Ultralightweight Cryptography for Passive RFID Systems

Umar Mujahid[1] and M. Najam-ul-Islam[2]

[1]Department of Electrical Engineering, Bahria University Islamabad, Pakistan
[2]High Speed Digital Design Research Group, Bahria University Islamabad, Pakistan
umarmujahid.bahria@gmail.com, najam@bahria.edu.pk

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 3000). 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. Finally, Cryptanalysis of these protocols has also been performed under the implications of the proposed security model.

Keywords: RFID, Synchronization, ultralightweight, mutual authentication protocols, triangular functions, passive tags

1. Introduction

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) as mentioned in [3].

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 [1]. 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 2, we present the UMAP protocols, and then in section 3 attributes of the proposed security model have been discussed. In section 4, cryptanalysis on the basis of security model has been introduced. Performance analysis of UMAP protocols has been presented in section 5. Finally conclusion has been discussed in section 6.

2. Ultralightweight mutual authentication protocols

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

Chein [1] presented classifications of authentication protocols based on cryptographic functions that can be used at Tag’s end.

i) 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.

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

iii) 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.

iv) 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 [3] (2006), EMAP [5] (2006), SASI [1] (2007), GOSSAMER [7] (2009), David-Prasad [11] (2009) and RAPP [13] (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 [25]. 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.

2.1 LMAP

Lightweight Mutual Authentication protocol (LMAP) [3] 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 K1, K2, K3, K4 are variables that will be updated in a well synchronized manner after each successful authentication protocol run. The basic working of the protocol has been presented in fig.1. In LMAP reader initiates the protocol by transmitting the 'Hello' message towards tag, and tag responds 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 n1 and n2 and conceals these numbers within the messages A, B and C in following manner.

$$A = IDS \oplus K_1 \oplus n_1 \qquad (1)$$

$$B = (IDS \lor K_2) + n_1 \qquad (2)$$

$$C = (IDS + K_3) + n_2 \qquad (3)$$

Reader concatenates and transmits these combinational messages towards tag. The tag will then retrieve the random numbers n1and n2 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, K1, K2, K3, K4). Now, tag computes and transmits D message towards reader.

$$D = (IDS + ID) \oplus n_1 \oplus n_2 \qquad (4)$$

From D messages reader authenticates tag and then after successful tag authentication reader will also update its Pseudonyms (IDS, K1, K2, K3, K4). 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.

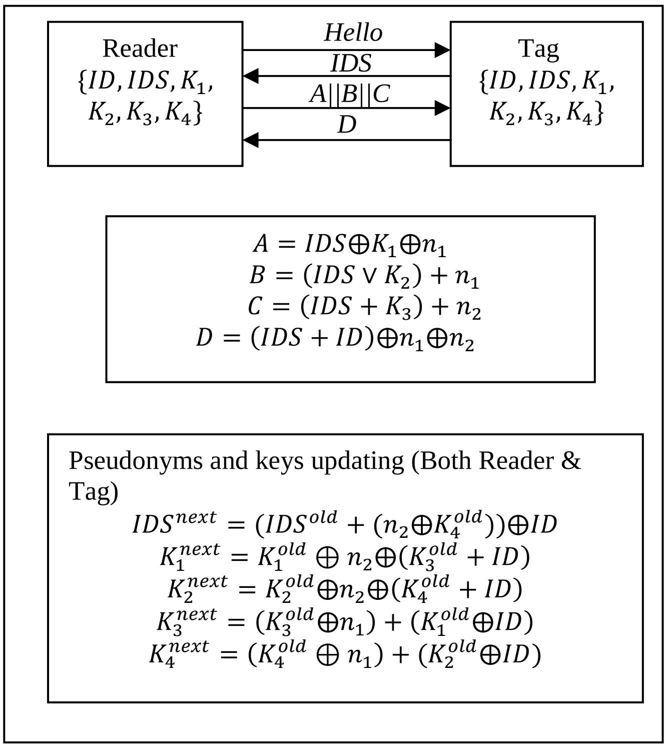

Reader
$\{ID, IDS, K_1, K_2, K_3, K_4\}$
Hello
IDS
$A||B||C$
D
Tag
$\{ID, IDS, K_1, K_2, K_3, K_4\}$
$A = IDS \oplus K_1 \oplus n_1$
$B = (IDS \lor K_2) + n_1$
$C = (IDS + K_3) + n_2$
$D = (IDS + ID) \oplus n_1 \oplus n_2$
Pseudonyms and keys updating (Both Reader & Tag)
$IDS^{next} = (IDS^{old} + (n_2 \oplus K_4^{old})) \oplus ID$
$K_1^{next} = K_1^{old} \oplus n_2 \oplus (K_3^{old} + ID)$
$K_2^{next} = K_2^{old} \oplus n_2 \oplus (K_4^{old} + ID)$
$K_3^{next} = (K_3^{old} \oplus n_1) + (K_1^{old} \oplus ID)$
$K_4^{next} = (K_4^{old} \oplus n_1) + (K_2^{old} \oplus ID)$

