Hindawi
Security and Communication Networks
Volume 2019, Article ID 3295616, 25 pages
https://doi.org/10.1155/2019/3295616

Review Article

Ultralightweight RFID Authentication Protocols for Low-Cost Passive RFID Tags

Madiha Khalid,[1] **Umar Mujahid**,[1,2] **and Najam-ul-Islam Muhammad**[1]

[1]*Department of Electrical Engineering, Bahria University, Islamabad, Pakistan*
[2]*Department of Information Technology, Georgia Gwinnett College, Georgia, USA*

Correspondence should be addressed to Madiha Khalid; madihazoheb.buic@bahria.edu.pk

Received 2 March 2019; Revised 26 May 2019; Accepted 23 June 2019; Published 21 July 2019

Academic Editor: Petros Nicopolitidis

The field of pervasive computing especially the Internet of Things (IoT) network is evolving due to high network speed and increased capacity offered by the 5G communication system. The IoT network identifies each device before giving it access to the network. The RFID system is one of the most prominent enabling technologies for the node identification. Since the communication between the node and the network takes place over an insecure wireless channel, an authentication mechanism is required to avoid the malicious devices from entering the network. This paper presents a brief survey on the authentication protocols along with the prominent cryptanalysis models for the EPC C1G2 RFID systems. A comparative analysis is provided to highlight the common weaknesses of the existing authentication algorithms and to emphasize on the lack of security standardization for the resource constraint IoT network perception layer. This paper is concluded by proposing an ultralightweight protocol that provides Extremely Good Privacy (EGP). The proposed EGP protocol avoids all the pitfalls highlighted by the cryptanalysis of the existing authentication protocols. The incorporation of the novel ultralightweight primitives, Per-XOR (P_x) and Inverse Per-XOR (P_x^{-1}), makes the protocol messages more robust and irreversible for all types of adversaries. A comprehensive security analysis illustrates that the proposed protocol proves to be highly resistive against all possible attack scenarios and ensures the security optimally.

1. Introduction

The concept of creating low-cost, reliable, and secure Internet of Things (IoT) networks for current and future applications is evolving by the virtue of high network speed and increased capacity offered by the 5*th* generation communication system. The IoT network consists of interrelated computing devices with unique identification, deployed in the environment to collect, process, and share the information, in order to facilitate the measurement of changes in the surroundings and to react independently primarily without human interaction [1–4]. The data collected by the network is also processed to generate valuable information that can be used to enhance the user experience in future [5]. The IoT platform is being used in various fields to achieve purposeful objectives such as logistics [6], smart cities [7], and supply chain management [8].

The IoT network initially identifies the electronic devices (nodes) before giving them access to the network. The Radio Frequency Identification (RFID) system is emerging as an enabling technology for the node discovery due to the features such as high speed, long range, and nonline of sight scanning [4]. The RFID enabled IoT networks are being preferred in various surveillance, monitoring, and healthcare applications. Table 1 highlights some of the prominent applications reported in the literature.

The architecture of the RFID enabled IoT network is composed of three components: the RFID system, the IoT middleware, and the Internet [15]. The RFID system facilitates the node identification and the data collection. The data gathered from the environment under observation is processed by the IoT middleware. The IoT middleware also acts as a gateway to the external Internet [16].

The architecture of the RFID system embedded in an IoT network consists of three main components; the Electronic Product Code (EPC) tag, the reader, and the database. The tag is a low-cost electronic chip with the unique identification number (*ID*). The reader identifies each tag associated with

Table 1: RFID enabled IoT applications.

System	Function
Smart Home Mobile RFID based IoT system [9]	This is a smart home service system to benefit the user in terms of cost, energy consumption and ease.
RFID & IoT for attendance monitoring system [10]	This is a real time attendance monitoring system that can be accessed by various parties, i.e., teachers, students and parents.
Harvard hybrid system [11, 12]	The system uses the RFID tags to track equipments, beds, patients and ICU babies.
Positive patient identification system [13]	The system facilitates the patient identification and speed up access to the patient's data.
Intel transfusion system [14]	This system identifies blood bags, recipients and staff. The purpose of this system is to enhance the safety of the blood transfusion.

the system by receiving the *ID* over the wireless channel. The database supports the reader in an identification process by storing attributes of all the tags affiliated with the RFID system [17].

The EPC standards have segregated the tags into classes based on their functionality. The description of each EPC class is given in Table 2 [18].

In the RFID enabled IoT networks, the node is identified by communicating the tag's *ID* to the reader over an insecure wireless channel. Therefore, the system is prone to many security and privacy threats [19]. A mutual authentication mechanism is an inevitable part of the tag identification process. In this paper, a brief survey on the existing mutual authentication protocols and the prominent cryptanalysis models for the EPC Class 1 Generation 2 (C1G2) RFID systems is presented. A comparative security analysis among the prominent protocols has been drawn to highlight some of the common weaknesses of the existing authentication algorithms for the resource constraint RFID systems. The paper also proposes the Extremely Good Privacy (EGP) protocol. The comprehensive security analysis of the EGP protocol ensures its security claims and robustness against all existing cryptanalysis models. The EPC C1G2 tags are the key component of the low-cost RFID systems due to characteristics like small size, low cost, and unlimited lifespan [20]. Other features of the EPC C1G2 identification system are enumerated as follows [18]:

(i) Operating frequency: 860 MHz-960 MHz

(ii) Memory capacity: 96-256 bits

(iii) Field programmable

(iv) Reprogrammable

(v) Communication: 640Kbits/s

(vi) Reads: 1700 tags/sec

The rest of the paper is organized as follows: Section 2 discusses the Ultralightweight Mutual Authentication Protocols (UMAPs) for the resource constraint RFID systems followed by Section 3 that describes multiple cryptanalysis models used for the security and privacy evaluation of the UMAPs. This section also presents a comparative analysis of the prominent UMAPs based on their strengths to provide Confidentiality, Integrity, Availability, and Authentication (CIAA) services. The EGP protocol is proposed in Section 4 along with the detailed cryptanalysis report. Finally, the paper is concluded in Section 5.

2. Ultralightweight Mutual Authentication Protocols

The node authentication mechanism during the identification process prevents the malicious users from entering the network through the perception layer. In 2007, Chien [21] divided the authentication protocols in four categories which are defined as follows:

(1) Heavyweight: these protocols incorporate the classical cryptographic suits such as hash functions and private and public key cryptography.

(2) Middleweight: this category includes the protocols that can support one-way hash functions and pseudorandom-number generators only.

(3) Lightweight: these protocols can support the lightweight functions such as Cyclic Redundancy Checks (CRCs) and lightweight pseudorandom number generators.

(4) Ultralightweight: this class allows the incorporation of simple bitwise logical function only, for the protocol design.

Table 3 presents a relationship among the protocol categorization and the EPC classes supported by some prominent examples.

