Pitfalls in Ultralightweight RFID Authentication Protocol

Umar Mujahid and Muhammad Najam-ul-islam

Department of Electrical Engineering, Bahria University Islamabad, Pakistan
umar.mujahid@bui.edu.pk, najam@ bui.edu.pk

Abstract: Radio frequency identification (RFID) is one of the most promising identification schemes in the field of pervasive systems. Non-line of sight capability makes RFID systems more protuberant than its contended systems. Since the RFID systems incorporate wireless medium, so there are some allied security threats and apprehensions from malicious adversaries. In order to make the system reliable and secure, numerous researchers have proposed ultralightweight mutual authentication protocols; which involve only simple bitwise logical operations (AND, XOR & OR etc.) to provide security. In this paper, we have analyzed the security vulnerabilities of state of the art ultralightweight RFID authentication protocol: RAPP. We have proposed three attacks (two DoS and one Desynchronization) in RAPP protocol and challenged its security claims. Moreover, we have also highlighted some common pitfalls in ultralightweight authentication protocol designs. This will help as a sanity check, improve and longevity of ultralightweight authentication protocol designs.

Keywords: Ultralightweight, RFID, RAPP, Synchronization, Denial of Service., cryptography.

1. Introduction

Currently, barcodes and RFID systems are the two widely used identification schemes. The efficient functional haste and prevailing features (Automation and Non-line of sight) of RFID systems cause its massive deployment than other contended schemes. Moreover, RFID systems can uniquely identify each item/ product (tag), while mostly barcodes can only identify the type of the item/product (not unique identification). The only hindrance in rapid growth of RFID technology is security concerns and overall cost of the tag, which should be 0.05 to 0.1 $ to be considered comparable with the barcodes [18]. The demand of low cost tags limits us to use passive RFID tags which involve simple computational operations for security and other functions. Typically, such tags can store only $32-1K$ bits and can support $250-4K$ logic gates for security related tasks. So, conventional cryptographic algorithms (such as AES, Triple DES etc.) and primitives (such as Hash function etc.) cannot be used to secure the system.

RFID systems mainly comprise of Radio Frequency (RF) tags or electronic chips, RF reader or transceiver stations and backend database. The RFID tag contains the secret information (Identity and keys) regarding the object on to which it has been attached. Whenever a tag enters in the vicinity of reader, it will be asked for its identity (ID). After receiving ID, the reader confirms its validity from central database of tags. Generally, we assume that channel between central database and the reader is secure, as we may use the traditional cryptographic algorithms (AES, 3DES, Hashing etc.) to ensure security of this channel. However researchers have proposed various cryptographic solutions including mutual authentication protocols to secure the channel between the reader and the tag. Based on the computational capabilities at tag's side, the authentication protocols have been classified into four categories [1]: Full – fledged, Simple, Lightweight and Ultralightweight:

a) Full-fledged protocols can incorporate the traditional cryptographical algorithms and solutions, like one way hash functions, public or private key cryptography, and so forth.

b) Simple authentication protocols can support pseudorandom number generators and one-way hash functions only.

c) Lightweight protocols can support only lightweight pseudorandom number generators and simple functions such as cyclic redundancy check (CRC) but cannot use hash functions.

d) Ultralightweight protocols can incorporate only simple bitwise logical operations and even pseudorandom number generators cannot be used at the tag's side.

For secure communication of low cost RFID systems, we use ultralightweight mutual authentication protocols. Ultralightweight Mutual Authentication Protocol (UMAP) family provides extremely low security. This is mainly due to wide use of simple $T-functions$ [36] for development of security algorithms, in addition to traditional cryptographic functions (which are in fact resource hungry). However, inclusion of non-triangular operations (Rotation, Permutation, Recursive Hash, etc.) in UMAP family protocols augments the resistance against various types of security attacks.

The rest of the paper is organized as follows: Section 2 describes the related works. Section 3 presents the basic working of RAPP protocol which is followed by the proposed cryptanalysis of RAPP protocol in Section 4. Section 5 discusses the pitfalls of ultralightweight authentication protocols and suggestions to avoid common mistakes. Finally, conclusion has been presented in Section 6.

2. Related Works

In 2006, P. Peris-Lopez et al. [3 – 5] laid the foundation of ultralightweight cryptography for passive RFID systems. They highlighted that the classical cryptographic primitives such as Pseudo Random Number Generators (PRNGs), hash functions, block ciphers etc. lie well beyond the computational capabilities of the low cost resource constrained systems. So, they proposed three extremely lightweight mutual authentication protocols (named UMAP family): LMAP (Lightweight Mutual Authentication Protocol), M2AP (Minimalist Mutual Authentication Protocol) and EMAP (An Efficient Mutual Authentication Protocol) for low cost passive RFID tags. The UMAP family protocols involves only simple bitwise logical operations (such as XOR, AND, OR etc.) to keep the cost of the system as low as possible. The hardware approximation of UMAP protocols show that the LMAP requires only 300 gates while EMAP and M2AP require only 150 and 300 gates respectively. The protocols mainly composed of three steps: tag identification, mutual identification, pseudonym and key updating (for next protocol sessions). The randomness of the protocol messages is ensured with three randomness test suites: DIEHARD [37], ENT [38] and NIST [39].However, Tieyan Li et al. [29, 30] performed security analysis of UMAP family protocols. They exploited the inherent weak diffusion properties of $T - functions$ [36] and found two effective attacks on the protocols: desynchronization and full disclosure. The former permanently abolishes the authentication capability of tag, while later completely discloses all the concealed secrets stored in a tag.