Figure 1. LMAP Protocol

2.2 EMAP

Efficient mutual authentication protocol (EMAP) [5] was another protocol from UMAP family. Here a new Parity function 'Fp' was added, 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 n1 & n2 and conceals these random numbers within messages A, B and C in the following manner.

$$A = IDS \oplus K_1 \oplus n_1 \qquad (5)$$

$$B = (IDS \lor K_2) \oplus n_1 \qquad (6)$$

$$C = (IDS \oplus K_3 \oplus n_2 \qquad (7)$$

Reader transmits these messages towards tag and tag retrieves concealed 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.

$$D = (IDS \land K_4) \oplus n_2 \qquad (8)$$

$$E = (IDS \land n_1 \lor n_2) \oplus ID \oplus K_1 \oplus K_2 \oplus K_3 \oplus K_4 \qquad (9)$$

After receiving D and E messages, 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. Working of EMAP has been presented in figure.2. 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.

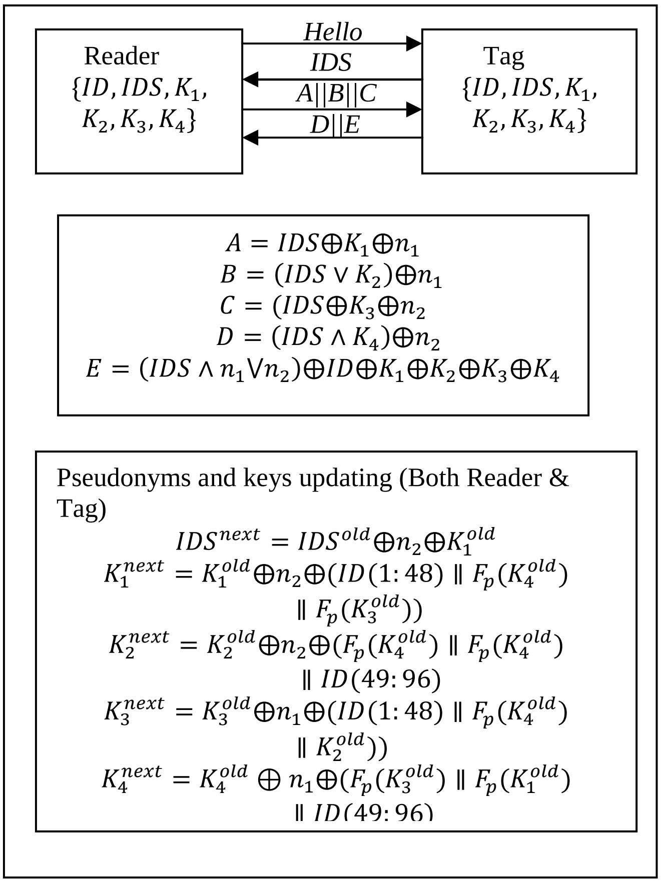

Reader
$\{ID, IDS, K_1, K_2, K_3, K_4\}$
Hello
IDS
$A||B||C$
$D||E$
Tag
$\{ID, IDS, K_1, K_2, K_3, K_4\}$
$A = IDS \oplus K_1 \oplus n_1$
$B = (IDS \vee K_2) \oplus n_1$
$C = (IDS \oplus K_3 \oplus n_2$
$D = (IDS \wedge K_4) \oplus n_2$
$E = (IDS \wedge n_1 \vee n_2) \oplus ID \oplus K_1 \oplus K_2 \oplus K_3 \oplus K_4$
Pseudonyms and keys updating (Both Reader & Tag)
$IDS^{next} = IDS^{old} \oplus n_2 \oplus K_1^{old}$
$K_1^{next} = K_1^{old} \oplus n_2 \oplus (ID(1:48) \parallel F_p(K_4^{old}) \parallel F_p(K_3^{old}))$
$K_2^{next} = K_2^{old} \oplus n_2 \oplus (F_p(K_4^{old}) \parallel F_p(K_4^{old}) \parallel ID(49:96)$
$K_3^{next} = K_3^{old} \oplus n_1 \oplus (ID(1:48) \parallel F_p(K_4^{old}) \parallel K_2^{old}))$
$K_4^{next} = K_4^{old} \oplus n_1 \oplus (F_p(K_3^{old}) \parallel F_p(K_1^{old}) \parallel ID(49:96)$

Figure 2. EMAP Protocol

2.3 SASI

In 2007, Chein presented a new ultralightweight mutual authentication protocol SASI [1] (Strong authentication and integrity). 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 presented in figure 3.