For low-cost systems, the silicon-based area of the EPC tags should be kept minimum to reduce the cost. Typically, an EPC C1G2 tag consists of $32Kbits$ response buffer [32] and can support maximum $4K$ Gate Equivalent (GE) for the crypto based operations. One gate equivalent corresponds to the area required for the fabrication of two input NAND gate [33]. Hence smaller GE for the authentication protocol implementation corresponds to the lesser cost overhead associated with the security-based operations.

Table 2: EPC classification of RFID tags.

Class	Description
Class 5	Class 5 tags are essentially active readers. They have the ability to communicate with all the EPC standard classes.
Class 4	Class 4 tags are active in nature. They can communicate with the reader and other tags by using peer to peer communication model.
Class 3	Class 3 tags are semi passive tags that can support broadband communication.
Class 2	Class 2 tags are the passive tags with extended functionality such as memory and computational resources.
Class 1/0	Class 1/0 tags are basic passive identity tags with limited resources.

Table 3: Mutual authentication protocol classification.

Protocol Classification	EPC Class association	Examples
Heavyweight	Class 5/4	Godor and Imre [22], Liu et al [23]
Middleweight	Class 3	Wang et al [24], Chou [25], Zhang and Qi [26]
Lightweight	Class 2	Lee et al [27], Liao et al [28]
Ultralightweight	Class 1/0	Tewari and Gupta [29], SLAP [30], KMAP [31]

Table 3 suggests that for C1G2 tags implementation of the UMAP is the only cost-effective option for the node verification at the identification stage. Numerous UMAPs have been presented over the last decade. This section describes the general structure of the UMAPs along with a brief survey of the existing protocols. Since 2006, more than thousand protocols have been proposed; however the basic working principle of these protocols remains the same. The UMAPs ensure that both the entities, i.e., the tag and the reader, are authentic components of an identification system with the help of a static and unique *ID* along with the pseudoidentification number and the keys (IDS, K) which are dynamic in nature. The dynamic variables update their status on both sides after every successful authentication session whereas the static *ID* remains constant. The mutual authentication process mainly consists of four steps which are as follows [34]:

(1) Tag identification: the tag receives a request for the latest identity pseudonym *IDS* after entering the communication range of the reader. The reader identifies the tag by retrieving the associated identification number and the keys from the database with the help of *IDS*.

(2) Reader authentication: after the tag authentication, the reader generates a private key for the authentication session and transmits message X to the tag. The message X consists of an encrypted version of the private key and the reader authentication challenge message. The reader's identity is verified if the response calculated at the tag's side is equal to the received challenge message.

(3) Tag authentication: the successful reader identity verification leads to the calculation and the transmission of the tag authentication challenge message Y for the valid reader.

(4) Dynamic variable update: the mutual authentication of communicating parties is followed by the dynamic variable updating process on both sides.

The block diagram of the generalized UMAP is presented Figure 1. The features that differentiate the UMAPs are the tag's memory architecture and the protocol's primitives. The UMAPs can be classified into three categories based on the nature of the operators used for the calculation of challenge/response messages. Description of each category along with the examples of prominent protocols is as follows.

2.1. UMAPs with Triangular Functions. In 2006, Peris-Lopez [35–37] laid the foundation of the ultralightweight cryptography. The main idea was to use the triangular functions such as bitwise *AND*, *OR*, *XOR*, and *modular addition* for the encryption of public messages which are being communicated among the resource constraint devices. The prominent UMAPs with triangular functions are Lightweight Mutual Authentication Protocol (LMAP) [35], Minimalistic Mutual Authentication Protocol (M^2AP) [36], and Efficient Mutual Authentication Protocol (EMAP) [37].

2.1.1. Lightweight Mutual Authentication Protocol (LMAP). The LMAP laid the foundation of UMAPs and falls under the umbrella of the triangular UMAPs. The memory architecture of the tag and the reader implementing the LMAP is given in Table 4. The protocol executes in following steps:

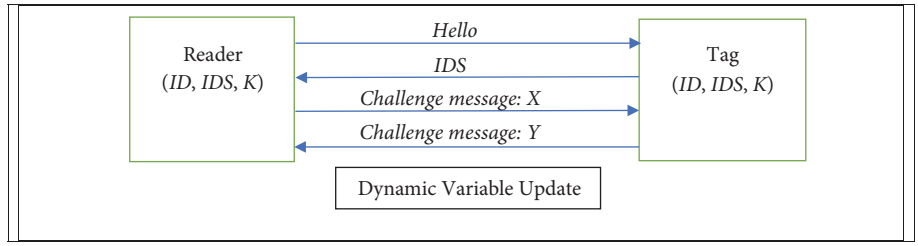


Figure 1: Flow diagram of generalized UMAP.

(1) The reader sends the "*Hello*" message to the tag.

(2) The tag replies with *IDS* to the reader. This *IDS* acts as an index in the database to locate the *keys* and the *ID* related to the tag. If the required data is not found, the protocol is terminated; otherwise it moves to next step.

(3) In step (3), the reader generates two pseudorandom numbers n_1 and n_2. These random numbers are used for the calculation of the messages A, B and C.

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

$$B = (IDS \vee K_2) \oplus n_1 \quad (2)$$

$$C = IDS + K_2 + n_2 \quad (3)$$

Finally, $X = A \parallel B \parallel C$ is transmitted to the tag.

(4) The tag extracts n_1 and n_2 from the messages A and C, respectively. The message B is a challenge token for the reader authentication. After successful reader authentication, the protocol moves to step (5).

(5) The tag generates and transmits message $D = Y$. The message D has two purposes: (a) concealed transfer of the tag's *ID*; (b) the tag authentication.

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

After the transmission of the message D, the dynamic variables at the tag's end are updated using following equations:

$$IDS^{NEW} = (IDS + (n_2 \oplus K_4)) \oplus ID \quad (5)$$

$$K_1^{NEW} = K_1 \oplus n_2 \oplus (K_3 + ID) \quad (6)$$

$$K_2^{NEW} = K_2 \oplus n_2 \oplus (K_4 + ID) \quad (7)$$

$$K_3^{NEW} = K_3 \oplus n_1 \oplus (K_1 + ID) \quad (8)$$

$$K_4^{NEW} = K_4 \oplus n_1 \oplus (K_2 + ID) \quad (9)$$

(6) The reader receives the message D, authenticates the tag, and updates the dynamic variables using (5)-(9). The process of updating dynamic variables on the reader's side only takes place in case of successful mutual authentication.

Table 4: Memory architecture of triangular UMAPs.

Protocol	Storage Location	
	Reader	Tag
LMAP M^2AP EMAP	$(ID, IDS, K_1, K_2, K_3, K_4)$	$(ID, IDS, K_1, K_2, K_3, K_4)$

The flow diagram of the LMAP is given in Figure 2. Despite being resource efficient, the LMAP is a weak protocol in terms of structure and equations. The triangular functions alone are unable to conceal the tag's secrets in public messages due to their imbalance nature. Several cryptanalysis attacks on the LMAP have proved that the protocol cannot be used as a standard for the RFID authentication purposes.

