Wireless Pers Commun (2017) 94:725–744
DOI 10.1007/s11277-016-3647-4

A New Ultralightweight RFID Authentication Protocol for Passive Low Cost Tags: KMAP

Umar Mujahid[1] · **Muhammad Najam-ul-Islam**[1] · **Shahzad Sarwar**[2]

Published online: 2 September 2016

Abstract Radio Frequency IDentification (RFID) is one of the most promising identification schemes in the field of pervasive systems. Unique identification and non-line of sight capabilities make RFID systems more protuberant than its contending systems. As RFID systems incorporate wireless channel, there are some allied security threats and apprehensions to the systems from malicious adversaries. In order to make the system reliable and secure, numerous Ultralightweight Mutual Authentication Protocols (UMAPs) have been proposed which involve only simple bitwise logical operations (AND, XOR & OR etc.) in their designs. However, almost all of the previously proposed UMAPs are reported to be vulnerable against various security attacks (Desynchronization and Full disclosure attacks etc.). In this paper, we propose a new pseudo-Kasami code based Mutual Authentication Protocol (KMAP). The proposed protocol, KMAP, avoids unbalanced logical operations (OR, AND) and introduces a new Ultralightweight primitive: pseudo-Kasami code (K_c). The newly proposed primitive (pseudo-Kasami code) enhances the diffusion properties of the protocol messages and makes hamming weight of the secrets unpredictable and irreversible. The security analysis illustrates that the KMAP provides excellent protocol functionalities and is also highly resistive against all possible attacks. The performance evaluation shows that the KMAP requires fewer resources on the tag in terms of on-chip memory, communication cost and computational operations.

Keywords Ultralightweight · RFID · Synchronization · KMAP · Passive tags

✉ Umar Mujahid
umar.mujahid@bui.edu.pk

Muhammad Najam-ul-Islam
najam@bahria.edu.pk

Shahzad Sarwar
s.sarwar@pucit.edu.pk

[1] Department of Electrical Engineering, Bahria University, Islamabad, Pakistan

[2] College of Information Technology, Punjab University, Lahore, Pakistan

1 Introduction

RFID is an automatic identification scheme which mainly comprises of three components: tags, readers and back-end database. In normal scenario, the reader acts as a scanner and enquiries all tags entering in its vicinity. Upon receiving the reader's query, the tags respond with their identity "ID". The reader uses it as an index to search for a matched entry in its database. If both the values coincide, only then the tag can have access to RFID associated particular systems. It is assumed that the channel between the reader and back-end database is secure since we can incorporate traditional cryptographic algorithms to secure that channel. However, the channel between the reader and the tag needs more attention, because limited computational capabilities at tag's side restrict us to use simple bitwise logical operations (*T-functions* [1]) for security. Typically, such tags can store 32–1 K bits and can support 250–4 K logic gates for security related tasks [2]. Table 1 summarizes the attributes of such low cost RFID tags.

To secure the channel between the reader and the tag, researchers have proposed various cryptographic solutions including mutual authentication protocols. Based on the computational capabilities at tag's side, the authentication protocols have been classified into four categories: Full–fledged, Simple, Lightweight and Ultralightweight.

1. Full-fledged protocols can incorporate the traditional cryptographic algorithms and solutions like one way hash functions, public or private key cryptography etc.
2. Simple authentication protocols can support pseudorandom number generators and one-way hash functions only.
3. Lightweight protocols can support only lightweight pseudorandom number generators and simple functions such as Cyclic Redundancy Check (CRC) but cannot use hash functions.
4. Ultralightweight protocols can incorporate only simple bitwise logical operations and even pseudorandom number generators can't be used at the tag's side.

For secure communication of the low cost RFID systems, we use Ultralightweight Mutual Authentication Protocols (UMAPs). These UMAPs provide extremely low security due to extensive use of simple *T-functions* in protocols design. However, inclusion of non-

Table 1 Properties of passive low cost RFID tags

S. No.	Performance metrics	Attributes
1.	Standard/regulation	EPC-C1G2/ISO 18000/6
2.	Memory storage	32–1 K bits
3.	Overall logic GE	5–10 K
4.	Logic GE for security related tasks	250–4 K
5.	Power source (active/passive)	Passive
6.	Price	0.05–0.1$
7.	Resistance to passive attacks	Yes
8.	Resistance to active attacks	No (but depends upon the protocol)
9.	Resistance to physical attacks	No
10.	Class of authentication protocols	Ultralightweight

triangular operations in protocols augments the resistance against various types of security attacks.

In this paper, we propose a new Ultralightweight Primitive (UP): pseudo-Kasami code (K_c) based mutual authentication protocol (KMAP). The inclusion of new UP (K_c) in protocol design not only enhances the diffusion properties of the messages but also makes hamming weight unpredictable and irreversible for adversaries.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 introduces the novel protocol (KMAP) followed by the security analysis in Sect. 4. Section 5 highlights the performance analysis of the proposed protocol. Finally Sect. 6 concludes the paper.

2 Related Work

Security and privacy are the two major concerns of RFID based identification systems which are associated with the tag's cost. On the basis of the tag's cost and computational capabilities, the RFID tags can be classified into two types: high and low cost tags. Our research work focuses on low cost RFID tags. High cost tags are resourceful enough to support traditional cryptographic algorithms and primitives such as AES, hash functions, stream ciphers etc. for security related tasks. However, such conventional cryptographic algorithms and primitives have excessive power, memory and silicon (chip) area requirements which are beyond the low cost tag's computational capabilities. Hence, a new field Ultralightweight cryptography has been introduced to ensure the security of low cost RFID tags in recent years. Ultralightweight cryptography avoids the use of costly operations and supports only simple *T-functions* and some special purpose UPs for security. Many UMAPs have already been proposed during last decade. However, almost all the previously proposed protocols have some serious security flaws and reported to be vulnerable within one year after their introduction [3]. A comprehensive survey of ultralightweight protocols and their vulnerabilities is presented as follows.

In 2006, Peris-Lopez et al. [4–6] laid the foundation of ultralightweight cryptography for passive RFID systems. The authors 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. Therefore, they proposed three extremely lightweight mutual authentication protocols (named UMAP family): Lightweight Mutual Authentication Protocol (LMAP), Minimalist Mutual Authentication Protocol (M^2AP) and An Efficient Mutual Authentication Protocol (EMAP) for low cost passive RFID tags. The UMAP family protocols involve 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 shows that the LMAP requires only 300 gates while EMAP and M^2AP require only 150 and 300 gates respectively. Moreover, the randomness of the protocol messages was ensured with three randomness test suites: DIEHARD [7], ENT [8] and NIST [9]. In 2007, Tieyan Li et al. [10, 11] performed security analysis of the UMAP family protocols. They exploited the inherent weak diffusion properties of *T-functions* and found two effective attacks on the protocols: desynchronization and full disclosure. The former permanently abolishes the authentication capability of the tag while later completely discloses all the concealed secrets stored in the tag.