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 than 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:

$$A = IDS \oplus K_1 \oplus n_1 \qquad (10)$$

$$B = (IDS \vee K_2) + n_2 \qquad (11)$$

$$C = (K_1 \oplus \overline{K_2}) + (\overline{K_1} \oplus K_2) \qquad (12)$$

Where,

$$\overline{K_1} = Rot(K_1 \oplus n_2, K_1) \qquad (13)$$

$$\overline{K_2} = Rot(K_2 \oplus n_1, K_2) \qquad (14)$$

On receiving of A||B||C, tag extracts n1 from A and n2 from B. Tag uses equations (13) and (14) to compute $\overline{K_1}$ & $\overline{K_2}$ which then be used in equation (12) to calculate the local value of C. Tag compares the locally calculated C 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.

$$D = (\overline{K_2} + ID) \oplus \left((K_1 \oplus K_2) \vee \overline{K_1}\right) \qquad (15)$$

On receiving of ‘D’ reader will verify the received D and updates its pseudonyms. Again here update process is similar except backups 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.

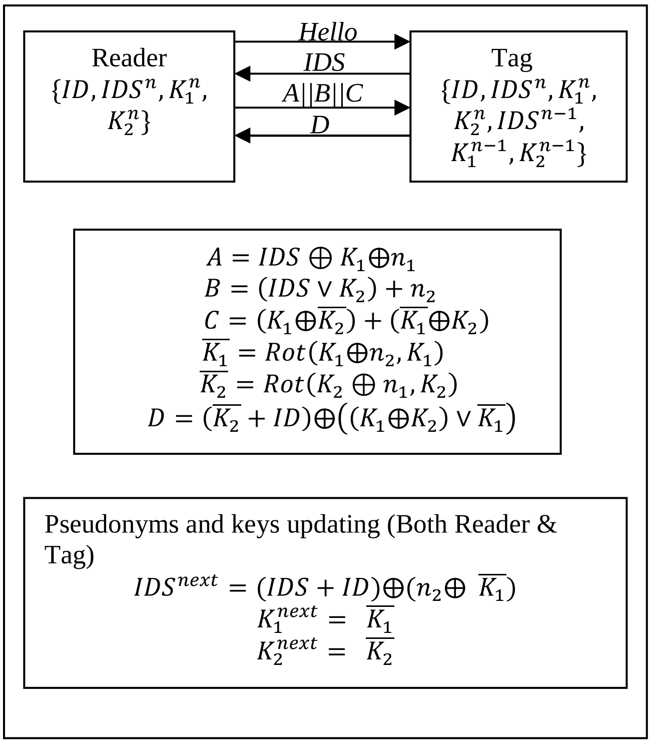

Reader
$\{ID, IDS^n, K_1^n, K_2^n\}$
Hello
IDS
$A||B||C$
D
Tag
$\{ID, IDS^n, K_1^n, K_2^n, IDS^{n-1}, K_1^{n-1}, K_2^{n-1}\}$
$A = IDS \oplus K_1 \oplus n_1$
$B = (IDS \vee K_2) + n_2$
$C = (K_1 \oplus \overline{K_2}) + (\overline{K_1} \oplus K_2)$
$\overline{K_1} = Rot(K_1 \oplus n_2, K_1)$
$\overline{K_2} = Rot(K_2 \oplus n_1, K_2)$
$D = (\overline{K_2} + ID) \oplus \left((K_1 \oplus K_2) \vee \overline{K_1}\right)$
Pseudonyms and keys updating (Both Reader & Tag)
$IDS^{next} = (IDS + ID) \oplus (n_2 \oplus \overline{K_1})$
$K_1^{next} = \overline{K_1}$
$K_2^{next} = \overline{K_2}$

Figure 3. SASI Protocol

2.4 Gossamer Protocol

In 2009, Peris-Lopez et.al proposed a new ultralightweight mutual authentication protocol: GOSSAMER [7]. 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 algorithm, working of the MixBits function can be seen. 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 Mix Bits composite string.

Basic working of Gossamer protocol has been shown in figure.4. The protocol works in the similar fashion as we have already discussed in the other protocols. Reader initiates the protocol by transmitting ‘Hello’ message towards tag. Tag responds with its current updated IDS. On receiving of IDS, reader looks for a matching entry in its database; if a match occurs then it further sends the concatenated message A||B||C, which are defined as:

$$A = Rot(Rot(IDS + K_1 + \pi + n_1, K_2) + K_1, K_1) \quad (16)$$

$$B = Rot(Rot(IDS + K_2 + \pi + n_2, K_1) + K_2, K_2) \quad (17)$$

$$C = Rot(Rot(n_3 + K_1^* + \pi + n_1', n_3) + K_2^* \oplus n_1', n_2) \oplus n_1' \quad (18)$$

Where,

$$n_1' = MixBits(n_3, n_2) \quad (19)$$

$$K_1^* = Rot(Rot(n_2 + K_1 + \pi + n_3, n_2) + K_2 \oplus n_3, n_1) \oplus n_3 \quad (20)$$

$$K_2^* = Rot(Rot(n_1 + K_2 + \pi + n_3, n_1) + K_1 + n_3, n_2) + n_3 \quad (21)$$

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 n1’, n3 & K1’ 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.

$$D = Rot(Rot(n_2 + K_2^* + ID + n_1', n_2) + K_1^* + n_1', n_3) + n_1' \quad (22)$$

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

This protocol is more sophisticated than other protocols of UMAP family, as there is no full disclosure attack available, which can break Gossamer. But in [9], Zeeshan, et.al. found Desynchronization attack against the Gossamer protocol.