2.1.2. Minimalistic Mutual Authentication Protocol (M^2AP). The second protocol from the triangular UMAP family is the M^2AP. This protocol is similar to the LMAP in terms of the tag's memory architecture and the protocol's primitives. The basic difference between the two protocols is the composition of public message Y. The memory architecture of the tag implementing the M^2AP is given in Table 4. The step by step execution of the protocol is elaborated as follows:

(1) The reader "*pings*" the tag, detected in its vicinity.

(2) The tag responds with the *IDS* which acts as an index to locate the data associated with the tag in the database for successful tag identification.

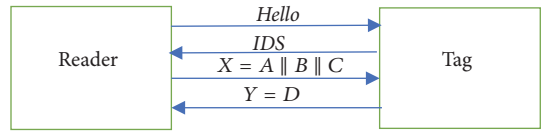
(3) After the tag identification, the reader generates two pseudorandom numbers n_1 and n_2. The reader then computes and transmits challenge message $X = A \parallel B \parallel C$ to the tag.

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

$$B = (IDS \wedge K_2) \vee n_1 \quad (11)$$

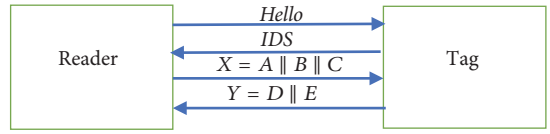
$$C = IDS + K_3 + n_2 \quad (12)$$

(4) The tag extracts n_1 and n_2 from A and C, respectively, and verifies the identity of the reader by calculating a response for message B. After successful reader



$A = IDS \oplus K_1 \oplus n_1$	$IDS^{NEW} = (IDS + (n_2 \oplus K_4)) \oplus ID$
$B = (IDS \vee K_2) \oplus n_1$	$K_1^{NEW} = K_1 \oplus n_2 \oplus (K_3 + ID)$
$C = IDS + K_2 + n_2$	$K_2^{NEW} = K_2 \oplus n_2 \oplus (K_4 + ID)$
$D = (IDS + ID) \oplus n_1 \oplus n_2$	$K_3^{NEW} = K_3 \oplus n_1 \oplus (K_1 + ID)$
	$K_4^{NEW} = K_4 \oplus n_1 \oplus (K_2 + ID)$

Figure 2: Block diagram of LMAP.



$A = IDS \oplus K_1 \oplus n_1$	$IDS^{NEW} = (IDS + (n_1 \oplus n_2)) \oplus ID$
$B = (IDS \wedge K_2) \vee n_1$	$K_1^{NEW} = K_1 \oplus n_2 \oplus (K_3 + ID)$
$C = IDS + K_3 + n_2$	$K_2^{NEW} = K_2 \oplus n_2 \oplus (K_4 + ID)$
$D = (IDS \vee K_4) \wedge n_2$	$K_3^{NEW} = K_3 \oplus n_1 \oplus (K_1 + ID)$
$E = (IDS + ID) \oplus n_1$	$K_4^{NEW} = (K_4 \oplus n_1) \oplus (K_2 + ID)$

Figure 3: Block diagram of M^{2}AP.

authentication, the tag calculates and transmits challenge message $Y = D \parallel E$.

$$D = (IDS \vee K_4) \wedge n_2 \tag{13}$$

$$E = (IDS + ID) \oplus n_1 \tag{14}$$

The message D is used for the tag authentication whereas the message E is used for the ID communication.

(5) After successful mutual authentication, the dynamic memory on both sides is updated using the following equations:

$$IDS^{NEW} = (IDS + (n_1 \oplus n_2)) \oplus ID \tag{15}$$

$$K_1^{NEW} = K_1 \oplus n_2 \oplus (K_3 + ID) \tag{16}$$

$$K_2^{NEW} = K_2 \oplus n_2 \oplus (K_4 + ID) \tag{17}$$

$$K_3^{NEW} = K_3 \oplus n_1 \oplus (K_1 + ID) \tag{18}$$

$$K_4^{NEW} = (K_4 \oplus n_1) \oplus (K_2 + ID) \tag{19}$$

The block diagram of the protocol is given in Figure 3. The cryptanalysis of M^{2}AP was similar to that of LMAP due to similarity in composition of public messages equations and memory architectures.

2.1.3. Efficient Mutual Authentication Protocol (EMAP). The EMAP is the third most prominent protocol from the triangular class. The primitives used for the encryption of communication between the tag/reader pair are *AND*, *OR*, and *XOR*. The memory architecture of the tag implementing the EMAP is given in Table 4. The working principle of the protocol is as follows:

(1) The tag receives a "*Hello*" message from the reader as it enters its communication range.

(2) The reader receives the *IDS*, which is used for the tag identification by locating the data associated with the communicating tag in the system's database.

(3) Once the tag is identified, the reader generates the random numbers and sends message $X = A \parallel B \parallel C$ to the tag.

$$A = IDS \oplus K_1 \oplus n_1 \tag{20}$$

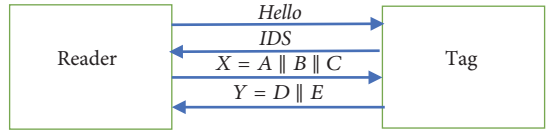
$$B = (IDS \vee K_2) \oplus n_1 \tag{21}$$

$$C = (IDS \oplus K_3) \oplus n_2 \tag{22}$$

(4) The tag extracts n_1 from message A and authenticates the reader by calculating the response for challenge message B. After successful reader authentication the tag extracts n_2 from the message C, to calculate and send challenge message $Y = D \parallel E$.

$$D = (IDS \wedge K_4) \oplus n_2 \tag{23}$$

$$E = (IDS \wedge n_1 \vee n_2) \oplus ID \oplus K_1 \oplus K_2 \oplus K_3 \oplus K_4 \tag{24}$$



$A = IDS \oplus K_1 \oplus n_1$	$IDS^{NEW} = IDS \oplus n_2 \oplus K_1$
$B = (IDS \vee K_2) \oplus n_1$	$K_1^{NEW} = K_1 \oplus n_2 \oplus (ID(1:48) \parallel F_p(K_4) \parallel F_p(K_3))$
$C = (IDS \oplus K_3) \oplus n_2$	$K_2^{NEW} = K_2 \oplus n_2 \oplus (F_p(K_4) \parallel F_p(K_4) \parallel ID(49:96))$
$D = (IDS \wedge K_4) \oplus n_2$	$K_3^{NEW} = K_3 \oplus n_1 \oplus (ID(1:48) \parallel F_p(K_4) \parallel F_p(K_2))$
$E = (IDS \wedge n_1 \vee n_2) \oplus ID \oplus K_1 \oplus K_2 \oplus K_3 \oplus K_4$	$K_4^{NEW} = K_4 \oplus n_1 (F_p(K_3) \parallel F_p(K_1) \parallel ID(49:96))$

Figure 4: Block diagram of EMAP.