In 2007, Chein [1] uses a new non-triangular primitive 'Rotation (Rot)' in protocol messages and proposed an ultralightweight RFID authentication protocol to provide Strong Authentication and Strong Integrity: SASI. Rotation (Rot) function is extremely lightweight as it requires only two registers for its operation; however it is a clock cycle consuming operation (since for each rotation 'l' clock cycles are required; where 'l' is the number of bits in both strings). Unfortunately Hung-Min Sun et al. [17] and Hernandez et al. [41] found desynchronization and full disclosure attacks in SASI protocol. Thus enlists the SASI protocol among the vulnerable authentication protocols.

Later, Yeh et al. [10], GOASSMER [6] and David-Prasad [9] protocols were also reported to be vulnerable against various desynchronization, traceability and full disclosure attacks [20, 24 and 25].

In 2012, Tian et al. [2] introduced new ultralightweight non-triangular primitive "Permutation" (Per) and proposed a new ultralightweight RFID Authentication Protocol using Permutation (RAPP). Permutation (Per) operation is highly effective and extremely lightweight in nature; however it reveals the information of hamming weight (hw) of the first parameter (operand). We will also use this inherent weakness of Per operation to highlight the Desynchronization and DoS attacks on RAPP protocol.

In 2013, Jeon and Yoon [11] proposed a new ultralightweight RFID authentication protocol named RAPLT (RFID Authentication Protocol for Low cost Tags) using non – triangular primitives (Separate and Merge operations). However Zhuang et al. [43] found desynchronization and traceability attacks in the protocol and showed that RAPLT is as vulnerable as its contended UMAPs.

Most of the previously proposed ultralightweight authentication protocols [1 – 13, 33 34] have similar flaws such as use of $T - functions$, linear functions (Rot, Per etc.) and poor messages composition etc. So, these parameters should be taken into account while designing a privacy friendly authentication protocols. Section 5 briefly describes the pitfalls in ultralightweight authentication protocol designs.

3. RAPP Scheme

RAPP involves three objects i.e. tag, reader and backend database. In RAPP, the channel between reader and backend database is assumed to be secure as stated earlier and can be connected via reliable wired connection. However on the other hand, the channel between the tag and reader is wireless and open for all possible adversary attacks. Each tag has an l-bit unique secret identifier ID, and other four elements $\{IDS$, K_1, K_2, and $K_3\}$. In RAPP, tag involves only three operations; bitwise XOR, left rotation, and permutation.

Permutation operation is defined as follows:

Consider X and Y are two $l - bit$ strings:

$$X = x_1 x_2 x_3 \dots . x_l, \quad x_i \in \{0,1\}, i = 1,2 \dots, l$$
$$Y = y_1 y_2 y_3 \dots . y_l, \quad y_i \in \{0.1\}, i = 1,2 \dots, l$$

Hamming weight of Y, $wt(Y)$is $m(0 \le m \le l)$ and $y_{k1} = y_{k2} = \cdots y_{km} = 1$; $y_{km+1} = Y_{km+2} \dots = y_{kl} = 0$

Where $1 \le k_1 < k_2 \dots < k_m \le l$ and $1 \le k_{m+1} \dots < k_l \le l$ then Permutation of X according to Y, *Per(X,Y)* will be

$$Per(X, Y) = x_{k1}, x_{k2} \dots x_{km} x_{kl} x_{kl-1} \dots x_{km+1}$$

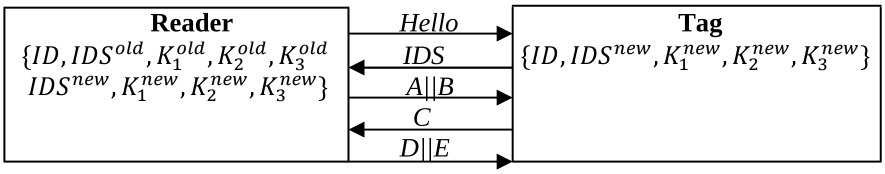
For example; $X = 110100$ & $Y = 011110$

$$Per\ (X, Y)\ = 101001$$

The permutation can be computed by considering the two pointers P_1 and P_2 as index values for their corresponding strings: X and Y. As in our example as $y_1 = 0$ so, x_1 bit will be moved to last position in the third string. Now, $y_2 = 1$ so the x_2 bit will be placed at the first place of the third string. The process will be repeated till the last entry of both X and Y strings.