2.5 David-Prasad protocol

In September 2009, David and Prasad [11] proposed a new ultralightweight mutual authentication protocol for passive low cost RFID tags. The protocol was also inspired from the its contending UMAP family protocols. 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; readers 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 (n1, n2), computes and then transmits messages A, B and D, which are as follows:

$$A = IDS \wedge K_1 \wedge K_2) \oplus n_1 \quad (23)$$

$$B = (\overline{IDS} \wedge K_2 \wedge K_1) \oplus n_2 \quad (24)$$

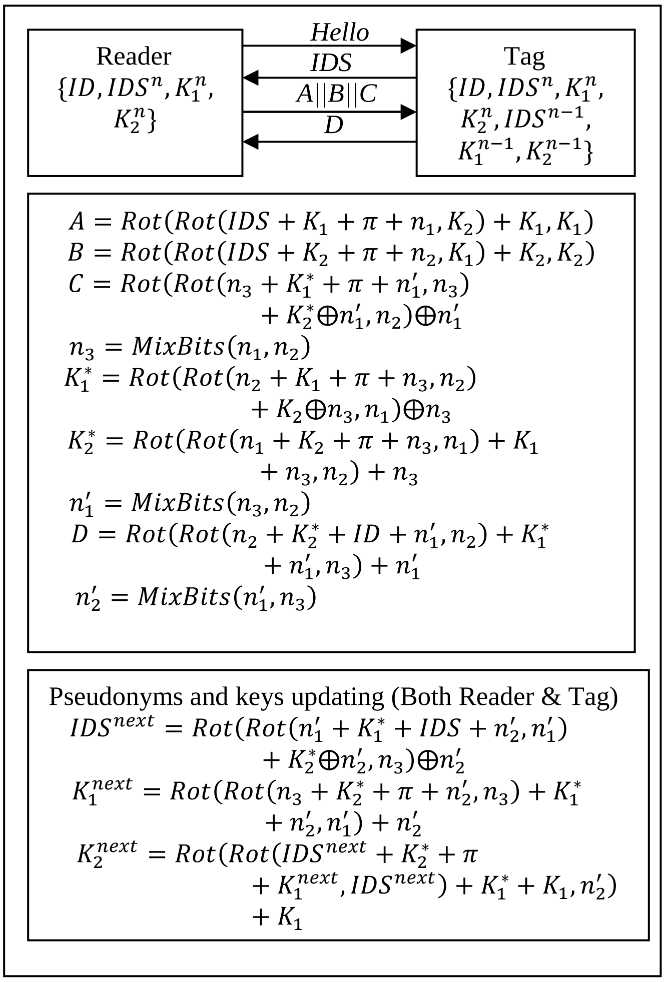
$$D = (K_1 \wedge n_2) \oplus (K_2 \wedge n_1) \quad (25)$$

Tag then extracts nonces (n1, n2) 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.

$$E = K_1 \oplus n_1 \oplus ID \oplus (K_2 \wedge n_2) \quad (26)$$

$$F = (K_1 \wedge n_1) \oplus (K_2 \wedge n_2) \quad (27)$$

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. Working of the protocol is shown in figure.5.


Hello
IDS
A||B||C
D
Reader
$\{ID, IDS^n, K_1^n, K_2^n\}$
Tag
$\{ID, IDS^n, K_1^n, K_2^n, IDS^{n-1}, K_1^{n-1}, K_2^{n-1}\}$

$$A = Rot(Rot(IDS + K_1 + \pi + n_1, K_2) + K_1, K_1)$$
$$B = Rot(Rot(IDS + K_2 + \pi + n_2, K_1) + K_2, K_2)$$
$$C = Rot(Rot(n_3 + K_1^* + \pi + n_1', n_3) + K_2^* \oplus n_1', n_2) \oplus n_1'$$
$$n_3 = MixBits(n_1, n_2)$$
$$K_1^* = Rot(Rot(n_2 + K_1 + \pi + n_3, n_2) + K_2 \oplus n_3, n_1) \oplus n_3$$
$$K_2^* = Rot(Rot(n_1 + K_2 + \pi + n_3, n_1) + K_1 + n_3, n_2) + n_3$$
$$n_1' = MixBits(n_3, n_2)$$
$$D = Rot(Rot(n_2 + K_2^* + ID + n_1', n_2) + K_1^* + n_1', n_3) + n_1'$$
$$n_2' = MixBits(n_1', n_3)$$