(5) The authentication session ends by updating the dynamic memory on both sides.

$$IDS^{NEW} = IDS \oplus n_2 \oplus K_1 \tag{25}$$

$$K_1^{NEW} = K_1 \oplus n_2 \oplus \left(ID(1:48) \parallel F_p(K_4) \parallel F_p(K_3)\right) \tag{26}$$

$$K_2^{NEW} = K_2 \oplus n_2 \oplus \left(F_p(K_4) \parallel F_p(K_4) \parallel ID(49:96)\right) \tag{27}$$

$$K_3^{NEW} = K_3 \oplus n_1 \oplus \left(ID(1:48) \parallel F_p(K_4) \parallel F_p(K_2)\right) \tag{28}$$

$$K_4^{NEW} = K_4 \oplus n_1 \left(F_p(K_3) \parallel F_p(K_1) \parallel ID(49:96)\right) \tag{29}$$

The function $F_p(x)$ generates a 24*bit* version of 96*bit* input x. The input is divided into twenty-four groups by combining 4*bits* in each group. The final output is obtained by taking bitwise *XOR* of all the entities present in each group and concatenating the result. The block diagram of the protocol is given in Figure 4.

2.2. UMAP with Single Nontriangular Function. The resource limitation of EPC C1G2 tags confines the computational cost of the UMAPs to 4*K* GE. Initially, the UMAPs only used the triangular functions for the calculation of the *challenge*/*response* messages. But triangular protocols were prone to multiple security attacks due to the lack of diffusion in the public messages. The reason behind the inability of encrypted string to conceal the secret values associated with the tag was the imbalance nature of the protocol's operators.

In 2007, Chien [21] introduced the idea of the ultralightweight nontriangular primitive as the protocol's operator. The use of single nontriangular primitive improved the strength of the UMAPs; however the cryptanalysis of nontriangular UMAPs still highlighted weaknesses in the protocol structure and operators. Some of the prominent UMAPs with single nontriangular primitives are Strong Authentication Strong Integrity (SASI) protocol [21], Gossamer's protocol [38], and Yeh et al. protocol [39].

2.2.1. Strong Authentication Strong Integrity Protocol. The SASI protocol was the first protocol in the field of nontriangular UMAPs. The nontriangular function used in the SASI protocol is the rotation function ($Rot(x, y)$). The rotation function has two definitions: left rotation of x by the hamming weight of y and left rotation of x *by mod of* y. For this section, we will consider hamming weight-based rotation function. The memory architecture of the tag implementing the SASI protocol is elaborated in Table 5. The reason behind storing the pair of latest dynamic variables was to provide protection against Denial of Service (DoS) attacks. The working principles of the SASI protocol are as follows:

(1) The reader requests the tag for a pseudoidentification number.

(2) The tag transmits its latest *IDS*. If the received *IDS* is found in the database, the protocol proceeds further otherwise the reader requests the tag for the *IDS* from the previous successful authentication session (IDS^{old}). The successful tag identification leads to the step (3).

(3) The reader generates two random numbers n_1 and n_2. The dynamic variables and random numbers are used by the reader to generate and transmit $X = A \parallel B \parallel C$.

$$A = IDS \oplus K_1 \oplus n_1 \tag{30}$$

$$B = (IDS \vee K_2) \oplus n_2 \tag{31}$$

$$\overline{K_1} = Rot((K_1 \oplus n_2), K_1) \tag{32}$$

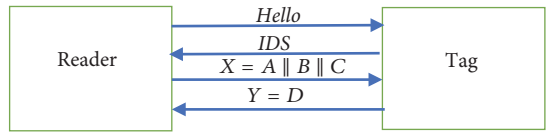
$$\overline{K_2} = Rot((K_2 \oplus n_1), K_2) \tag{33}$$

$$C = (K_1 \oplus \overline{K_2}) + (\overline{K_1} \oplus K_2) \tag{34}$$

(4) The pseudorandom numbers n_1 and n_2 are concealed in and communicated to the tag via message A and B, respectively. The message C is used for the reader authentication.

Table 5: Memory architectures of UMAPs with single nontriangular function.

Protocol	Storage Location	
	Reader	Tag
SASI	(ID, IDS, K_1, K_2)	$(ID, IDS, K_1, K_2, IDS^{old}, K_1^{old}, K_2^{old})$
Gossamer's Protocol	(ID, IDS, K_1, K_2)	$(ID, IDS, K_1, K_2, IDS^{old}, K_1^{old}, K_2^{old})$
Yeh et al.'s Protocol	$(ID, IDS, IDS^{old}, K, K^{old})$	(ID, IDS, K)



$A = IDS \oplus K_1 \oplus n_1$	$IDS^{old} = IDS$
$B = (IDS \vee K_2) \oplus n_2$	$K_1^{old} = K_1$
$\overline{K_1} = Rot((K_1 \oplus n_2), K_1)$	$K_2^{old} = K_2$
$\overline{K_2} = Rot((K_2 \oplus n_1), K_2)$	$IDS = (IDS^{old} + ID) \oplus (n_2 \oplus \overline{K_1})$
$C = (K_1 \oplus \overline{K_2}) + (\overline{K_1} \oplus K_2)$	$K_1 = \overline{K_1}$
$D = (\overline{K_2} + ID) \oplus ((K_1 \oplus K_2) \bigvee \overline{K_1})$	$K_2 = \overline{K_2}$

Figure 5: Block diagram of SASI protocol.

(5) After successful reader authentication, the tag transmits message $Y = D$ for the tag authentication and the ID transmission.

$$D = \left(\overline{K_2} + ID\right) \oplus \left((K_1 \oplus K_2) \vee \overline{K_1}\right) \tag{35}$$

(6) After mutual authentication, the dynamic variables on both sides are updated using following equations:

$$\begin{aligned} IDS^{old} &= IDS, \\ K_1^{old} &= K_1, \\ K_2^{old} &= K_2 \end{aligned} \tag{36}$$

$$\begin{aligned} IDS &= \left(IDS^{old} + ID\right) \oplus \left(n_2 \oplus \overline{K_1}\right), \\ K_1 &= \overline{K_1}, \\ K_2 &= \overline{K_2} \end{aligned} \tag{37}$$

The flowchart of the SASI protocol is given in Figure 5.