In the same year, Chein [12] introduced the concept of non-triangular primitive 'Rotation (Rot)' in protocol messages and proposed an ultralightweight RFID authentication protocol: Strong Authentication and Strong Integrity (SASI). The 'Rot' function requires only two registers for its operation and hence proves to be extremely lightweight operation. However, it is a clock cycle consuming function, since for each rotation 'l' clock cycles are required (where 'l' is the number of bits in both strings). Right after SASI's introduction, Avoine et al. [13], Sun et al. [14] and Hernandez et al. [15] reported various desynchronization and full disclosure attacks on the protocol. Hence the success journey of the SASI protocol also got over shortly.

Later, GOASSMER [16], David–Prasad [17], Yeh et al. [18] and Lee et al. [19] protocols were also found to be vulnerable against various desynchronization, traceability and full disclosure attacks [20–23].

In 2012, Tian et al. [24] presented a more sophisticated non-triangular primitive "Permutation" (Per) and proposed a new UMAP using Permutation (RAPP). Initially, the 'Per' operation seemed to be highly effective and extremely lightweight, later it was explored that it reveals the information of hamming weight (hw) of the operand. In 2013, Shao et al. [25] and Ahmadian et al. [26] exploited this inherent weakness of the permutation operation and highlighted full disclosure and desynchronization attacks on the RAPP, enlisting RAPP among vulnerable UMAPs.

In 2013, Jeon et al. [27] extended the concept of permutation (Per) and introduced two similar primitives: Merge (*Mer*) and Separation (*Sep*). These new primitives were extensively used in the RAPLT (RFID Authentication Protocol for Low cost Tags) protocol messages to ensure the integrity and confidentiality of the messages. However, Zhuang et al. [28] showed that RAPLT can't avoid even a simple desynchronization and replay attack models and hence it is as vulnerable as its contending protocols.

In Ref. [2, 4–6, 12, 16–19, 24, 27, 29–33] numerous ultralightweight RFID authentication protocols have been presented with several variations in their designs however, cryptanalysis performed in [10, 11, 13–15, 20–22, 25, 26, 28, 34–45] highlight the pitfalls in almost all the previously proposed ultralightweight authentication protocols. This paper aims to propose a new ultralightweight RFID authentication protocol for extremely low cost passive RFID tags which avoids all the previously highlighted pitfalls and should withstand against all possible attacks (active and passive).

3 KMAP: A New Ultralightweight RFID Authentication protocol

In this section, we propose a new pseudo-Kasami code based Mutual Authentication Protocol: KMAP. In KMAP, tags avoid all unbalanced operators and involve only two ultralightweight balanced operators [*XOR* and circular left rotation (*Rot*)]. The newly proposed ultralightweight primitive: pseudo Kasami code (K_c) involves only Rot and *XOR* operations and proves to be extremely lightweight. The pseudo Kasami code (K_c) of any variable can be computed as follows:

Computation of pseudo-Kasami code (K_c);

Suppose X is a 'n' bit string, where:

$$X = x_1 x_2, \ldots, x_n, \quad x_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n$$

Then computation of the pseudo-Kasami code of $X, K_c(X)$ involves following two steps:

1. The tag extracts the random numbers (n_1, n_2) from messages sent by the reader. The tag then calculates the seed for pseudo-Kasami code in following manner:

   (a) Compute $P = n_1 \oplus n_2$.
   (b) Seed $= hw(P) mod K$
   (For EPC-Class-1 Generation-2 tags [2], $n = 96$ bits and for efficient hardware implementation, $K = 64$) Hence, the $Seed \in \{0, 1, 2 \ldots, 63\}$.

2. Seed (calculated in Step 1) selects the number of bits of the string X and performs following operations to compute the final pseudo-Kasami code, $K_c(X)$:

   (a) Computes a new shifted string $X^{'}$ (shifted version of X) by cyclic left rotation of the selected number of bits of X.
   (Seed $\in \{0, 1, 2, \ldots, 63\}$; Seed $= 0$ represents last one bit cyclic left rotation and Seed $= 63$ represents last 64 bits cyclic left rotation).
   (b) Take XOR between X and $X^{'}$.

To better understand the concept of Pseudo Kasami codes consider the following example:

Example Given $X = 100101110101101100010110$ and assume seed $= 6$, Then pseudo-Kasami code of $X, K_c(X)$ will be:

$$K_c(X) = 110011010000011001111010$$

Figure 1 shows the computation of the above example.

Like typical RFID systems, KMAP also involves three entities: tag, reader and backend database. The communication link between the reader and the backend database is assumed to be secure since here traditional cryptographic algorithms can be used. The link between the reader and the tag needs more attention because of resources constraints at the tag's end. Each tag has one static *ID*, pseudonym *IDS* and two keys K_1 *and* K_2. The length of each variable is of 96-bits . To avoid the possible desynchronization attacks, both the

$X = 100101110101101100010110$ and Seed $= 6$;

a) Bits selection: As seed$= 6$, so last six bits of X will be selected for cyclic left rotation.

$X = 100101110101101100\ \boxed{010110}$

b) Computation of shifted version string: Cyclic left rotate the selected bits and compute a new string , $X^{'}$:

$X^{'} = \boxed{010110}\ 100101110101101100$

c) Take XOR between X and its shifted version $X^{'}$:

$$\begin{array}{rl} X = & 100101110101101100010110 \\ \oplus\ X^{'} = & 010110100101110101101100 \\ \hline K_c(X) = & 110011010000011001111010 \end{array}$$

Fig. 1 The computation of pseudo-Kasami code (example)

reader and the tag keep the two entries (old and updated) of pseudonyms and keys (IDS, K_1, K_2).

A protocol session (message) counter has also been integrated in KMAP which stops the functionality of the tag for some particular time, if counter's value exceeds threshold (≤ 8). The specification of KMAP is shown in Fig. 2. The working details of the protocol are as follows:

1. The reader sends a "*Hello*" message to the tag and initiates the protocol session.
2. On receiving the reader's query, the tag responds with its *IDS*.
3. The reader uses this *IDS* as an index and searches for a matched entry in the database. If received *IDS* is IDS_{old} then reader uses $(K_{1,old}, K_{2,old})$ for computation of *A*, *B* and *C* messages.

If received *IDS* is new one (IDS_{new}) then the reader uses $\left(K_{1,new} \,\&\, K_{2,new}\right)$ for the computation of *A, B* and *C* messages. Further, the reader generates two random numbers (n_1, n_2) and conceals them in messages (*A* and *B*). The reader also calculates $P = n_1 \oplus n_2$ and seed for computation of pseudo-Kasami codes of the variables(n_1, n_2, K_1, K_2). The seed is computed by taking *mod*of hamming weight of *P* given by $hw(P)\ mod\ K$. The reader uses the calculated K_c of the variables for the computation of message *C* Finally the reader sends $A||B||C$ messages to the tag. However, if the *IDS* is not in the database then the reader immediately terminates protocol session with the particular tag.