RAPP protocol involves three steps: tag identification, mutual authentication, pseudonym and keys updating. Fig.1 depicts the specifications of RAPP protocol. Basic working of RAPP is as follows:

i) Reader initiates the protocol by sending a '*Hello*' message towards the tag.
ii) Upon receiving the reader's query, tag responds with its IDS.
iii) Reader uses this IDS as an index to search a matched entry in the backend database. If $IDS = IDS^{new}$, then the reader generate pseudorandom number (n_1) and uses ($K_1^{new}, K_2^{new}, K_3^{new}$) to compute A & B messages. If $IDS = IDS^{old}$ then the reader will first generate pseudorandom number (n_1) and uses ($K_1^{old}, K_2^{old}, K_3^{old}$) to compute A & B messages. The message B provides authentication of reader and integrity of the messages. The reader then sends A and B messages towards the tag. However, if IDS does not match with any of the entry in database then the reader will immediately terminate the link as this may be an invalid tag or adversary.
iv) After receiving A & B messages, the tag extracts $n_1$from A and computes a local value of B. If locally computed B equates to the received B; only then the tag will compute and transmit message C towards the reader. Otherwise the tag will do nothing and terminate its protocol session.



$$A = Per(K_2, K_1) \oplus n_1$$
$$B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1)$$
$$C = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID$$
$$D = Per(K_3, K_2) \oplus n_2$$
$$E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2)$$

Pseudonym and keys updating: (**Both reader and 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 1. The RAPP Protocol

v) Upon reception of message C, the reader computes the local value of C and compares locally computed and received C message; if a match occurs only then the reader generates a random number (n_2) and computes D & E messages. Reader also updates the IDS and keys (K_1, K_2, K_3) for future correspondence with the particular tag.

vi) The tag extracts pseudorandom number (n_2) from message D and compute a local value of message E. If locally computed E coincides with received E , then tag will also update its pseudonym (IDS) and keys (K_1, K_2, K_3).

4. Vulnerabilities in RAPP

The attacks presented in this section are inspired from [19, 43] and cryptanalysis are hybrid (combine the assumptions and observations presented in [19, 43]) in nature. This hybrid cryptanalysis model helps in filtering the unwanted results and hence improves the success rate significantly.

First observation which lays the question mark on the security claims of RAPP protocol is that, the reader doesn't know if D and E messages are indeed received or substantiated by valid tag. If D and E messages are not received by the tag then obviously the reader will update its pseudonyms while the tag will keep the previous pseudonym and keys. Secondly, we also know that in RAPP, reader has the capacity to retain the backup values of the pseudonyms while tag can also have the current values of pseudonym and keys. Moreover, while computing the permutation, $Per\ (X, Y)$ the lsb of Y will not affect the overall output of the permutation operation. These security loop holes of RAPP provoke some serious desynchronization, Denial of Service (DoS) and even full disclosure attacks on the protocol. In this section, we have presented three attacks on RAPP : two DoS and one desynchronization attacks, which are as follows:

4.1 Denial of service attack (DOS) ***(Attack 1)*****:**

This is an active attack, since initially adversary intercepts the communication between genuine reader and tag and then replays the modified messages for the proper execution of attack. In RAPP, valid reader initiates the protocol by sending the "$Hello$" message towards the tag. The tag responds with its "IDS". Then the reader looks for this IDS in the database and after validating IDS, it then generates a random number (n_1) and calculates A and B messages. The reader transmits these messages towards the tag. Now, the attacker interrupts the message A and B and modifies the message A to A^* , where $A^* = A \oplus [I]_j$ and $[I]_j$ is $96\ bit$ string that contains all zeroes except on j^{th} location. This alteration will directly toggle the j^{th} bit of (n_1) pseudorandom number; which is concealed in message A. Because of this alteration, tag extracts the altered random number n_1^* and consequently calculates B^* where,

$$B^* = Per\big(K_1 \oplus K_2, Rot(n_1^*, n_1^*)\big) \oplus \mathrm{Per}(n_1^*, K_1) \qquad (1)$$

Now, if the received value of B and computed value B differs then the tag will immediately terminate the communication and will consider the accosting object a counterfeit reader. To make our cryptanalysis successful, we have to alter B in such a way that $B = B^*$; which will be acceptable for the tag. So, consider eq.1 which comprises of two operations: $Per\big(K_1 \oplus K_2, Rot(n_1^*, n_1^*)\big)$ and $\mathrm{Per}(n_1^*, K_1)$. To make our attack simple and plausible, we will firstly describe some observations of permutation (Per) and rotation (Rot) functions.

Observation 1: Permutation operation discloses the information of hamming weight means it is obvious that

$$Hw\big(Per(X, Y)\big) = Hw(X)$$

Observation 2: Let M is the $96 - bit$ string, $M = m_0 m_1 \dots m_j \dots m_n$ and $[I]_j = i_0 i_1 \dots i_j \dots i_n$ (where $[I]_j$ contains all zeroes except on j^{th} location). Now, $M \oplus [I]_j$ will give us two results;

$$\begin{cases} Hw(M) \geq Hw[I]_j & if\ m_j = i_j \\ Hw(M) \leq Hw[I]_j & if\ m_j \neq i_j \end{cases}$$

So, $\Pr\left(Hw(M) \geq Hw[I]_j\right) = \frac{1}{2}$ & $\Pr\ (Hw(M) \leq Hw[I]_j) = \frac{1}{2}$

Observation 3: Let $C = Per\ (A, B)\ and\ C^* = \ Per\ (A, B^*)$ where $B^* = B \oplus [I]_j$ or if $C = Per\ (A, B)$ and $C^* = Per\ (A^*, B)$ then $\Pr(C = C^*) = \frac{1}{2}$. Because in permutation, alteration in the bits position will only change middle part of the resultant, while edges of the resultant remains same. Secondly $B\ or\ B^*$ will not directly affect the overall output of the permutation operation. The proof of this observation has been proposed in [19] and presented in Appendix A.