Pseudonyms and keys updating (Both Reader & Tag)

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

Figure 4. GOSSAMER Protocol

2.6 RAPP protocol

In 2012, Yun Tian et.al proposed a new ultralightweight RFID mutual authentication protocol with permutation (RAPP) [13] . 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, K1, K2, & K3 (each is of 96-bit long). To avoid Desynchronization attacks in addition to current pseudonyms reader 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 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 n1, otherwise vice versa. After calculating A and B, reader then transmits these messages toward tag. Where A and B messages are as follows:

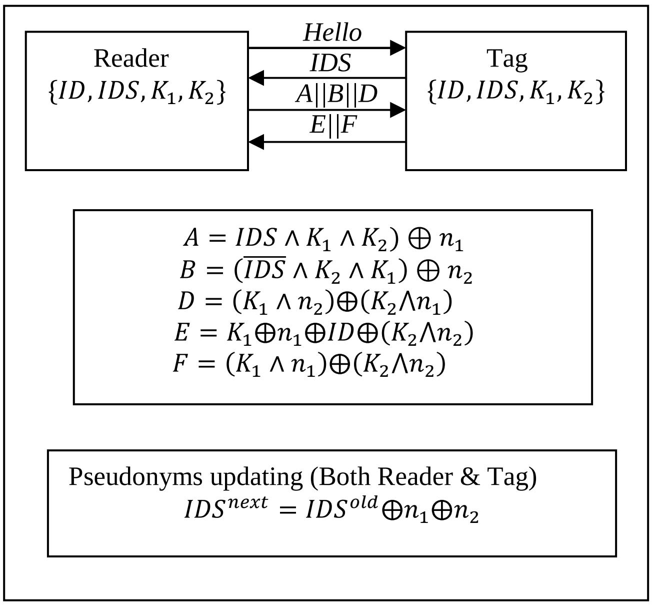

Reader
$\{ID, IDS, K_1, K_2\}$
Hello
IDS
$A||B||D$
$E||F$
Tag
$\{ID, IDS, K_1, K_2\}$
$A = IDS \wedge K_1 \wedge K_2) \oplus n_1$
$B = (\overline{IDS} \wedge K_2 \wedge K_1) \oplus n_2$
$D = (K_1 \wedge n_2) \oplus (K_2 \wedge n_1)$
$E = K_1 \oplus n_1 \oplus ID \oplus (K_2 \wedge n_2)$
$F = (K_1 \wedge n_1) \oplus (K_2 \wedge n_2)$
Pseudonyms updating (Both Reader & Tag)
$IDS^{next} = IDS^{old} \oplus n_1 \oplus n_2$

Figure 5. David-Prasad Protocol

$$A = Per(K_2, K_1) \oplus n_1 \quad (28)$$
$$B = Per\big(K_1 \oplus K_2, Rot(n_1, n_1)\big) \oplus Per(n_1, K_1) \quad (29)$$

Tag extracts n1 message from A and calculate the local version of B. If local value of B and received value of B is same then tag transmits C message towards reader.

$$C = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID \quad (30)$$

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 n2. Both n1 and n2 will be used for key update. Reader calculates D and E messages and transmits them towards tag. Then reader also updates its pseudonyms for future correspondence with the particular tag.

$$D = Per(K_3, K_2) \oplus n_2 \quad (31)$$
$$E = Per\big(K_3, Rot(n_2, n_2)\big) \oplus Per(n_1, K_3 \oplus K_2) \quad (32)$$

Tag extracts n2 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 basic working of the protocol is shown in figure 6.