2.2.2. Gossamer's Protocol. In 2008, Peris-Lopez presented nontriangular UMAPs to overcome the weaknesses of the SASI protocol. In the Gossamer's protocol, the memory architecture of the system was enhanced by saving the latest copy of dynamic variables on the tag's side. The memory architecture of the protocol is given in Table 5. The nontriangular primitive of the Gossamer's protocol is mix bit function ($mixbit(a, b)$). The mix bit function consists of two subfunction: the rotation and the modular addition function. These subfunctions are used independently and in collaborative manner to calculate the challenge/response messages. The working principle of $x = mixbit(a, b)$ is elaborated in Figure 7.

The protocol executes in five steps which are defined as follows:

(1) The reader sends a request for the IDS to the tag present in its vicinity.

(2) The reader tries to locate the tags information by searching the database with the help of the received IDS. The tag is identified if its information is found in the database.

(3) The reader generates pseudorandom private keys n_1 and n_2. The reader then sends message $X = A \parallel B \parallel C$ to the tag.

$$A = Rot\left(\left(Rot\left(IDS + K_1 + \pi + n_1, K_2\right) + K_1\right), K_1\right) \tag{38}$$

$$B = Rot\left(\left(Rot\left(IDS + K_2 + \pi + n_2, K_1\right) + K_2\right), K_2\right) \tag{39}$$

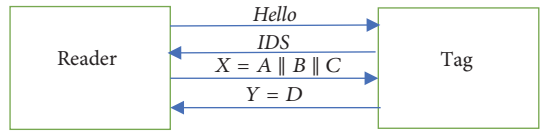
$$n_3 = Mixbit\left(n_1, n_2\right) \tag{40}$$

$$K_1^* = Rot\left(\left(Rot\left(n_2 + K_1 + \pi + n_3, n_2\right) + K_2 \oplus n_3, n_1\right) \oplus n_3\right) \tag{41}$$

$$K_2^* = Rot\left(\left(Rot\left(n_1 + K_2 + \pi + n_3, n_1\right) + K_1 + n_3, n_2\right) + n_3\right) \tag{42}$$

$$n_1' = Mixbit\left(n_3, n_2\right) \tag{43}$$

$$C = Rot\left(\left(Rot\left(n_3 + K_1^* + \pi + n_1', n_3\right) + K_2^* \oplus n_1', n_2\right) \oplus n_1'\right) \tag{44}$$



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

Figure 6: Block diagram of Gossamer's protocol.

$$x = mixbit(a, b)$$
$$x = a$$
$$for(i = 0; i + +; i < 96)$$
$$\{x = (x << 1) + x + x + b\}$$

Figure 7: $mixbit(a, b)$ algorithm.

(4) The reader is authenticated by generating a response to the message C. After that, the tag calculates and transmits the challenge message $Y = D$.

$$D = Rot\left(\left(Rot\left(n_2 + K_2^* + ID + n_1', n_2\right) + K_1^* + n_1', n_3\right) + n_1\right) \quad (45)$$

(5) The dynamic variable on both sides are updated after a successful mutual authentication.

$$IDS^{old} = IDS;$$
$$K_1^{old} = K_1; \quad (46)$$
$$K_2^{old} = K_2$$
$$n_2' = Mixbit\left(n_1', n_3\right) \quad (47)$$
$$IDS = Rot\left(\left(Rot\left(n_1' + K_1^* + IDS + n_2', n_1'\right) + K_2^* \oplus n_2', n_3\right) \oplus n_2'\right) \quad (48)$$
$$K_1 = Rot\left(\left(Rot\left(n_3 + K_2^* + \pi + n_2', n_3\right) + K_1^* + n_2', n_1'\right) + n_2'\right) \quad (49)$$
$$K_2 = Rot\left(\left(Rot\left(IDS^{new} + K_2^* + \pi + K_1^{new}, IDS^{new}\right) + K_1^* + K_1^{new}, n_2'\right) + K_1^{new}\right) \quad (50)$$

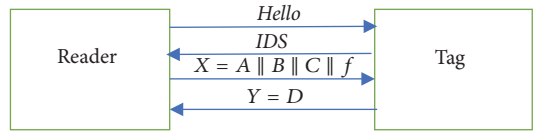
The constant π used in the protocol assumes the value 0 $x3243F6A8885A308D313198A2$. The block diagram of the Gossamer's protocol is presented in Figure 6.

2.2.3. Yeh et al. Protocol. In 2010, Yeh et al. [39] proposed a process oriented UMAP. The feature that differentiates this protocol from its predecessors is the DoS avoidance mechanism. In this protocol, the pairs of latest dynamic variables are stored at the reader side instead of the tag. The reader also maintains a flag to identify whether the tag/reader pair is fully synchronized or not. The nontriangular function used in the protocol is the rotation function ($Rot(a, b)$). The memory architecture of the UMAP is given in Table 5. The working principle of the Yeh et al. protocol is as follows: the protocol is the rotation function ($Rot(a, b)$). The memory architecture of the UMAP is as follows:

(1) The reader initiates the communication by sending a "*Hello*" message to the tag.

(2) As a response, the tag transmits the IDS stored in its dynamic memory.

(3) After successful tag identification through the database, the reader generates two pseudorandom numbers n_1 and n_2. If the $IDS = IDS^{new}$, the reader sets an internal flag $f = 0$; otherwise the flag's value sets to 1, the key K updates and becomes equal to the tag's ID. After key updation, the reader calculates and sends message $X = A \parallel B \parallel C \parallel f$ to the tag.



$A = (IDS \oplus K) \oplus n_1$	$\ddot{K}^* = Rot(K \oplus n_1, n_2)$
$B = (IDS \vee K) \oplus n_2$	$D = (\ddot{K}^* \oplus n_2) + n_1$
$K^* = Rot(K \oplus n_2, n_1)$	$IDS^{New} = (IDS + (ID \oplus \ddot{K}^*)) \oplus n_1 \oplus n_2$
$C = (K^* \oplus n_1) + n_2$	$K^{New} = K^*$
$f = flag\ bit;\ \begin{cases} f = 0\ if\ IDS = IDS^{NEW} \\ f = 1\ if\ IDS = IDS^{OLD} \end{cases}$	

Figure 8: Block diagram of Yeh et al. protocol.

$$A = (IDS \oplus K) \oplus n_1 \tag{51}$$

$$B = (IDS \vee K) \oplus n_2 \tag{52}$$

$$K^* = Rot(K \oplus n_2, n_1) \tag{53}$$

$$C = (K^* \oplus n_1) + n_2 \tag{54}$$

$$f = flag\ bit;\quad \begin{cases} f = 0 & if\ IDS = IDS^{NEW} \\ f = 1 & if\ IDS = IDS^{OLD} \end{cases} \tag{55}$$

(4) Upon receiving the challenge message, the tag updates the value of the key K based on the flag status. After that n_1 and n_2 are extracted and the reader is authenticated.