Now, we turn over to our main issue of altering acceptable B^*.So, according to observation 3: $\text{Per}(n_1^*, K_1) = Per(n_1, K_1)\ \oplus [I]_j$ can be achieved, if an attacker repeats this relationship for some appropriate $(n-2)$ sessions. So, this iterative process will neutralize the effect of bit flipping of pseudorandom number (n_1).Secondly, we can find the relationship $Per\left(K_1 \oplus K_2, Rot(n_1^*, n_1^*)\right) = Per\left(K_1 \oplus K_2, Rot(n_1, n_1)\right)$ if $Rot(n_1^*, n_1^*) = \ Rot(n_1, n_1) \oplus [I]_j$ i.e. $Rot(n_1^*, n_1^*)$ & $Rot(n_1, n_1)$ differs in LSB. This can be computed as follows:

Let $k = Hw(n_1)$ & $k^* = Hw(n_1^*)$, according to observation 1. $\Pr\ (k = k^*) = \frac{1}{2}$ (For both cases)

Hence,

$$Rot(n_1^*, n_1^*) = [n_1 \oplus [I]_j] \ll\ k^*$$
$$= ([n_1] \lll\ k^*)\ \oplus [I]_{j+k}$$

So, when adversary tries all j combinations; it yields $j = -k\ mod\ L$ for some $0 \leq j \leq n-1$.

This causes $Rot(n_1^*, n_1^*) = \ Rot(n_1, n_1) \oplus [I]\ _j$ which infers the following equation realizable

$Per\left(K_1 \oplus K_2, Rot(n_1^*, n_1^*)\right) =$

$Per\left(K_1 \oplus K_2, Rot(n_1, n_1)\right)\quad with\ \frac{1}{2}$ computational probability. Thus the overall success probability is equal to $\frac{1}{4(2^{(n-2)})}$. Now after validating B message, the tag will compute C using n_1^*, which will be rejected by the reader. Hence whenever the tag wants to communicate with the reader, attacker interrupts and fabricates the messages A and B accordingly. Fabricated messages force the tag to extract n_1^* in addition to valid n_1 and consequently after validating B^* ,it will compute C^*; which is unacceptable for further protocol execution. So, in this way attacker will not let the tag to communicate with the legitimate reader thus launching a DoS attack.

4.2 Denial of service attack (DOS) *(Attack 2)*:

In this attack, attacker sends the "*Hello*" message towards tag, and tag responds with its IDS. Then attacker randomly generates and sends A and B messages. The tag extracts n_1 from message A and computes message B to check the correctness of messages. This involves permutation, rotation and XOR operations; which incorporates (ALU) excessive computation and registers to store the intermediate values. Now the adversary engages the tag in this computation by repeatedly (with high frequency) sending the random messages to exhaust the tag as shown in the following fig.2. This will finally lead towards the denial of service attack, since the tag cannot then communicate with the valid reader during this attack. This attack can also be extended to exhaust the valid reader. In that scenario, attacker pretends to be a valid tag and sends random string of IDS with high frequency. On receiving of invalid 'IDS', reader will keep on requesting for the older IDS values. And because of high frequency, it will not able to communicate with the valid tags. The concept of the attack is shown in the following fig.3. The main idea of this attack has inspired from [25], in which authors have proposed the denial of service attack for GOASSMER protocol. However inclusion of counter (messages counter) at tag's side can help to avoid such DoS attacks.

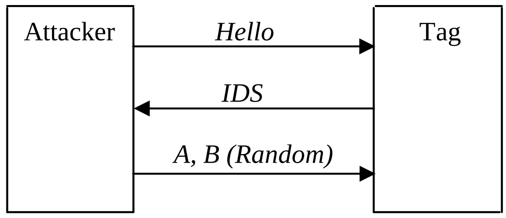


Figure 2. DoS attack on Tag

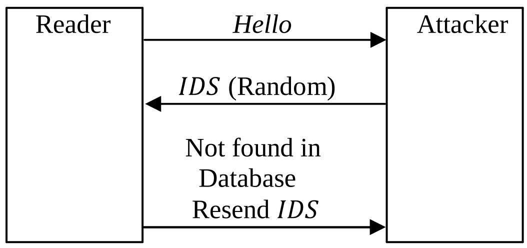


Figure 3. DOS attack on Reader

4.3 Desynchronization attack (Attack 3):

This attack is basically the extension of (DOS) attack 1. Firstly, we assume that both the reader and the tag are synchronized on the same state $S_i(IDS_i, K_{1i}, K_{2i}, K_{3i})$. In RAPP, reader also stores the previous pseudonyms values of state $S_{i-1}\left(IDS_{i-1}, K_{1(i-1)}, K_{2(i-1)}, K_{3(i-1)}\right)$ to combat against the desynchronization attacks. The main purpose of the desynchronization attack is to force both parties to keep different states. In other words desynchronization attack is successful on RAPP, if tag updates S_i^* state while the reader has updated its state to S_i and keeps the S_{i-1} as its previous state. In our proposed attack, initially attacker allows the reader and the tag to run the protocol. Then the attacker stores the whole communicating messages $(A, B, C, D\ and\ E)$ but blocks the $D\ and\ E$ messages from reaching at tag. So, reader has updated its state to S_{i+1} and keeping S_i as its old state while the tag will keep state S_i. Now, attacker starts new protocol run with legitimate tag. Tag transmits its IDS_i to attacker, which then transmits A^*& B^*towards tag, where, $A^* = A \oplus [I]_j$& $B^* = B \oplus [I]_j \oplus [I]_k, 0 \leq j \leq n-1$ & $0 \leq k \leq n-1$ for some appropriate numbers of i and j (Here A & B were pre-captured messages of S_i state). The tag now extracts $n_1^* = n_1 \oplus [I]_j$from A^* and checks the precision of B message. The message B will be accepted if $Per\left(K_1 \oplus K_2, Rot(n_1, n_1)\right) \oplus Per(n_1, k_1) =$