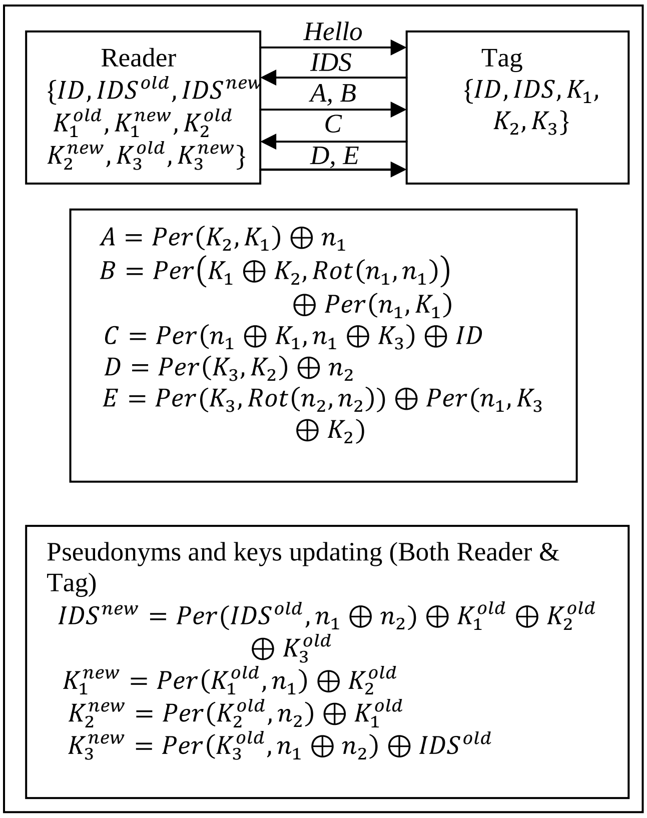

Reader
$\{ID, IDS^{old}, IDS^{new}$
$K_1^{old}, K_1^{new}, K_2^{old}$
$K_2^{new}, K_3^{old}, K_3^{new}\}$
Hello
IDS
A, B
C
D, E
Tag
$\{ID, IDS, K_1,$
$K_2, K_3\}$
$A = Per(K_2, K_1) \oplus n_1$
$B = Per\big(K_1 \oplus K_2, Rot(n_1, n_1)\big) \oplus Per(n_1, K_1)$
$C = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$
$D = Per(K_3, K_2) \oplus n_2$
$E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$
Pseudonyms and keys updating (Both Reader & Tag)
$IDS^{new} = Per(IDS^{old}, n_1 \oplus n_2) \oplus K_1^{old} \oplus K_2^{old} \oplus K_3^{old}$
$K_1^{new} = Per(K_1^{old}, n_1) \oplus K_2^{old}$
$K_2^{new} = Per(K_2^{old}, n_2) \oplus K_1^{old}$
$K_3^{new} = Per(K_3^{old}, n_1 \oplus n_2) \oplus IDS^{old}$

Figure 6. RAPP Protocol

3. Security model for cryptanalysis of ultralightweight authentication protocols

The security of the protocols 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 integrity, tag anonymity and untraceibility. While Desynchronization, full disclosure, cloning and replay etc. fall in the attack category.

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

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

c) 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.

d) 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 [2, 4, 6, 8-10, 12, 14-18, 20, 22, 25, 28] 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 in Table.1.

Table 1. Security Model for UMAP

S.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	(i) Eavesdropping
4	General Adhoc/Probabilistic 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.

i) Desynchronization**:** In this layer, the adversaries 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.

ii) 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.

iii) 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.

iv) General adhoc/Probabilistic 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.

4. Security analysis of UMAP protocols

Manjulata and Adarsh Kumar [27] described the detailed security analysis of lightweight protocols and primitives. In this section, we will perform security analysis of the various ultralightweight 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 of security model then it will lose its candidacy for being a Standard UMAP protocol.

4.1 Security analysis of LMAP & EMAP

4.1.1 Desynchronization attack

Desynchronization attack is easily applicable on 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.

$$A = IDS \oplus K_1 \oplus n_1 \quad (33)$$

$$B = (IDS \vee K_2) + n_1 \quad (34)$$

$$C = (IDS + K_3) + n_2 \quad (35)$$

As these messages are from a valid reader, so tag generates a message D using n1 and n2. But now attacker interrupts the link and block D.

$$D = (IDS + ID) \oplus n_1 \oplus n_2 \quad (36)$$

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.

4.2 Security analysis of SASI protocol

4.2.1 Desynchronization attack

Sun, Hung-Min and Wei-Chih Ting performed Desynchronization attack on SASI in [2]. 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 an 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 IDS3, attacker sends an error signal towards tag and asks for IDS_1. Tag immediately responds with IDS1 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, $K1_3$, $K2_3$).

4.2.2 Adhoc/ Probabilistic Attacks

To understand adhoc/probabilistic attacks we take a scenario. For example, if a 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 IDS2 from tag, attacker asks for IDS1 (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 n1. So, it will transmit D message towards attacker and updates its Pseudonyms (IDS_3, $K1_3$, $K2_3$). Now, next time if a genuine reader wants to communicate with this meticulous tag; it will not find its entry in the database.

4.3 Security Analysis of David-Prasad Protocol

4.3.1 Desynchronization attack

Desynchronization attack [25] 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, let's have a look this cryptanalysis for David-Prasad.

4.3.2 Adhoc/Probabilistic attacks

As, we know that XOR & AND operations give unalike results with 75% probability ratio. We can see this thing from the truth table 2.

Table 2. XOR & AND 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 equation 26 and 27 (David-Prasad messages from Tag) we can find the ID (Secret) of tag with 75% probability of correctness. The operation is as follows:

$$E \oplus F = [K_1 \oplus n_1 \oplus ID \oplus (K_2 \wedge n_2)] \oplus (K_1 \wedge n_1) \oplus (K_2 \wedge n_2) \qquad (37)$$

$$= (K_1 \oplus n_1) \oplus ID (K_1 \wedge n_1) \qquad (38)$$

$$= ID$$

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.