(5) The successful reader verification leads to the calculation and transmission of the tag authentication challenge message D.

$$\ddot{K}^* = Rot(K \oplus n_1, n_2) \tag{56}$$

$$D = (\ddot{K}^* \oplus n_2) + n_1 \tag{57}$$

(6) In case of successful mutual authentication, the dynamic memory on both sides is updated.

$$IDS^{New} = (IDS + (ID \oplus \ddot{K}^*)) \oplus n_1 \oplus n_2 \tag{58}$$

$$K^{New} = K^* \tag{59}$$

The block diagram of the Yeh et al. protocol is given in Figure 8.

2.3. UMAPs with Hybrid Nontriangular Function. The cryptanalysis of the UMAPs with single nontriangular functions proved the inability of the protocols to provide Confidentiality, Integrity, Availability and Authentication (CIAA) to the communicating parties. In order to further improve the security, the concept of using hybrid nontriangular functions was introduced. This idea improved the security and privacy services offered by the UMAPs. There are many hybrid nontriangular UMAPs available in the literature. In this subsection, the protocols under consideration are RFID Authentication Protocol with Permutation (RAPP) [4], RFID Authentication Protocol for Low cost Tags (RAPLT) [40], Robust Confidentiality Integrity and Authentication (RCIA) protocol [41], and Succinct and Lightweight Authentication Protocol (SLAP) [30].

2.3.1. RFID Authentication Protocol with Permutation (RAPP). The RAPP protocol was different from previously presented UMAPs, in terms of the primitives used for encryption and the sequence of interaction between the tag and the reader. The protocol only used three operations, i.e., *XOR*, rotation ($Rot(x, y)$) and permutation ($Per(x, y)$).

The $Rot(x, y)$ corresponds to the left rotation of x by the hamming weight of y. The description of permutation function ($z = per(x, y)$) is as follows. Let z be a $L - bit$ word and z_i be the ith bit of z where $1 \le i \le L$, and z_1 and z_L be the LSB and MSB of the word z, respectively. Suppose x and y are two L-bit words and hamming weight of y is m. Moreover $y_i = 1$ if $i \in I_1 = \{k_m, k_{m-1} \dots k_1\}$ and $y_i = 0$ if $i \in I_0 = \{k_L, k_{L-1} \dots k_{m+1}\}$.

$$\begin{aligned} k_m &> k_{m-1} > \cdots > k_1 \\ k_L &> k_{L-1} > \cdots > k_{m+1} \end{aligned} \tag{60}$$

The permutation of x according to y, i.e., $per(x, y)$ is equal to

$$Per(x, y) = x_{k_m}, x_{k_m - 1} \dots x_{k_1}, x_{k_L}, x_{k_{L-1}} \dots x_{k_{m+1}} \tag{61}$$

Unlike conventional sequence of interaction, the dynamic memory of the tag is updated after getting a confirmation message of successful mutual authentication of the tag/reader pair. The aim of this message was to make the protocol resistant to the desynchronization attacks. The memory architecture of the protocol is given in Table 6 and the detail description of the protocol is as follows:

(1) The tag receives a "*Hello*" message from the reader as soon as it enters its vicinity.

(2) The tag responds with the value of *IDS* stored in its dynamic memory. The reader identifies the tag by

Table 6: Memory architecture of UMAPs with hybrid nontriangular functions.

Protocol	Storage Location	
	Reader	Tag
RAPP	$(ID, IDS, K_1, K_2, K_3, IDS^{old}, K_1^{old}, K_2^{old}, K_3^{old})$	(ID, IDS, K_1, K_2, K_3)
RAPLT	$(ID, IDS, K_1, K_2, IDS^{old})$	$(ID, IDS, K_1, K_2, IDS^{old})$
RCIA	$(ID, IDS, K_1, K_2, IDS^{old}, K_1^{old}, K_2^{old})$	$(ID, IDS, K_1, K_2, IDS^{old}, K_1^{old}, K_2^{old})$
SLAP	$ID, IDS^{NEW}, K_1^{NEW}, K_2^{NEW}, IDS^{OLD}, K_1^{OLD}, K_2^{OLD}$	$ID, IDS^{NEW}, K_1^{NEW}, K_2^{NEW}, IDS^{OLD}, K_1^{OLD}, K_2^{OLD}$

retrieving the information indexed by the *IDS* value in the database.

(3) After successful tag identification, the reader generates a random number n_1 and sends challenge message $X = A \parallel B$ to the tag.

$$A = Per\left(K_2, K_1\right) \oplus n_1 \tag{62}$$

$$B = Per\left(K_1 \oplus K_2, Rot\left(n_1, n_2\right)\right) \oplus Per\left(n_1, K_1\right) \tag{63}$$

(4) The tag verifies the reader's identity and sends the challenge message $Y = C$ to the reader.

$$C = Per\left(n_1 \oplus K_1, n_1 \oplus K_3\right) \oplus ID \tag{64}$$

(5) In case of successful mutual authentication, the reader updates the dynamic variables, generates another random number n_2, and sends the mutual authentication verification message $D \parallel E$ to the tag.

$$D = Per\left(K_3, K_2\right) \oplus n_2 \tag{65}$$

$$E = Per\left(K_3, Rot\left(n_2, n_2\right)\right) \oplus Per\left(n_1, K_3 \oplus K_2\right) \tag{66}$$

(6) The tag updates its dynamic memory after verifying the origin of message $D \parallel E$.

$$IDS^{New} = Per\left(IDS, n_1 \oplus n_2\right) \oplus K_1 \oplus K_2 \oplus K_3 \tag{67}$$

$$K_1^{New} = Per\left(K_1, n_1\right) \oplus K_2 \tag{68}$$

$$K_2^{New} = Per\left(K_2, n_2\right) \oplus K_1 \tag{69}$$

$$K_3^{New} = Per\left(K_3, n_1 \oplus n_2\right) \oplus IDS \tag{70}$$

The block diagram of the protocol is given in Figure 9.

2.3.2. RFID Authentication Protocol for Low Cost Tags (RAPLT). In 2013, Jeon and Yoon improved the permutation function and proposed two new nontriangular function, i.e., Merge ($Mer(a, b, c, d)$) and separate ($Sep(d, c, b, , a)$) operations in RFID Authentication Protocol for Low cost Tags (RAPLT). These operations are considered to be more reliable and secure compared to the permutation function.

Assume a and b are L bit numbers whereas c and d are $2L$ bit strings. The formation of a, b, c, d and pseudocode for $Mer(a, b, c, d)$ and $Sep(d, c, b, , a)$ operations are given in Figure 10.

Both operations have inverse relation and are extremely lightweight in nature. In RAPLT protocol, both the tag and the reader stores a pair of the latest *IDS*, the latest keys and the tag's *ID*. The working principle of the RAPLT protocol is as follows:

(1) The reader initiates the protocol by sending a "*Hello*" message to the tag.

(2) As a response, the tag sends the index pseudonym (*IDS*) for the tag identification.