$Per(K_1 \oplus K_2, Rot(n_1^*, n_1^*)) \oplus Per(n_1^*, k_1)$. So, here if $Rot(n_1, n_1) = Rot(n_1^*, n_1^*) \oplus [I]_j$ will differ only in j^{th} bit and same is for $Per(n_1, k_1) = Per(n_1^*, k_1) \oplus [I]_k$ for $(n-2)$ iterations then the overall success probability will be $\frac{1}{2(n-2)}$ as we have discussed in attack.1 and its basic details can be found in Appendix A. Then, after meeting the above condition the tag computes C^* and transmits towards reader (attacker), which can be ignored by attacker. Attacker computes and sends $D^* = D \oplus [I]_j$& $E^* = E \oplus [I]_j \oplus [I]_k$for $0 \le j \le n-1$ & $0 \le k \le n-1$. The tag extracts n_1^* from $D^*$and then check the correctness of E message and E will be accepted if $Per(K_3, Rot(n_2, n_2) \oplus Per(n_1, K_3 \oplus K_2) = Per(K_3,\ Rot(n_2^*, n_2^*) \oplus Per(n_2^*, K_3 \oplus K_2)$ which is actually equivalent to $Per(K_3, Rot(n_2, n_2) \oplus [I]_j \oplus Per(n_1, K_3 \oplus K_2) \oplus [I]_k$.The success probability of the attack can be computed by considering the observations mentioned in attack-1. $D^* = D \oplus [I]_j$ directly toggles the j^{th} bit of n_2 which is then $n_2 \oplus [I]_j$ .Let $L = Hw(n_2)$ and $L^* = Hw(n_2^*)$ which controls the number of rotations in protocols. As per observation-1. $\Pr(L = L^*) = \frac{1}{2}$ (For both cases). So,

$$Rot(n_2^*, n_2^*) = (n_2 \oplus [I]_j) \gg L^*$$

$$=(n_2 \ll L) \oplus [I]_{j+L}$$

(Assuming $L = L^*$). Therefore attacker tries all j combinations; it then yields $j = -L \bmod n$ for some $(0 \le j \le n-1)$. This causes $Rot(n_1^*, n_1^*) = Rot(n_1, n_1) \oplus [I]_j$ and hence results in $Per(K_3, Rot(n_2, n_2) = Per(K_3, Rot(n_2^*, n_2^*)$ with $\frac{1}{2}$ probability. And $Per(n_1, K_3 \oplus K_2) = Per(n_2^*, K_3 \oplus K_2) \oplus [I]_j$ requires $(n-2)$ sessions for coinciding. Table 1 summarizes the proposed attack. Finally, the overall probability will become $\frac{1}{8(2^{(n-2)})}$.

To achieve such situation, attacker have to repeat the scenario for some appropriate j, then if it gets new IDS in next protocol run then it means that tag has accepted invalid pseudorandom numbers. Next time when a valid reader communicates with this tag, the reader will not recognize this tag and hence desynchronize with the particular tag.

Table.1 Changing A& D and conjecturing B&E

```
For j = 0 to n − 2
  For i = 0 to 1
  {Sends hello message to tag;
 Receives IDS_1 from Tag
Sends A*&B* to tag
If receives C from tag then
  For j = 0 to n − 2
   For i = 0 to 1
  {Sends D*&E* to tag;
   Sends hello message to tag
    If receives IDS_2 ≠ IDS_1 then attacker returns
   Successful otherwise repeat the procedure
   }
```

5. Pitfalls in Ultralightweight Mutual authentication protocols

From Section 2, we can observe that the most of the UMAPs are broken within one year (after its introduction). The main reason that shortens the life span of an ultralightweight authentication protocol is that the most of the authors/inventors commit similar mistakes or incorporate weak primitives while designing of an ultralightweight authentication protocols. In this section, we discuss some typical flaws in ultralightweight authentication protocols that frequently undermine the new protocols. These typical pitfalls and recommendations for avoidance are as follows:

5.1 Inclusion of T – functions

A $T-function$ is basically mapping of $n-bit$ input words into $n-bit$ output words (all n output $bits$ depends upon the n input bits) [36]. So, it means all the Boolean functions and logical operations in modern processors (including cryptographic processors) are $T-functions$. Additionally the composition of $T-function$ also results in a $T-function$.

Although these $T-functions$ involve simple computations and considered to be cost effective (in terms of hardware) but these functions exhibit poor diffusion properties [20]. The plain use of these functions (for concealing secrets) is particularly dangerous in cryptographic applications. The only way to address this inadequacy is by combining these operations with other non-triangular primitives (such as Recursive Hash, Rot etc.). But many researchers do not follow this basic combining principle and design protocols entirely based on $T-functions$. The UMAP family (LMAP, EMAP, M^2AP) [3 – 5] and David-Prasad [9] protocols are the examples of such $T-functions$ dependent insecure protocols.