4. Upon receiving of $A||B||C$ messages, the tag performs following operations:

   (a) Extracts random nonce (n_1) from message *A*:

$$n_1 = Rot^{-1}\left(Rot^{-1}(A, K_2), IDS \oplus K_1\right) \tag{1}$$

   (b) Extracts random nonce (n_2) from message *B*:

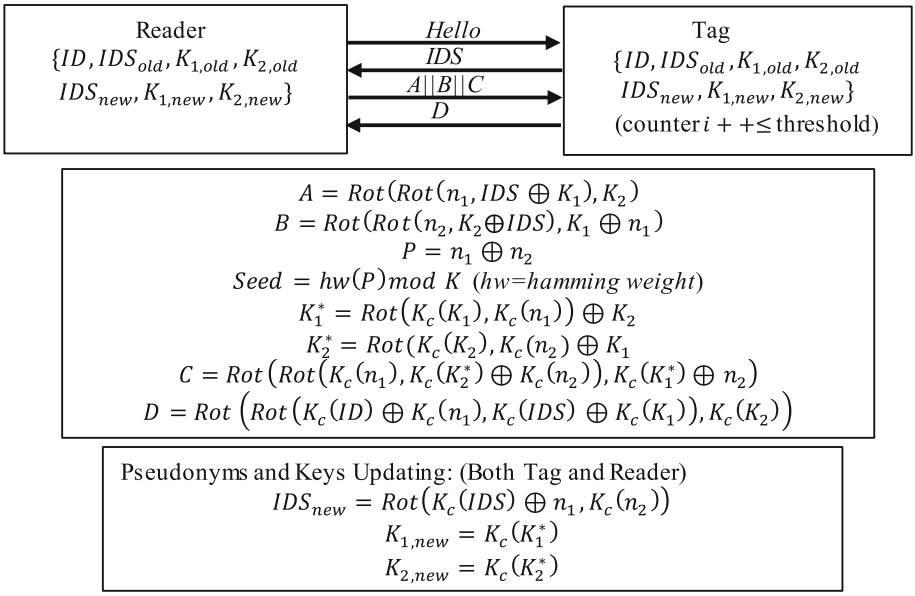

Fig. 2 KMAP protocol

$$n_2 = Rot^{-1}\left(Rot^{-1}(B, K_1 \oplus n_1), K_2 \oplus IDS\right) \quad (2)$$

(c) Calculates seed for computation of pseudo-Kasami Codes using $P = n_1 \oplus n_2$ and $hw(P) mod K$.

(d) Calculates internal keys $\left(K_1^* \& K_2^*\right)$ and computes the local value of message C, C^*. The tag then compares the calculated C^* with the received C. If both the values coincide, then the tag further performs two tasks: 1) transmits D message to the reader and 2) updates its pseudonym (IDS) & keys $(K_1 \& K_2)$.

5. On receiving of message D the reader computes a local value of D and if both values coincide only then the reader also updates its pseudonym (IDS) and keys $(K_1 \& K_2)$ in its database for future correspondence with the particular tag.

The statistical properties of the messages A, B, C and D have been analysed with Diehard [7], ENT [8], and NIST [9] randomness tests. We have generated 300 MB file of each message and some of the results are shown in Table 2. We can observe from the table that messages are purely random and are not easily distinguishable from a random source.

4 Security Analysis

We have analysed the security of the KMAP protocol in two main aspects: basic functionalities of the protocol and resistance against various cryptanalysis models. The functionalities of the protocol include mutual authentication, tag content privacy and integrity. We have considered desynchronization, replay, traceability, adhoc based full disclosure attacks and formal structural cryptanalysis attacks. The brief description of the security analysis is presented in Fig. 3.

4.1 Functionality of the Protocol

1. *Mutual authentication* The valid readers and the valid tags can only authenticate each other. The transmitted messages A, B, C and D are composed of preshared secret ID and keys $(K_1 \& K_2)$. Only genuine pair of the reader and the tag can generate these messages which will be authenticated by each party.

   (a) *Authentication of the reader* In KMAP, the reader is authenticated by checking the legitimacy and correctness of the message C. This message is composed of

Table 2 Randomness test with ENT, Diehard and NIST

	A	B	C	D
Entropy (bits/bytes)	7.96998	7.99769	7.99987	7.98879
Compression rate	0 %	0 %	0 %	0 %
X^2 statistic	254.87	256.45	242.96	248.24
Arithmetic mean	128.273	127.659	127.618	127.537
Monte Carlo π estimation	3.14628	3.1482	3.14763	3.14693
Diehard battery (overall p value)	0.5831	0.4951	0.4691	0.5949
NIST battery	Pass	Pass	Pass	PASS

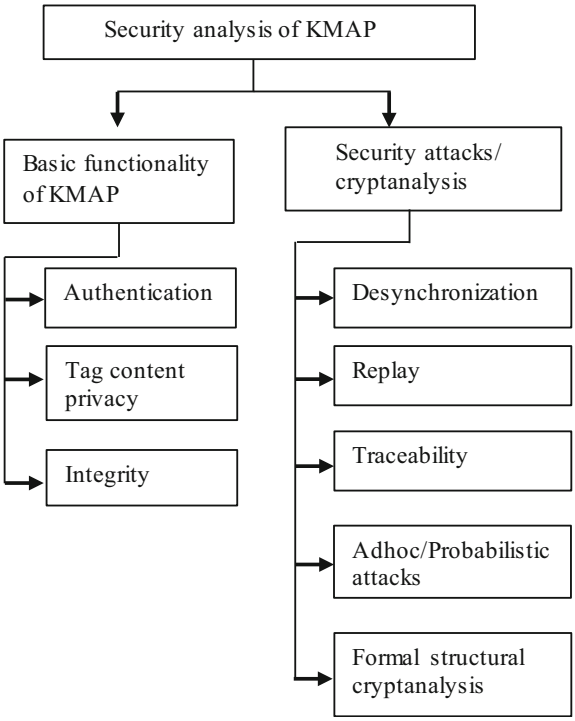

Fig. 3 Security analysis model

preshared secret keys $(K_1 \& K_2)$ and pseudorandom numbers $(n_1 \& n_2)$. Therefore only the valid tag can check the precision of the message C. The correctness of the message C thus determines the authenticity of the reader.

(b) *Authentication of the tag* Once the tag authenticates reader it transmits its shared secrets in the form of message D. Only the valid reader can check the legitimacy of the message D. The correctness of the message D determines the authenticity of the tag.