4.3.3 Full disclosure attack (Tango attack)

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 compares the number of one's for two consecutive sessions with a threshold value. Here we have mentioned in Table.3, some GA for each of the three secret values on the basis of hamming distances (10000 tests).

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] and considerTable.4 for retrieval of the concealed secret using tango attack.

Table. 3. Good Approximations (GA)

Target	Good Approximations (GA)	Hamming Distance
K_1	GA-K$_1$=$D, F, (A \oplus D), (\overline{A \oplus F})$ $(\overline{B \oplus D}), (B \oplus F), (A \oplus B \oplus D),$ $(A \oplus B \oplus F)$	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$_2$=$D, F, \overline{(A \oplus D)}, (A \oplus F),$ $(B \oplus D), (\overline{B \oplus F}), (A \oplus B \oplus D),$ $(A \oplus B \oplus F)$	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=$(\overline{E \oplus F}), (A \oplus B \oplus E)$ $(A \oplus D \oplus E), (A \oplus E \oplus F)$ $(B \oplus D \oplus F), (D \oplus E \oplus F),$ $(\overline{A \oplus B \oplus D \oplus E}),$ $(A \oplus D \oplus E \oplus F), (\overline{B, D, E. F})$	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, 3 5.8±6.14, 62.4±2.7

Threshold value, $\gamma = \left(\frac{1}{2}\right) * N_A * N_s$ (39)

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 our proposed Security model.

5. 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 performance analysis of protocol under the implication of the proposed security model in

Table.4. Tango attack on David-Prasad

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]	$(B \oplus D \oplus F)$	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
	$(\overline{B, D, E. F})$	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]	$(B \oplus D \oplus F)$	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
	$(\overline{B, D, E. F})$	1,1,0,1,1,0,0,1

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

table.5, which summarize all the discussed protocols requirements (Memory requirements etc.) and their satisfaction success against the model. In LMAP protocol, each tag owns one static ID, one dynamic identity pseudonym (IDS) and four keys (K_1,K_2,K_3,K_4). Each entry in tag is of L bits long (96 bits), hence overall 6L (576 bits) memory is required on tag for LMAP protocol.. Protocol incorporates only simple bitwise operations such as XOR, AND, OR and modulo-2 addition. Because of poor protocol design methodology and extensive use of T functions, LMAP does not survive against even the simplest desynchronization attacks EMAP also requires the same memory (6L) on tag for storage of permanent ID, IDS and four keys (K_1,K_2,K_3,K_4). Authors of EMAP improved some design methodology of the protocol but due to extensive use of T functions, protocol cannot defend the simple cryptanalysis attack. SASI and GOSSAMER require 7L (672 bits) of memory on tag, as in addition to current pseudonyms and keys both protocols also store the old pseudonyms and keys to combat against desynchronization attacks. But as we have shown in section 4, both protocols are vulnerable to desynchronization attacks. David-Prasad protocol requires 6L of memory on tag. Protocol uses simple T-functions to generate its messages but this simplicity leads the protocol towards failure against simple cryptanalysis attacks. RAPP protocol requires the least memory requirement on tag; 5L. But RAPP is also not able to pass all four layers (tests) of the security model. As we can see from Table.5. that, none of the protocols satisfy all layers of proposed security model. And, if we select any protocol; which doesn't successfully pass all security layers then our RFID system's communication will be on risk.

6. 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] Hung-Yu Chien," SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity" IEEE Transaction on Dependable and Secure Computing, Vol. 4, No. 4, pp. 337 – 340, 2007

[2] Hung-Min Sun, Wei-Chih Ting et.al," On the Security of Chien's Ultralightweight RFID Authentication Protocol" IEEE Transaction on Dependable and Secure Computing, Vol. 8, No. 2, pp. 315 – 317, 2011

[3] Pedro Peris-Lopez, Julio Hernandez-Castro et.al. "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags."Proceedings of 2nd Workshop on RFID Security, Austria, pp.100-112, 2006.

[4] Tieyan Li et.al," Security Analysis of family of Ultra-Lightweight RFID Authentication Protocols", Journal of Software, Vol. 3, No. 3, pp. 1-10, 2008.

[5] Peris-Lopez, Pedro, Julio Cesar Hernandez et.al. "EMAP: An efficient mutual-authentication protocol for low-cost RFID tags." The 1st International Workshop on Information security (OTM-2006), France, pp. 352-361, 2006.

[6] Tianjie, Elisa Bertin at.al. "Security analysis of the SASI protocol." IEEE Transactions on Dependable and Secure Computing, Vol.6, No. 1, pp. 73 – 77, 2009.