5.2 Linearity

Linearity should also be avoided or dealt with carefully while designing of such ultralightweight authentication protocols. Formally, an operation $'g'$ is considered to be linear if $g(x \oplus y) = g(x) \oplus g(y)$. Inclusion of such linear operations in protocol designs provide well defined platform for successful cryptanalysis of the protocol. So, to avoid linearity either we should analyze bitwise message designs or incorporate hybrid ultralightweight primitives in protocol designs. The RCIA and R^2AP [7, 8] are the state of art UMAPs which involve hybrid ultralightweight primitives in their designs to avoid linearity.

5.3 Biased operators

Another important weakness of many ultralightweight protocols is that some of the operations used have biased output results. For example the logical operations such as $AND(\wedge), XOR(\oplus)\ and\ OR(\vee)$ based internal computational operations give similar results, where $a \oplus b$ and $a \vee b$ give identical results with 75% of success rate and similarly $a \oplus b$ and $\overline{a \wedge b}$ also result the identical output with 75% of success rate. This can constitute potential security threat because these logical operations reveal information for both of their variables. For example, in David-Prasad protocol [9] if we take XOR between its two publically disclosed messages E and F then we can disclose its secret ID with 75% success rate. Again the combining of non-triangular primitives with

Boolean functions avoid the biasness in the results and hence provide significant security.

5.4 Weak Primitives

Most of the researchers use weak or linear non-triangular primitives such as Permutation and Rotation in designing of their new protocols. For permutation we have already highlighted in section 4 that it reveals the information of hamming weight (hw) of the first parameter (operand), which cause desynchronization or even full disclosure attacks. Since Rotation function is extremely lightweight in nature so most of the block ciphers and hash functions still mostly rely on ARX (Addition Rotation and XOR) [32] designs. Typically, there are two types of rotations: Modular rotations and Hamming weight based rotations. Modular rotations have better entropy ($log_2 n = 6.6$) since each shift is equiprobable and considerably robust. While the hamming – weight based rotations have worse entropy ($log_2 n = 4.4$) which means that the number of bits rotated is between 31 and 64 in 99% of the cases. (where $n = 96\ bits$ [35])
Actually, the rotation operation is a permutation and therefore it also exhibits the pitfalls of linear functions.
In ultralightweight authentication protocols, mostly the rotation operations are data dependent which then have only n possible outputs. Hence Permutation and Rotation operations should not be used alone as they reveal the information of hamming – weight of the variables.

5.5 Poor messages Composition

Designing of secure messages exchanged over an ultralightweight protocol is a difficult task, particularly in such constraint environment. Generally speaking the publically disclosed messages should guarantee good confusion and diffusion properties of the secrets. In typical cryptographical algorithms, these two properties are achieved by using iterated substitution and permutation blocks. However, due to limited computational capabilities of passive low cost tags, messages are usually designed by using $T - functions$ and some special purpose primitives, which give insufficient level of confusion and diffusion for secrets.
In M^{2}AP [5] for instance, the IDS update phase is defined as:

$$IDS^{next} = (IDS + (n_1 \oplus n_2)) \oplus ID$$

Where we can see that the tag's static ID is simply XORed with mixture of secret and publically known variables. This operation clearly exhibits poor confusion and diffusion properties which may leads to major leakage of the secrets. Moreover, the messages should be carefully design enough so, when an adversary applies multiple logical operations between publically disclosed messages then it should not reveal any secret information.

5.6 Desynchronization

Usually, the desynchronization attacks are active and occur because of poor structure (design) of the protocol. In this paper, we have also highlighted the same attack in RAPP protocol. Almost all the previously proposed ultralightweight protocols have been shown to be vulnerable to desynchronization attacks. The main reason behind this dilemma is the missing of previously computed pseudonym (IDS) and keys values either at tag or reader side. Usually cryptanalysts exploit this weakness of the UMAPs and hence make both the reader and tag desynchronize. The storing of an extra copy of keys and pseudonym is the only optimal way to avoid such desynchronization attacks.

5.7 Recommendations for Security analysis

Security analysis of the proposed protocol is considered as an integral part of the protocol, which mainly highlights the robustness of the protocol over various attack models and scenarios. Many researchers use typical formal cryptanalysis models such as BAN [44], GNY [45] and AVISPA [46] etc. However such typical cryptanalysis models does not work as intended and despite being accompanied by formal security proof in such formal models was broken shortly For example in [48] authors incorporate BAN logic to formally analyze their CRC based ultralightweight protocol. But instead of using them as simple error detection tool, they employed them for encryption, so some of the BAN logic rules do not hold any more. Some of the authors use AVISPA to evaluate the EPC C1G2 protocol (LMAP) [47]. AVISPA discovered only two attacks in LMAP and authors proposed simple patch to overcome the highlighted loopholes. But literature shows that [26, 29] the LMAP has received multiple attacks and vulnerable to many cryptanalysis models even in the presence of the extended patch.
So, it is recommended that the formal security analysis of the UMAPs should be performed with Tango [21], Recursive Linear Cryptanalysis (RLC), Recursive Differential Cryptanalysis (RDC) [20] and Grobner Basis attacks [24] models. The protocol will be considered robust and secure only if it satisfies the above mentioned structural cryptanalysis (which are specifically designed for UMAPs).