2. *Tag content privacy* Each tag is assigned a unique static identity (*ID*), which is used for correspondence (connectivity) with the reader. The main objective of the mutual authentication protocols is to secure *ID* of the tag from adversary and other malicious activities. In KMAP, the tag ensures the confidentiality by encrypting the publically disclosed messages A, B, C and D with preshared secret keys $(K_1$ and $K_2)$. The freshness of the messages is ensured by pseudorandom numbers $(n_1$ and $n_2)$. Only the genuine reader with the knowledge of tag's *ID* and secrets keys $(K_1$ and $K_2)$ can read the contents of the tag. Khovratovich et al. [45] proposed theoretical framework for the security analysis of the Addition, Rotation and XOR (ARX) based systems. They used rotational cryptanalysis for the security appraisal of ARX systems and showed that systems with less than $t/log_2(P_r)$ rotations/additions/XOR operations are vulnerable to the rotational cryptanalysis. In KMAP, pseudo Kasami codes (K_c) make extensive use of the rotation and XOR operations and firmly avoid rotational cryptanalysis. Moreover, the optimal composition of the protocol messages (A, B, C and D) enhances the complexity for computation of conjecture information (*ID*, pseudorandom numbers and Keys) for adversary. The complexity of recovering conjecture n_1, n_2 and *ID* from

publically disclosed messages is as follows.

Complexity of Recovering n_1 and n_2:

The pseudorandom numbers n_1 and n_2 are concealed in messages (A and B):

$$A = Rot(Rot(n_1, IDS \oplus K_1), K_2) \tag{3}$$

$$B = Rot(Rot(n_2, K_2 \oplus IDS),\ K_1 \oplus n_1) \tag{4}$$

The goal of the adversary is to disclose n_1 and n_2 from Eqs. 3 and 4 respectively. The overall complexity can be computed in following manner:

(a) The outer rotation of Eq. 4 is undone with complexity $O(log_2(K_2)$:

$$\begin{aligned} L &= Rot^{-1}(A, K_2) \\ &= Rot(n_1, IDS \oplus K_1) \end{aligned} \tag{5}$$

(b) Adversary requires a complexity $O(2^{K_1} \times log_2(K_2))$ to *XOR* all possible values of K_2 from *IDS* to get a optimal solution.

$$M = IDS \oplus K_1 \tag{6}$$

(c) Now Inner rotation of Eq. 5 is undone by using all corresponding $2^{K_1} \times log_2(K_2)$ values. This doubles the complexity as $O(2 \times 2^{K_1} \times log_2(K_2))$.

$$Rot^{-1}(L, M) = n_1 \tag{7}$$

(d) For further computation of n_2, adversary requires the complexity $O(2^2 \times 2^{K_1} \times log_2(K_2))$ to XOR all possible values of K_1 and n_1 to get optimal results.

$$S = K_1 \oplus n_1 \tag{8}$$

(e) By taking the inverse rotation of Eq. 2, the complexity for adversary becomes $O(2^3 \times 2^{K_1} \times log_2(K_2))$:

$$\begin{aligned} Q &= Rot^{-1}(B, S) \\ &= Rot(n_2, K_2 \oplus IDS) \end{aligned} \tag{9}$$

(f) It further increases the complexity $O(2^3 \times 2^{K_2} \times 2^{K_1} \times log_2(K_2))$ to take *XOR* between *IDS* and all possible values of K_2

$$R = K_2 \oplus IDS \tag{10}$$

(g) Then finally inner rotation is undone with the overall complexity $O(2^4 \times 2^{K_2} \times 2^{K_1} \times log_2(K_2))$

$$Rot^{-1}(Q,R) = n_2 \tag{11}$$

Similarly, adversary requires $O\left(2^{hw(P)} \times log_2(K)\right)$ complexity for computation of original value from its pseudo-Kasami code (K_c). For computation of ID, adversary requires $O\left(2^5 \times 2^{5\times hw(P)} \times log_2(K_2) \times log_2(K_1) \times log_2(n_1) \times 2^{n_1} \times 2^{K_1}\right)$ complexity. In KMAP, these pseudo-random numbers and keys are updated after each successful completion of protocol session. With the computational complexity as discussed above, it becomes practically impossible for an adversary to retrieve the conjecture secrets.

3. *Integrity* The messages B and C not only provide the evidence of authentication of the reader but also assure the integrity of the transmitted messages. For example, if an attacker tries to modify the random number, n_1, by altering the few bits of the message A, then the impact of this alteration directly transfers to the message B The tag will extract invalid random number, n_2, and this will lead to computation of invalid C (because of computation of invalid seed for pseudo-Kasami codes). The tag will not authenticate such reader and will terminate the protocol session. In KAMP it is impossible for an adversary to adjust the value of B to a correct value because $K_1 \oplus n_1$ (integrated in message B) provides the authenticity of the message A. Hence, the superlative composition of the messages in KMAP ensures the integrity of the each transmitted message.

4.2 Security Attacks/Cryptanalysis

1. *Desynchronization attack* In KMAP, both the reader and the tag maintain their synchronization with each other by rational updating of their shared secrets (K_1, K_2 & IDS) after each successful completion of authentication session. This synchronization mainly depends upon the correctness of the messages (C & D) which can only be generated by legitimate parties. Now, there are two possible approaches which can break the synchronization between the reader and the tag:

   (a) *Adversary disrupts message 'C'* Since, the tag does not receive message C sent by the reader, it will not authenticate the reader and eventually will not update its pseudonym and keys. The tag will also terminate the protocol session with the reader and will not compute message 'D'. Therefore, both the reader and tag will remain in the same state (synchronized).
   Another possible scenario can be an attempt of an adversary to modify message 'C' to make the reader and tag out of synchronization without being noticed. However, this is not possible in our proposed scheme because the authenticity and integrity of pseudorandom numbers (n_1, n_2) are ensured and confirmed before the computation of C and D messages which explicitly involves these random numbers. It means adversary can't make any change without being noticed.

   (b) *Adversary disrupts message 'D'* Since, the reader doesn't receive $'D'$ message sent by the tag, it will make the tag update its pseudonyms (IDS^{i+1}) and keys $\left(K_1^{i+1}, K_2^{i+1}\right)$, but the reader will remain in the previous state $\left(IDS^i, K_1^i, K_2^i\right)$. However, in KMAP, both the tag and reader store the two entries of keys and S $\left(IDS^i, K_1^i, K_2^i, IDS^{i-1}, K_1^{i-1}, K_2^{i-1}\right)$, therefore they can still authenticate each

other using previous values. The storage of two entries (old and new) of *IDS* and keys avoids the all possible types of desynchronization attacks.
Secondly, an adversary may try to make both the reader and the tag use different random numbers (n_1, n_2) by tampering the messages (A, B). But in our proposed scheme this tampering will be noticed by the tag, because little change in message A will lead to computation of entirely different n_2.

