Aref. "Desynchronization attack on RAPP ultralightweight authentication protocol." Information processing letters 113.7 (2013): 205-209.

[15] Shao-hui, Wang, et al. Security analysis of RAPP an RFID authentication protocol based on permutation. Cryptology ePrint Archive, Report 2012/327, 2012.

[16] Bagheri, Nasour, et al. "Cryptanalysis of RAPP, an RFID Authentication Protocol." IACR Cryptology ePrint Archive 2012 (2012): 702.

[17] GS1 EPCglobal tag data standards version 1.4, http//www.epcglobalinc.org/standards/.