6. Conclusion

In this paper, we have analyzed the vulnerabilities of RAPP protocol and highlighted three attacks in RAPP: Two DoS and one Desynchronization attacks. The proposed attacks are inspired from [19, 43] and improves the success rate of the attack by combining both approaches. We have also discussed some prudent engineering practices and offered recommendations to follow, together with typical mistakes to avoid, when designing of ultralightweight authentication protocols. This will help as sanity check to improve the security and reliability of the new proposals.

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] 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.
[3] Pedro Peris-Lopez, Julio Hernandez-Castro et.al. "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. "The second Workshop on RFID Security, Austria, pp.100-112, 2006.
[4] 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.
[5] P. Peris-Lopez, J.C. Hernandez- Castro, J.M.E. Tapiador, A. Ribagorda, "M2AP: a minimalist mutual-authentication protocol for low cost RFID tags", International Conference on

Ubiquitous Intelligence and Computing, Wuhan China, pp. 912–923, 2006.
[6] Peris-Lopez, Pedro, et al. "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol." The 9th International Workshop on Information Security Applications, Korea, pp. 56-68, 2009.
[7] Umar Mujahid, M. Najam-ul-Islam, and M. Ali Shami, “RCIA: A New Ultralightweight RFID Authentication Protocol Using Recursive Hash,” International Journal of Distributed Sensor Networks, vol. 2015, No. 642180, 8 pages, 2015.
[8] Xu Zhuang, Yan Zhu and Chin-Chen Chang,” A New Ultralightweight RFID Protocol for Low-Cost Tags: R ^{2}AP”, Wireless Personal Communications, Vol. 79, No.3, pp 1787-1802, 2014.
[9] 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, pp. 172-179, 2009.
[10] Yeh, Kuo-Hui et al. ."An efficient ultralightweight authentication protocol for RFID systems." The 4th International Workshop on RFID Security and Privacy, Turkey, pp 49-60, 2010.
[11] Soo.Jeon et.al,” A New Ultra-lightweight RFID Authentication Protocol Using Merge and Separation Operations”, Int. Journal of Math. Analysis, Vol. 7, No. 52, pp 2583 – 2593, 2013.
[12] 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.
[13] 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.
[14] 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.
[15] Zeeshan Bilal, Keith Martin and Qasim Saeed,” Multiple Attacks on Authentication Protocols for Low-Cost RFID Tags”, Applied Mathematics & Information Sciences, Vol.9, No. 2, pp. 561-569, 2015.
[16] Soo Jeon and Eun-Jun Yoon,” Cryptanalysis and Improvement of a New Ultra-lightweight RFID Authentication Protocol with Permutation” Applied Mathematical Sciences, Vol. 7, 2013, No. 69, pp. 3433 – 3444, 2013.
[17] Sun, Hung-Min, Wei-Chih Ting, and King-Hang Wang. "On the Security of Chien's Ultralightweight RFID Authentication Protocol." IEEE Transactions on Dependable & Secure Computing, Vol.8, No.2, pp.315-317, 2011.
[18] Umar Mujahid, M.Najam-ul-islam,” Ultralightweight Cryptography for Passive RFID Systems”, International Journal of Communication Networks and Information Security”, Vol.6, No.3, pp.173-181, 2014.
[19] Zahra Ahmadian, Mahmoud Salmasizadeh and Mohammad Reza Aref, "Desynchronization attack on RAPP ultralightweight authentication protocol." Information processing letters, Vol.113, No.7, pp. 205-209, 2013.
[20] 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.
[21] 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.
[22] 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.
[23] 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.
[24] Han, Daewan, "Gröbner Basis Attacks on Lightweight RFID Authentication Protocols." Journal of Information Processing Systems, Vol. 7, No.4, pp.691-706, 2011.
[25] 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.
[26] Li, Ticyan, and Guilin Wang. "Security analysis of two ultra-lightweight RFID authentication protocols." International Information Security Conference (SEC), South Africa, pp.109-120, 2007.
[27] D'Arco, Paolo, and Alfredo De Santis. "On ultralightweight RFID authentication protocols." IEEE Transactions on Dependable and Secure Computing. Vol 8, No.4, pp. 548-563, 2011.
[28] Muhammad Zubair, Umar Mujahid, Najam-ul-Islam and Jameel Ahmed, "Cryptanalysis of RFID Ultra-lightweight Protocols and Comparison between its Solutions Approaches", BUJICT Journal, Vol.5, No. 1, pp. 58-63, 2012.
[29] Tieyan Li et al. “Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol”, The Second International Conference on Availability, Reliability and Security (ARES 2007), Vienna, pp. 224 – 231. 2007.
[30] Tieyan Li and Guilin Wang,” Security Analysis of family of Ultra-Lightweight RFID Authentication Protocols”, Journal of Software, Vol. 3, No. 3, 2008.
[31] Wang Shao-hui et al. ,” Security Analysis of RAPP: An RFID Authentication Protocol based on Permutation”, Cryptology ePrint Archive, Report 2012/327, https://eprint.iacr.org/2012/327 , 2012.
[32] D.Khovratovich et al.,”Rotational cryptanalysis of ARX”, The 17th international conference on fast software encryption (FSE-2010), Korea, pp.333-346, 2010.
[33] Mehmet Hilal et al. ,” Mersenne twister-based RFID authentication protocol”, Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 23, pp. 231 – 254, 2015.
[34] Zeeshan Bilal and Keith Martin,” Ultra-lightweight Mutual Authentication Protocols: Weaknesses and Countermeasures”, International conference on availability, reliability and security, Germany, pp.304-309, 2013.
[35] GS1 EPCglobal tag data standards version 1.4, Available from; http//www.epcglobalinc.org/standards/.
[36] A. Klimov and A. Shamir. “New Applications of T-Functions in Block Ciphers and Hash Functions”, Fast Software Encryption, France, pp. 18–31, 2005.
[37] G. Marsaglia and W.W. Tsang. “Some difficult-to-pass tests of randomness”, Journal of Statistical Software, Vol. 7, No. 3, pp.37–51, 2002.
[38] J. Walker. ENT Randomness Test, Available from: http://www.fourmilab.ch/ random/, 1998.
[39] C. Suresh, Charanjit J., J.R. Rao, and P. Rohatgi,” A cautionary note regarding evaluation of AES candidates on smart-cards”. The Second Advanced Encryption Standard (AES) Candidate Conference., Italy, Available from: http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm, 1999.
[40] Pedro Peris López,” Lightweight Cryptography in Radio Frequency Identification (RFID) Systems”, PhD thesis, UNIVERSIDAD CARLOS III DE MADRID, 2008.
[41] Julio C. Hernandez. et.al "Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations.", ArXiv, Cryptography and Security, Report; Report No. 0811.4257, http://arxiv.org/abs/0811.4257, 2008.
[42] Peris-Lopez, Pedro, Julio C. Hernandez-Castro, Juan ME Tapiador, and Jan CA van der Lubbe. "Security flaws in a recent ultralightweight RFID protocol." arXiv preprint (Technical Report) No. 0910.2115, 2009.