2. *Replay attack* The attacker may disrupt message $'D'$ in an ongoing genuine protocol session and replays the previously captured $'D'$ message towards the reader. However, the reader will not authenticate such message, because in each authentication session the reader generates different random numbers (n_1, n_2) and the computation of the $'D'$ message and pseudo-Kasami code (K_c) of the variables involve these random numbers.
   Another possibility is of an adversary trying to impersonate as a reader and replays the old message $A||B||C$ (corresponding to old *IDS*), but this sort of attacks will not change any of the internal secrets. The proposed scheme caters this type of the attacks by storing of two values of the local variables. Even replay attack proposed by Sun et al. [36] will not affect synchronization of the KMAP tags.
3. *Traceability attack* Traceability is also one of the important security threats that can affect the basic functionalities of the RFID systems. When the tag responds the reader with static values then it may cause traceability attack not only possible but non-trivial. In traceability attacks, the adversaries can identify and track the movements of the RFID tags.
   In our scheme, the tag uses its *IDS* instead of its original *ID* for interaction with the reader and *IDS* will be updated after each successful authentication session. Moreover, our proposed scheme involves random numbers (n_1, n_2) for update operation, therefore this makes tracking impossible for adversaries. The adversary can't even track the tag movements through exchanged messages (A, B, C, D) because each message involves random numbers (n_1, n_2) in their designs.
   Another possibility of tracking is of an adversary trying to disrupt the message 'C' and restrict the tag to use old pseudonyms and keys. However, in KMAP, a protocol session message counter has also been integrated within the tag which stops the functionality of the tag for some particular time if counter's value exceeds the threshold (≤ 8).
   Juels and Weis [46] proposed a formal definition of traceability and a structural model to validate the untraceibility claims of the protocols. Raphael reformulated the Juels traceability model to evaluate the UMAPs [47]. We have used the Raphael traceability model to analyse the proposed protocol. The description of the model is as follows.

Formal Traceability analysis In RFID systems, protocol parties $(\mathcal{P})$ tags $(\mathcal{T})$ and the readers $(\mathcal{R})$ communicates with each other. An Adversary $(\mathcal{A})$ can interact actively and passively and control the communication between all the parties. The Adversary $(\mathcal{A})$ can run the following queries:

Execute $(\mathcal{R}, \mathcal{T}, i)$ query: This query models the passive Adversary $(\mathcal{A})$. The Adversary $(\mathcal{A})$ can eavesdrop the communication channel and can record (obtain) all transmitted messages in protocol session i between the reader and the tag.
Send $(\mathcal{P}_1, \mathcal{P}_2, i, m)$ query: This query models the active Adversary $(\mathcal{A})$. The Adversary $(\mathcal{A})$ can impersonate either as $\mathcal{P}_1(\mathcal{P}_1 = \mathcal{T})$ or $\mathcal{P}_2(\mathcal{P}_2 = \mathcal{R})$ and send message m in protocol session i to its contended party $(\mathcal{P}_1 or \mathcal{P}_2)$.

Corrupt $(\mathcal{T},\ S^{'})$ query: This query allows the Adversary $(\mathcal{A})$ to obtain the concealed secret S of the tags and then set $S = S^{'}$. Since the RFID tags generally are not tamper–resistant, so this query assumes that the Adversary $(\mathcal{A})$ has physical access to the tag. *Test* $(i, \mathcal{T}_0, \mathcal{T}_1)$ query: This query mainly defines the untraceibility (UNT) and doesn't correspond to any Adversary $(\mathcal{A})$ capabilities. In the protocol session i, this query gives Adversary $(\mathcal{A}), ID_b$ from the set (ID_0, ID_1) corresponding to the tags $(\mathcal{T}_0, \mathcal{T}_1)$ where $b \in \{0, 1\}$. The Adversary $(\mathcal{A})$ succeeds if it can guess the correct value of b.

The untraceibility (UNT) problem can be defined as game $(\mathcal{G})$. The game $(\mathcal{G})$ is played between Adversary $(\mathcal{A})$ and protocol parties $(\mathcal{P})$. The game $(\mathcal{G})$ involves following three phases.

Phase 1 (learning)	The Adversary $(\mathcal{A})$ can send any number of Execute, Send and Corrupt queries.
Phase 2 (challenge)	During game $(\mathcal{G})$, the Adversary $(\mathcal{A})$ runs the test query on two fresh tags $(\mathcal{T}_0, \mathcal{T}_1)$. The Adversary $(\mathcal{A})$ is then given the challenge identifier ID_b from the set (ID_0, ID_1) where $b \in \{0, 1\}$. The Adversary $(\mathcal{A})$ continuously sends Execute, Send and Corrupt queries with restriction that the tags $(\mathcal{T}_0, \mathcal{T}_1)$ are not issued any Corrupt query.
Phase 3 (guessing)	Finally, the Adversary $(\mathcal{A})$ terminates the game $(\mathcal{G})$ and outputs a conjecture bit $b^{'}$ of the value b.

The KMAP protocol proves to be untraceable against the above mentioned formal traceability model [47]. The adversary can send multiple queries to learn the relation between *ID* and the publically transmitted messages. An adversary computes the equation with probability p_r in the form of:

$$[ID]_i = [N_1]_i \blacksquare [N_2]_i \ldots \blacksquare [N_l]_i \tag{12}$$

where N_j is the publically disclosed message and $\blacksquare$ denotes the logical operation between the messages. If probability $p_r > \frac{1}{2}$, the adversary wins the game and the protocol will not resist traceability.

The result of the game $(\mathcal{G})$ is based on the following equation:

$$\mathit{Adversary}_{\mathcal{A}}^{UNT}(k) = |\Pr[\mathcal{A}\ \mathit{wins}] - \Pr[\mathit{random\ coin\ flip}]| = \left| p_r - \frac{1}{2} \right| > \varepsilon(K)$$

The main reason which allows the adversary to compute Eq. 12, is the use of only unbalanced operators in protocol designs. By using the three phases of model, the adversary can compute the following ambiguous equation for the KMAP:

$$Rot^{-1}\big(Rot^{-1}(D, K_c(K_2)) \oplus K_c(K_1), K_c(IDS)\big) = K_c(ID) \oplus K_c(n_1) \tag{13}$$

All the variables used in Eq. 13 only occur in message D and all of the operators in protocol designs are balanced. It is practically impossible for an adversary to compute the pseudo-Kasami code of any variable. For each new authentication session, the adversary will encounter new pseudorandom numbers and hence new pseudo Kasami codes. Merging of these variables with other messages will make the equation more complex and ambiguous. Thus the adversary $(\mathcal{A})$ will not able to identify the challenge identifier and hence

$$\mathcal{A}dversary_{\mathcal{A}}^{UNT}(k) = |\Pr[\mathcal{A}\ wins] - \Pr[random\ coin\ flip]| = \left|\frac{1}{2} - \frac{1}{2}\right| = 0 < \varepsilon(K).$$

Therefore, KMAP protocol provides the optimal untraceibility and anonymity to its associated tags.