(3) After the successful tag identification through *IDS*, the reader generates two random numbers (n_1, n_2) and computes $X = A_1 \parallel A_2 \parallel B_3$.

$$N_1 = n_1 \oplus ID; \quad N_2 = n_2 \oplus IDS \tag{71}$$

$$Mer\left(N_1, N_2 \cdot K_1 \parallel K_2, A_1 \parallel A_2\right) \tag{72}$$

$$M_1 = ID \oplus n_1 \oplus K_2; \quad M_2 = IDS \oplus n_2 \oplus K_1 \tag{73}$$

$$Sep\left(M_1, M_2 \cdot K_1 \parallel K_2, B_1 \parallel B_2\right) \tag{74}$$

$$B_3 = B_1 \oplus B_2 \tag{75}$$

(4) The tag authenticates the reader by generating a response for the message B_3. A successful reader verification leads to the calculation and transmission of $Y = C_3$.

$$Mer\left(K_1, K_2, K_1 \parallel K_2, K'_1 \parallel K'_2\right) \tag{76}$$

$$Mer\left(n_2, N_1, K'_1 \parallel K'_2, C_1 \parallel C_2\right) \tag{77}$$

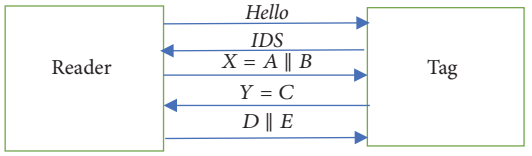
$$C_3 = C_1 \oplus C_2 \tag{78}$$

(5) The tag authentication is followed by the *IDS* update on both sides.

$$IDS^{NEW} = n_1 \tag{79}$$

The flow diagram of the RAPLT protocol is given in Figure 11.

2.3.3. Robust Confidentiality Integrity and Authentication Protocol (RCIA). The Robust Confidentiality Integrity and Authentication (RCIA) protocol is designed on the theme of



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

Figure 9: Block diagram of RAPP protocol.

$\boldsymbol{a} = a_1 a_2 a_3 \ldots a_L; \boldsymbol{b} = b_1 b_2 b_3 \ldots b_L$	
$\boldsymbol{c} = c_1 c_2 c_3 \ldots c_{2L}; \boldsymbol{d} = d_1 d_2 d_3 \ldots d_{2L}$	
$(\boldsymbol{Mer}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}))$	$(\boldsymbol{Sep}(\boldsymbol{d}, \boldsymbol{c}, \boldsymbol{b}, , \boldsymbol{a}))$

$i, j = 1$
for $n = 1$ *to* $2L$
If $c_n = 0$
$d_n = a_i$ & $i = i + 1$
else
$d_n = b_j$ & $j = j + 1$
end if
	$i, j = 1$
	for $n = 1$ *to* $2L$
	If $c_n = 0$
	$a_i = d_n$ & $i = i + 1$
	else
	$IDS = IDS^{Old}$
	$b_j = d_n$ & $j = j + 1$
	end if

Figure 10: Merge and separate operate.

the RAPP protocol. This protocol is associated with hybrid category of the UMAPs as it uses two nontriangular functions, i.e., rotation ($Rot(x, y)$) and recursive hash ($R_h(x)$). The working principle of the recursive hash ($R_h(x)$) functions consists of following steps:

(i) Consider x as an L bit string and decimate the input x into k chunks with equal numbers of bits per chunk. ($\#\ of\ bits\ per\ chunk = L/k$).

(ii) Assume a seed value s from the range $[0, k-1]$.

(iii) The seed s calculated in above step selects the corresponding memory block (k_s) of the decimated string x.

(iv) Final answer of recursive hash function is obtained by concatenating the results of following operations.

(a) Take *XOR* between the selected memory block k_s and all the other blocks except the block itself.

(b) Left rotate the block k_s by the hamming weight of itself ($Rot(k_s, k_s)$).

For efficient hardware implementation, the 96 *bit* input of the recursive hash function is decimated into $k = 12$ chunks, each containing 8 *bits*. Both the tag and the reader store seven *L bits* strings associated with the tag. These numbers are ID, (IDS, K_1, K_2) and $(IDS^{old}, K_1^{old}, K_2^{old})$. The RCIA protocol executes in five steps which are as follows:

(1) The reader sends a "*Hello*" message to the tag.

(2) The tag sends IDS to the reader. If the value is found in the database, the protocol proceeds otherwise the reader requests for IDS^{old} and matches it with the database value. The protocol proceeds to the next step only when the received IDS is found in the database.

(3) The reader generates random numbers n_1 and n_2. It also calculates $R = n_1 \oplus n_2$. This value is used to find the seed value. The equation for seed calculation is $s = R \mod k$. The calculation of seed value leads to calculation and transmission of $X = A \parallel B \parallel C$ messages.

$$A = Rot\ (IDS, K_1) \oplus n_1 \tag{80}$$

$$B = (Rot\ (IDS \wedge n_1, K_2) \wedge K_1) \oplus n_2 \tag{81}$$

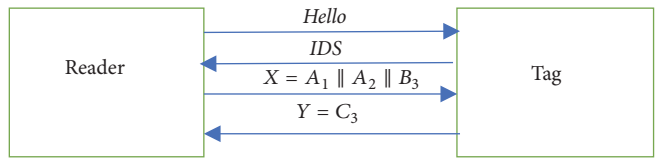
$$\begin{aligned} R &= n_1 \oplus n_2; \\ s &= R \mod k \end{aligned} \tag{82}$$

$$K_1^* = Rot\ (R_h\ (K_2), R_h\ (n_1)) \wedge K_1 \tag{83}$$

$$K_2^* = Rot\ (R_h\ (K_1), R_h\ (n_2)) \wedge K_2 \tag{84}$$

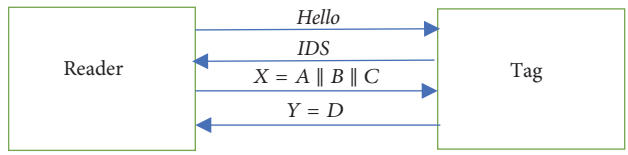
$$\begin{aligned} C = &Rot\ (R_h\ (K_1^*), R_h\ (K_2^*)) \\ &\wedge Rot\ (R_h\ (n_1), R_h\ (n_2)) \end{aligned} \tag{85}$$

(4) The message C is used for the reader authentication. After one sided successful authentication, the tag



$N_1 = n_1 \oplus ID$	$B_3 = B_1 \oplus B_2$
$N_2 = n_2 \oplus IDS$	$Mer(K_1, K_2, K_1 \| K_2, K'_1 \| K'_2)$
$Mer(N_1, N_2.K_1 \| K_2, A_1 \| A_2)$	$Mer(n_2, N_1, K'_1 \| K'_2, C_1 \| C_2)$
$M_1 = ID \oplus n_1 \oplus K_2; M_2 = IDS \oplus n_2 \oplus K_1$	$C_3 = C_1 \oplus C_2$
$Sep(M_1, M_2.K_1 \| K_2, B_1 \| B_2)$	$IDS^{NEW} = n_1$

Figure 11: Block diagram of RAPLT.