[7] Peris-Lopez, Pedro, et al. "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol." The 9th International Workshop on Information Security Applications, pp. 56-68, 2009.

[8] Yeh, Kuo-Hui, and N. W. Lo. "Improvement of two lightweight RFID authentication protocols." Information Assurance and Security Letters Vol.1, No.1, pp 6-11, 2010.

[9] Bilal, Zeeshan, Ashraf Masood, and Firdous Kausar. "Security analysis of ultra-lightweight cryptographic protocol for low-cost RFID tags: Gossamer protocol."

The 12th International Conference on Network-Based Information Systems, Indianapolis, USA, pp. 260-267, 2009.

[10] Muhammad Zubair, Umar Mujahid et.al," Cryptanalysis of RFID Ultralightweight protocols and comparison between its solution approaches", Bahria University Journal of information & communication technology, Vol.5, No. 1, pp. 58-63, 2012

[11] David, Mathieu, and Neeli R. Prasad. "Providing strong security and high privacy in low-cost RFID networks." International conference on Security and privacy in mobile information and communication systems, Italy, pp172-179, 2009.

[12] Barrero, David F.et.al. "A genetic tango attack against the David–Prasad RFID ultra-lightweight authentication protocol." Expert Systems (Journal) Vol. 31, no. 1, pp. 9-19, 2014.

[13] Tian, Yun, Gongliang Chen, and Jianhua Li. "A new ultralightweight RFID authentication protocol with permutation." IEEE Communications Letters, Vol.16, no. 5, pp.702-705, 2012.

[14] Bagheri, Nasour, Masoumeh Safkhani et al. "Cryptanalysis of RAPP, an RFID Authentication Protocol", Cryptology ePrint Archive, Report 2012/702, https://eprint.iacr.org/2012/702 , 2012.

[15] Nie, Tingyuan, and Teng Zhang. "A study of DES and Blowfish encryption algorithm." IEEE Region 10 Conference, TENCON-2009, Singapore, pp. 1-4, 2009.

[16] Engels, Daniel, et al. "Hummingbird: ultra-lightweight cryptography for resource-constrained devices." The 14th International Conference on Financial Cryptography and Data Security, Spain, pp.3-18, 2010.

[17] Song, Boyeon, and Chris J. Mitchell. "RFID authentication protocol for low-cost tags" The 1st ACM conference on Wireless network security, USA, pp. 140-147, 2008.

[18] Rizomiliotis, Panagiotise et.al"Security analysis of the Song-Mitchell authentication protocol for low-cost RFID tags." IEEE Communications Letters, Vol.13, No. 4, pp. 274-276, 2009.

[19] Pedro Peris López," Lightweight Cryptography in Radio Frequency Identification (RFID) Systems", PhD thesis, UNIVERSIDAD CARLOS III DE MADRID, 2008.

[20] Pedro Peris-Lopez, et.al "Quasi-linear cryptanalysis of a secure RFID ultralightweight authentication protocol ", The 6th International Conference on Information Security and Cryptology, China, pp. 427-442, 2011.

[21] Yeh, Kuo-Hui, N. W. Lo, and Enrico Winata. "An efficient ultralightweight authentication protocol for RFID systems."Workshop on RFID Security and Privacy, Turkey, pp 49-60, 2010.

[22] Zahra Ahmadian, Mahmoud. et.al "Recursive Linear and Differential Cryptanalysis of ultralightweight authentication protocols", IEEE Transactions on Information Forensics and Security, Vol.8. No.7, pp. 1140 – 1151, 2013.

[23] Julio C. Hernandez. et.al "Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations." ArXiv, Cryptography and Security, Report; 0811.4257, http://arxiv.org/abs/0811.4257, 2008.

[24] Avoine, Gildas, Xavier Carpent, and Benjamin Martin. "Privacy-friendly synchronized ultralightweight authentication protocols in the storm." Journal of Network and Computer Applications, Vol.35, No. 2, pp. 826-843, 2012.

[25] GS1 EPCglobal tag data standards version 1.4, Available from; http//www.epcglobalinc.org/standards/.

[26] Hernandez-Castro, Julio Cesar, et.al "Cryptanalysis of the David-Prasad RFID ultralightweight authentication protocol." Workshop on RFID Security and Privacy, Turkey, pp. 22-34, 2010.

[27] Manjulata and Adarsh Kumar." Survey on Lightweight Primitives and Protocols for RFID in Wireless Sensor Networks", International Journal of Communication Networks and Information Security, Vol. 6, No. 1, pp. 29-43, 2014

[28] Natarajan Meghanathan," A Survey on the Communication Protocols and Security in Cognitive Radio Networks", International Journal of Communication Networks and Information Security, Vol. 5, No. 1, pp.19-38, 2013.

Table.5. Performance Analysis of UMAP

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