4. *Adhoc and probabilistic full disclosure attack* In KMAP, the protocol messages are composed of non-triangular primitives such as pseudo-Kasami codes (K_c) and Rotation (Rot) instead of simple *T-functions* (XOR, OR, AND). The inclusion of simple *T-functions* for composition of the protocol messages inroads the malicious activities which eventually leads towards probabilistic and Adhoc based full disclosure of secrets. In KMAP, new non-triangular primitive pseudo-Kasami codes (K_c) have been proposed and extensively used in the protocol messages which not only enhance the statistical properties (randomness) of the protocol messages but also amplify their diffusion properties. In most of the probabilistic full disclosure attacks, the attacker takes *XOR*between various combinations of the protocol messages and finds the suitable/optimal approximations of the secrets. However, in KMAP, use of pseudo-Kasami codes increases the overall computational complexity of conjecture approximation, hence there can be many pairs that yield similar results.
5. *Formal Structural cryptanalysis* Most of the disclosure attacks proposed on UMAP protocols are protocol specific and are not extendible to the broader class of UMAPs. Only four formal cryptanalysis models: Tango, Recursive Linear Cryptanalysis (RLC), Recursive Differential Cryptanalysis (RDC) and Casper & FDR Tools exist for security analysis of the UMAPs. The detailed working of these formal security analysis models is presented as follows:

   (a) *Tango attack* In 2009, Hernandez-Castro et al. [21] presented the first structural cryptanalysis framework (Tango attack) for security analysis of the UMAPs. Initially, the attack targeted David–Prasad protocol but later Hernandez-Castro et al. extended the attack for other UMAPs as well. The attack following comprises two phases:

      (i) *Selection of Good Approximations (GAs)* The tango attack mainly exploits the inherent poor diffusion properties of *T-functions* [1]. In this phase, the attacker constitutes the Good Approximations (GAs) of the secrets (K, ID) using multiple combinations of the exchanged messages. Among all possible GAs, the adversary shortlists those GAs for its cryptanalysis which are systematically closer to the target secret. The adversary makes this decision on the basis of hamming distance between approximation and the secret. According to EPC-Class-1, Generation-2 [33] tags, all the variables are of 96 bits, so any GA will get selected if $hd(GA, Secret) < 48$ (where hd is hamming distance).

      (ii) *Combination and comparison of GA* In the second phase, the adversary combines the multiple GAs for a particular secret (obtained in various protocol sessions) and computes a GAs based matrix. The adversary further compares the column wise number of 1's (A) with precomputed threshold (γ) for final conjecture secret; where threshold function of $th(A)$ can be computed as follows:

$$th(A) = \begin{cases} if \ A_i \geq \gamma & assign\ 1 \\ if \ A_i \leq \gamma & assign\ 0 \end{cases}$$

where $\gamma = 0.5 \times N_A \times N_S$, N_A = Number of GA for the secrets, N_S = Number of eavesdropped sessions

However, the tango attack fails to find the optimal conjecture secrets for UMAPs which incorporate non-triangular functions such as Rot, Recursive Hash(R_h)[6], Reconstruction [29] and pseudo Kasami codes (K_c) etc. This limitation of the tango attack has also been highlighted by Hernandez-Castro et al. in [21]. In KMAP, we have extensively used non-triangular functions *Rot* and Pseudo Kasami codes (K_c) for the composition of protocol messages. The inclusion of the pseudo Kasami codes (K_c) in protocol messages increases the overall computational complexity for retrieval of conjecture tag's secret *ID* to $O\left(2^5 \times 2^{5\times hw(P)} \times log_2(K_2) \times log_2(K_1) \times log_2(n_1) \times 2^{n_1} \times 2^{K_1}\right)$.The hamming distance (hd) between A and $K_c(A)$ also varies according to the seed value $(Seed = hw(P)\ mod\ K)$ and hence makes hamming weight unpredictable. Therefore, it is impossible to find the optimal approximations of the secrets using nonlinear (twice left rotated and pseudo-Kasami encoded) messages. Hence, the KMAP is highly resistive to the tango attack.

(b) *Recursive Linear Cryptanalysis (RLC)* RLC [20] also exploits the weak inherent properties of the T-functions for its execution. RLC involves following three steps:

(i) Adversary first determines the unknown secret variables transmitted within single protocol session.
(ii) Creates a system of linear equations for each bit of the secret variable.
(iii) Start solving these equations recursively, starting from the Least Significant Bit (LSB) to retrieve the concealed secrets.

RLC can recover all the secret bits one by one with probability one, provided that the adversary is able to compute optimal number of linear equations. RLC is passive, deterministic and requires only one protocol session for its execution. However, RLC fails to retrieve secrets of those UMAP protocols which incorporate nonlinear (non-triangular) functions in their designs. The inclusion of the nonlinear primitives creates hindrance in computation of linear set of equations and hence fails this linear cryptanalysis. Since, the KMAP abundantly uses nontriangular function (pseudo-Kasami codes) therefore it avoids RLC optimally.

(c) *Recursive differential cryptanalysis (RDC)* Recursive Differential Cryptanalysis (RDC) [20] is more powerful attack than RLC and can be applied to all those protocols for which RLC fails to compute secrets. RDC is probabilistic attack and requires more than one authentication session for its execution. RDC blocks the last confirmation message and restricts both the reader and the tag to use previous *Keys* and *IDS*. Now, for each new authentication session all the dynamic secrets remain same except random numbers $(n_1 \text{and} n_2)$. In KMAP, RDC will fail to find the differential relationship between these nonces (because of double rotation) as the seed of the pseudo-Kasami codes is independent of the protocol execution therefore blocking of the update confirmation messages will not affect computation of the pseudo-Kasami codes. Hence, RDC also fails to construct linear equations for computation of the KMAP secret variables. Inventors of the RLC and RDC have also highlighted this weakness of both methods [20].

(d) *Casper and FDR Tools* Casper and FDR tools perform automatic formal security analysis. Firstly, the security protocol is described in abstract language understandable [48, 49] where Casper further produces the Communicating Sequential Processes (CSP) description of the same protocol. Then Failure Divergence Refinement (FDR) is used to validate the CSP description of the protocol. FDR uses the assumptions of Dolev–Yao [50] security model to find the attacks in the protocols or show that no attack exists. The abstract language description encompasses all the free variables, processes, specifications, functions and intruder information for proper formal analysis. The testing performed for the desired specification is as follows.

- *Secret* $(\mathcal{T}, [K_1, K_2], \mathcal{R})$: The tag and the reader share two secret session keys $[K_1, K_2]$ with each other.
- *Agreement* $(\mathcal{R}, \mathcal{T}, [ID, K_1, K_2])$: The reader is successfully authenticated by the tag after successful verification of the message C and both parties $(\mathcal{P})$ agree on the values of the secret $ID$and keys(K_1, K_2).
- *Agreement* $(\mathcal{R}, \mathcal{T}, [ID, K_1, K_2])$: The tag is authenticated by the reader, after successful verification of the message D and both parties $(\mathcal{P})$ agree on the values of the secret $ID$and keys(K_1, K_2).

Since FDR2 passes these specifications without any attack, the KMAP protocol is verified to achieve the desired functionalities.

5 Performance Evaluation

This section evaluates the performance of the KMAP in terms of computational operational cost, memory (storage) requirement, tag communication cost and security. The computational operations mainly focus on the tag's logical operations (operators used).

The KMAP involves only two simple bitwise logical operations: XOR and Rot. Another ultralightweight primitive (pseudo-Kasami code) has also been used in protocol design which in fact composed of XOR and Rot functions. Hence all of these operations used in KMAP are extremely lightweight in nature and fall well within the ultralightweight class.