[43] Zhuang, X., Wang, Z. H., Chang, C. C., & Zhu, Y,"Security analysis of a new ultra-lightweight RFID protocol and its improvement", Journal of Information Hiding and Multimedia Signal Processing Vol. 4, pp. 165–180, 2013.
[44] Wessels, J., and CMG FINANCE BV. "Applications of BAN logic." Available from: http://www. win. tue. nl/ipa/activities/springdays2001/banwessels, 2001.
[45] Pieter and Michiel,"Analysis of the OpenPGP and OTR protocols", using GNY logic (Online Tutorial), Available from: http://www.ai.rug.nl/mas/finishedprojects/2007/PieterMichiel/gny.html
[46] Alessandro Armando, David Basin, Yohan Boichut et al. "The avispa tool for automatic validation of internet security protocols and applications" The 17th international conference on Computer Aided Verification, UK, pp. 281 – 285, 2005.
[47] Salekul Islam,"Security analysis of LMAP using avispa" International Journal of Security and Networks, Vol. 9, No. 1, pp 30 – 39, 2014.
[48] Cai Qingling, Zhan Yiju and Wang ,"A minimalist Mutual Authentication Protocol for RFID systems &BAN logic analysis", International Colloquium on Computing, Communication, Control and Management, Guangzhou, pp 449 – 453, 2008.
[49] Rkia Aouinatou, Mostafa Belkasmi and Mohamed Askali," A dynamic study with side channel against An Identification Based Encryption", International Journal of Communication Networks and Information Security, Vol. 7, No.1, 2015.

APPENDIX

Appendix A

Observation 3 (Proof) [19]:

Hence as, $C = Per\ (A, B)\ and\ C^* = Per\ (A, B^*)$ where $B^* = B \oplus [I]_j$ or if $C = Per\ (A, B)\ and\ C^* = Per\ (A^*, B)$ then $\Pr(C = C^*) = \frac{1}{2}$

Proof: Let $R_1 = \{j_1, j_2, \dots j_m\}$ is the set of indexes whose corresponding bit position in y is 1, and $R_0 = \{j_{m+1}, j_{m+2}, \dots j_L\}$ is the set of indexes whose corresponding bit position is 0. Then Permutation will be $C = Per\ (A, B) = A_{jm}, A_{jm-1} \dots A_{j2} A_{j1} \dots . A_{jL}$

Now, consider the following two cases:

Case-1: If last two bits of Y are not same; then $j_1 = 0, j_{m+1} = 1$ or $j_1 = 1, j_{m+1} = 0$, in both cases the set of indexes will be $R_1 = \{j_{m+1}, j_2, \dots j_m\}$ & $R_0 = \{j_1, j_{m+2}, \dots j_L\}$. Thus: $C = Per\ (A, B) = A_{jm}, A_{jm-1} \dots A_{j2} A_{jm+1} A_{j1} A_{jm+2} \dots . A_{jL}$

Case-2: If last two bits of Y are same then $j_1 = 0, j_2 = 1$ or $j_{m+1} = 0, j_{m+2} = 1$, and set of indexes will be $R_1 = \{j_3 \dots j_m\}$, $R_0 = \{j_1, j_2, j_{m+1}, \dots j_L\}$, $R_1 = \{j_{m+1}, j_{m+2}, j_3 \dots j_m\}$, $R_0 = \{j_{m+3}, \dots j_L\}$ respectively.

Hence $C = Per\ (A, B) = A_{jm}, A_{jm-1} \dots A_{j1} A_{j2} \dots . A_{jL}$

or $C = Per\ (A, B) = A_{jm}, A_{jm-1} \dots A_{j2} A_{j1} \dots . A_{jL}$.