Regarding memory (storage) requirements, each tag owns one static ID and two entries of IDS and keys (old and new). So, a ROM of $1L$ (96 bits) is required to store static ID and $6L$ (576 bits), rewritable memory is required for storage of old and newly updated variables; keys (K_1, K_2) and IDS A 3-bit message counter is also required to overcome the possible DoS attacks.

As far as communication cost is concerned, we only need to count the messages sent by the tag in one protocol session.

Here in KMAP, the tag transmits two messages (192 bits) together hence the overall communication cost is $2L$. The KMAP provides the optimal security as compared to its contended protocols of UMAP family [2, 4–6, 12, 16–19, 24, 27, 29–33]. None of these protocols completely satisfies the security model presented in Sect. 4 and even fails to avoid DoS attack [3]. However, as discussed in Sect. 4, KMAP can endure and robust against all types of cryptanalysis attacks mentioned in security model. Table 3 illustrates the performance comparison of our proposed protocol with other UMAP family protocols.

Table 3 Performance analysis of various UMAP family protocols (tag side)

	LMAP [4]	SASI [12]	David–Prasad [17]	Yeh et al. [18]	GOASSMER [16]	RAPP [24]	KMAP (proposed)
Computational operation on tag	$\oplus$, *OR*, +	$\oplus$, *OR*, +, *AND*, *Rot*	$\oplus$,*AND*	*Not*, $\oplus$, *OR*, *Rot*	$\oplus$, +, *Rot*, *MixBits*	$\oplus \cdot$ *Rot*, *Per*	$\oplus$, *Rot*
Memory requirement on tag	6*L**	7*L*	5L	3*L*	7*L*	5*L*	7*L*
Communication messages generated by tag	2*L*	2*L*	3*L*	2*L*	2*L*	2*L*	2*L*
Total number of messages for mutual authentication	4*L*	4*L*	5*L*	2*L*	4*L*	5*L*	4*L*
Resistance to desynchronization attacks	No	No	No	No	No	No	Yes
Resistance to full disclosure attacks	No	No	No	No	Yes	No	Yes
Resistance to traceability attacks	No	No	No	No	No	No	Yes
Resistance to formal security attacks	No	No	No	No	No	No	Yes

* Length of the string

6 Conclusion

In this paper, we have proposed pseudo-Kasami-code based ultralightweight RFID Mutual Authentication Protocol (KMAP). The computation of this newly proposed primitive (pseudo-Kasami-code) is extremely lightweight as it involves only circular left rotation and bit wise *XOR* operations. The pseudo-Kasami-code (K_c) has two very important properties: hamming weight unpredictability (for original variable) and irreversibility, which make RFID systems more secure and robust against all possible full disclosure attacks. Since most of full disclosure attacks are based on reverse engineering and hamming weight based estimation of conjecture secrets, so the use of pseudo-Kasami-code (K_c) increases the non-linearity in protocol messages and avoids all possible reverse engineering scenarios. The security and performance analysis shows the optimal compatibility of the proposed scheme with extremely low cost passive RFID tags.

References

1. Klimov, A., & Shamir, A. (2005). New applications of T-functions in block ciphers and hash functions. In *Proceedings of FSE'05*, *volume 3557 of LNCS* (pp. 18–31). Springer.
2. Mujahid, U., Najam-ul-Islam, M., & Ali Shami, M. (2015). RCIA: A new ultralightweight RFID authentication protocol using recursive hash. International Journal of Distributed Sensor Networks 2015, Article ID 642180. doi:10.1155/2015/642180.
3. Mujahid, U., & Najam-ul-Islam, M. (2015). Pitfalls in ultralightweight RFID authentication protocol. *International Journal of Communication networks and information Security, 7*(3), 169.
4. Peris-Lopez, P. Hernandez-Castro, J., et.al (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Proceedings of 2nd workshop on RFID security* (pp. 100–112), Austria.
5. Peris-Lopez, P., Hernandez, J. C., et.al (2006). EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In *1st International workshop on information security (OTM-2006)* (pp. 352–361), France.
6. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (20016). M2AP: A minimalist mutual-authentication protocol for low cost RFID tags. In *Proceedings of 2006 international conference on ubiquitous intelligence and computing* (pp. 912–923).
7. Marsaglia, G., & Tsang, W. W. (2002). Some difficult-to-pass tests of randomness. *Journal of Statistical Software, 7*(3), 37–51.
8. Walker, J. (1998). ENT randomness test. http://www.fourmilab.ch/random/.
9. Suresh, C., Charanjit, J., Rao, J. R., Rohatgi, P. (1999). A cautionary note regarding evaluation of AES candidates on smart-cards. In *2nd Advanced Encryption Standard (AES) candidate conference*. http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm.
10. Li, T., et.al. (2007). Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol. In 2nd *International conference on availability, reliability and security*, *2007*. ARES.
11. Li, T., & Wang, G. (2008). Security analysis of family of ultra-lightweight RFID authentication protocols. *Journal of Software, 3*(3), 1–10.
12. Chien, H.-Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transaction on Dependable and Secure Computing, 4*(4), 337–340.
13. Avoine, G., Carpent, X., & Martin, B. (2010). *Strong authentication and strong integrity (SASI) is not that strong* (pp. 50–64). Turkey: Workshop on RFID Security and Privacy.
14. Jeon, S., & Yoon, E.-J. (2013). Cryptanalysis and Improvement of a new ultra-lightweight RFID authentication protocol with permutation. *Applied Mathematical Sciences, 7*(69), 3433–3444.
15. Hernandez, J. C., et al (2008). Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations. In *ArXiv, Cryptography and Security, Report 0811.4257*. http://arxiv.org/abs/0811.4257.
16. Peris-Lopez, P., et al. (2009). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In *9th International workshop on information security applications* (pp. 56–68).

17. David, M., & Prasad, N. R. (2009) Providing strong security and high privacy in low-cost RFID networks. In *International conference on Security and privacy in mobile information and communication systems* (pp. 172–179), Italy.
18. Yeh, K.-H. et.al. (2010) An efficient ultralightweight authentication protocol for RFID systems. In *Workshop on RFID security and privacy* (pp. 49–60), Turkey.
19. Lee, Y. C., Hsieh, Y. C., You, P. S., Chen, T. C. (2009). A new ultralightweight RFID authentication protocol with mutual authentication. In *Proceeding of the WASE international conference on information engineering*, vol. 2 IEEE Computer Society.
20. Ahmadian, Z., Salmasizadeh, M., et al. (2013). Recursive linear and differential cryptanalysis of ultralightweight authentication protocols. *IEEE Transactions on Information Forensics and Security, 8*(7), 1140–1151.
21. Hernandez-Castro, J. C., et al (2010). Cryptanalysis of the David–Prasad RFID ultralightweight authentication protocol. In *Workshop on RFID Security and Privacy* (pp. 22–34), Turkey.
22. Bilal, Z., Masood, A., & Kausar, F. (2009). Security analysis of ultra-lightweight cryptographic protocol for low-cost RFID tags: Gossamer protocol. In *12th International conference on network-based information systems* (pp. 260–267), Indianapolis, USA.
23. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & van der Lubbe, J. C. A. (2009). Security flaws in a recent ultralightweight RFID protocol. arXiv preprint arXiv:0910.2115.
24. Tian, Y., Chen, G., & Li, J. (2012). A new ultralightweight RFID authentication protocol with permutation. *IEEE Communications Letters, 16*(5), 702–705.
25. Shao-hui, W., et al (2012). Security analysis of RAPP: An RFID authentication protocol based on permutation. *Cryptology ePrint Archive*. Report 2012/327. https://eprint.iacr.org/2012/327.
26. Ahmadian, Z., Salmasizadeh, M., & Aref, M. R. (2013). Desynchronization attack on RAPP ultralightweight authentication protocol. *Information Processing Letters, 113*(7), 205–209.
27. Jeon, S., et al. (2013). A new ultra-lightweight RFID authentication protocol using merge and separation operations. *International Journal of Mathematical Analysis, 7*(52), 2583–2593.
28. Zhuang, X., Wang, Z. H., Chang, C. C., & Zhu, Y. (2013). Security analysis of a new ultra-lightweight RFID protocol and its improvement. *Journal of Information Hiding and Multimedia Signal Processing, 4*, 165–180.
29. Zhuang, X., Zhu, Y., & Chang, C.-C. (2014). A new ultralightweight RFID protocol for low-cost tags: R ^{2}AP. *Wireless Personal Communications, 79*(3), 1787–1802.
30. Engels, D., et al. (2010). Hummingbird: ultra-lightweight cryptography for resource-constrained devices. In *14th International conference on financial cryptography and data security* (pp. 3–18), Spain.
31. Song, B., & Mitchell, C. J. (20008). RFID authentication protocol for low-cost tags. In *1st ACM conference on wireless network security, USA* (pp. 140–147).
32. Özcanhan, M. H., et al. (2015). Mersenne twister-based RFID authentication protocol. *Turkish Journal of Electrical Engineering and Computer Sciences*. doi:10.3906/elk-1212-95.
33. Bilal, Z, & Martin, K. (2013). Ultra-lightweight mutual authentication protocols: Weaknesses and countermeasures. In *International conference on availability, reliability and security* (pp. 304–309).
34. Rizomiliotis, P., et al. (2009). Security analysis of the Song-Mitchell authentication protocol for low-cost RFID tags. *IEEE Communications Letters, 13*(4), 274–276.
35. Bilal, Z., Martin, K., & Saeed, Q. (2015). Multiple Attacks on authentication protocols for low-cost rfid tags. *Applied Mathematics and Information Sciences, 9*(2), 561–569.
36. Sun, H.-M., Ting, W.-C., & Wang, K.-H. (2011). On the security of Chien's ultralightweight RFID authentication protocol. *IEEE Transactions on Dependable and Secure Computing, 8*(2), 315–317.
37. Mujahid, U., & Najam-ul-islam, M. (2014). Ultralightweight cryptography for passive RFID systems. *International Journal of Communication Networks and Information Security., 6*(3), 173–181.
38. Avoine, G., Carpent, X., & Martin, B. (2012). Privacy-friendly synchronized ultralightweight authentication protocols in the storm. *Journal of Network and Computer Applications, 35*(2), 826–843.
39. Barrero, D. F., et al. (2014). A genetic tango attack against the David–Prasad RFID ultra-lightweight authentication protocol. *Expert Systems (Journal), 31*(1), 9–19.
40. Peris-Lopez, P., et.al (2011). Quasi-linear cryptanalysis of a secure RFID ultralightweight authentication protocol. In 6th *International conference on information security and cryptology* (pp. 427–442), China.
41. Han, D. (2011). Gröbner basis attacks on lightweight RFID authentication protocols. *Journal of Information Processing Systems, 7*(4), 691–706.
42. Li, T., & Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In *International information security conference* (SEC) (pp. 109–120), South Africa.
43. D'Arco, P., & De Santis, A. (2011). On ultralightweight RFID authentication protocols. *IEEE Transactions on Dependable and Secure Computing, 8*(4), 548–563.

44. Zubair, M., Mujahid, U., Najam-ul-Islam, M., & Ahmed, J. (2012). Cryptanalysis of RFID ultralightweight protocols and comparison between its solutions approaches. *BUJICT Journal, 5*(1), 58–63.
45. Khovratovich, D., et al (2010). Rotational cryptanalysis of ARX. In *17th international conference on fast software encryption (FSE-2010)* (pp. 333–346).
46. Juels, A., & Weis, S., (2005). Authenticating pervasive devices with human protocols. In *25th International cryptology conference, santa barbara* (pp. 293–308).
47. Phan, R. C.-W. (2009). Cryptanalysis of a new ultralightweight RFID authentication protocol-SASI. *IEEE Transactions on Dependable and Secure Computing, 6*(4), 316.
48. M. Aiash., Mapp, G., Phan, R., Lasebae, A., & Loo, J. (2012). A formally verified device authentication protocol using Casper/FDR. In *11th International conference on trust, security and privacy in computing and communications (TrustCom)* (pp. 1293–1298).
49. Lowe, G. (1998). Casper: A compiler for the analysis of security protocols. *Journal of Computer Security, 6*(1), 53–84.
50. Dolev, D., & Yao, A. C. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory, 29*(2), 198–208.

Umar Mujahid is working as Assistant Professor in Electrical Engineering Department, Bahria University Islamabad. He has done his Ph.D. in information Security from Bahria University, Islamabad. Mr. Umar Mujahid secured first position in MS (Telecommunication Engineering) and was awarded with Gold Medal. Currently his research interests are Ultralightweight Cryptography for passive RFID systems, Efficient hardware prototyping of cryptographic processors, formal security analysis and cryptanalysis. He has published more than 32 research papers in various International Journals and conferences.

Muhammad Najam ul Islam is working as Professor & Dean in the Faculty of Engineering & Sciences at Bahria University, Islamabad. His current research interests include Information Security, Flexible Radios and Renewable Energy. He has published more than 40 papers in reputed international journals and conferences. Dr. Najam did his Ph.D. in Electrical Engineering from Telecom ParisTech, France. During his doctoral studies, he was working at EURECOM, Sophia Antipolis on the research theme of flexible radios for multi-standard wireless communication devices. Before moving to France, Najam completed his Masters in Computer Sciences and Bachelors in Electrical Engineering from LUMS, and UET, Lahore, Pakistan respectively. Dr. Najam has also worked for Govt. of Pakistan in R&D projects from 1998 to 2004.

Shahzad Sarwar is currently working as Assistant Professor in Punjab University College of Information Technology, Lahore, Pakistan. He did his Ph.D. in Electrical Engineering and Information Technology Vienna University of Technology, Austria in 2008. Dr. Shahzad is involved in research projects related to computer networks, information security and integration of open source software in IT education.