$A = Rot(IDS, K_1) \oplus n_1$	$D = (Rot(R_h(ID), K_1^*) \wedge (Rot(R_h(K_2^*), R_h(n_2)) \oplus IDS)$
$B = (Rot(IDS \wedge n_1, K_2) \wedge K_1) \oplus n_2$	
$R = n_1 \oplus n_2; s = R \bmod k$	$IDS^{NEW} = Rot((R_h(IDS) \oplus n_2, n_1)$
$K_1^* = Rot(R_h(K_2), R_h(n_1)) \wedge K_1$	$K_1^{NEW} = K_1^*$
$K_2^* = Rot(R_h(K_1), R_h(n_2)) \wedge K_2$	$K_2^{NEW} = K_2^*$
$C = Rot(R_h(K_1^*), R_h(K_2^*)) \wedge Rot(R_h(n_1), R_h(n_2))$	

Figure 12: Block diagram of RCIA protocol.

updates its dynamic variables and sends the message D.

$$D = (Rot(R_h(ID), K_1^*) \wedge (Rot(R_h(K_2^*), R_h(n_2)) \oplus IDS) \quad (86)$$

(5) The reader uses the string D for the tag authentication after which the dynamic variables also updates on the reader's side. The update equations are as follows:

$$IDS^{NEW} = Rot((R_h(IDS) \oplus n_2, n_1) \quad (87)$$

$$K_1^{NEW} = K_1^*; \quad K_2^{NEW} = K_2^* \quad (88)$$

Figure 12 shows the block diagram of the RCIA protocol.

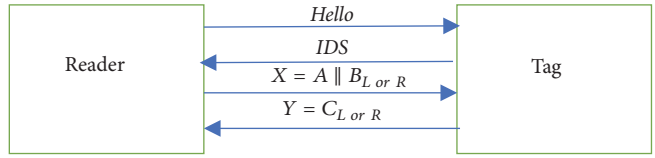
2.3.4. Succinct and Lightweight Authentication Protocol (SLAP). In 2016, an ultralightweight authentication protocol named Succinct and Lightweight Authentication Protocol (SLAP) was proposed. The SLAP algorithm is composed of three operators, i.e., *XOR*, rotation ($Rot(a, b)$), and Conversion ($Conv(a, b)$) function. These functions are lightweight with respect to the implementation cost and are appropriate for the passive electronic chips. The conversion function is the main feature of the protocol that guarantees irreversibility, confidentiality, full confusion, and low complexity.

The conversion function ($Conv(a, b)$) consists of three subfunctions. Suppose the size of input strings (a, b) is L bits, i.e.,

$$a = a_L a_{L-1} a_{L-2} \ldots a_1; \quad b = b_L b_{L-1} b_{L-2} \ldots b_1 \quad (89)$$

The description of these functions is as follows:

(i) *Grouping.* The inputs a and b are divided into segments based on the hamming weight and a threshold t. Consider the input a; based on the hamming weight $m = Hw(a)$ divide the input string into two parts, i.e., $(a^1 = a_L a_{L-1} \ldots a_{m+1})$ and $(a^2 = a_m a_{m-1} \ldots a_1)$. Continue the segmentation process based on hamming weight until the smallest segment size becomes equal to the threshold value t. The input string b is also segmented based on $n = hw(b)$ and the threshold t. The respective segments are concatenated to form the output (a', b') of grouping function.



$A = Conv(K_1, K_2) \oplus n$	$IDS^{NEW} = Conv(IDS, n \oplus (B_{L\ or\ R} \parallel C_{L\ or\ R})$
$B = Conv(Rot(K_1, n), K_1 \oplus K_2) \oplus Rot(Conv(K_2, K_2 \oplus n), K_1)$	$K_1^{NEW} = Conv(K_1, n) \oplus K_2$
$C = Conv(Conv(B, K_1^{NEW}), Conv(K_1^{NEW}, K_2^{NEW} \oplus n)) \oplus ID$	$K_2^{NEW} = Conv(K_2, n) \oplus K_1$

Figure 13: Block diagram of SLAP.

(ii) *Rearrange*. In this step, the regrouping of a' and b' bits takes place. As the length of input strings is same, exchanging the grouping form between of a' and b' gives two L-bit numbers. Finally, each subgroup is left rotated by its hamming weight. The output of rearrange function (a'', b'') is the shuffled version of a and b.

(iii) *Composition*. The final output of conversion function is obtained by taking *XOR* of the shuffled version of a and b, i.e., $c = a'' \oplus b''$.

The RFID system implementing the SLAP stores the latest pair of dynamic variables on both communicating ends. The working principle of the protocol is as follows:

(1) The tag receives a "*hello*" message from the reader after entering its vicinity.

(2) The tag responds with its identity pseudonym *IDS*. This value is used for the tag identification at the reader's side.

(3) After successful identification, the reader generates a random number n and conceal it in the message A. The reader also generates a challenge message B. The reader transmits message A along with left or right half of B based on $Hw(B)$. If $Hw(B)$=odd, $X = A \parallel B_l$; otherwise $X = A \parallel B_r$.

$$A = Conv(K_1, K_2) \oplus n \tag{90}$$

$$B = Conv(Rot(K_1, n), K_1 \oplus K_2) \oplus Rot(Conv(K_2, K_2 \oplus n), K_1) \tag{91}$$

(4) The tag authenticates the reader by generating a response to message B. After successful reader authentication, the tag calculates the message C and transmits the left or right half of C based on $Hw(C)$. If $Hw(C)$=odd, $Y = C_l$; otherwise $Y = C_r$.

$$C = Conv(Conv(B, K_1^{NEW}), Conv(K_1^{NEW}, K_2^{NEW} \oplus n)) \oplus ID \tag{92}$$

(5) After identity verification, the dynamic variables of each side are updated using following equation:

$$IDS^{NEW} = Conv(IDS, n \oplus (B_{L\ or\ R} \parallel C_{L\ or\ R})) \tag{93}$$

$$K_1^{NEW} = Conv(K_1, n) \oplus K_2; \quad K_2^{NEW} = Conv(K_2, n) \oplus K_1 \tag{94}$$

The block diagram of the SLAP is given in Figure 13.

A brief survey on the existing protocols proves that increase in computational complexity of the authentication mechanism improves the CIAA capabilities of the protocol at the cost of increased gate equivalents. Section 3 presents a set of eminent cryptanalysis models that are being used to evaluate the security and the privacy features of the UMAPs. The literature review shows that almost all the existing UMAPs have been subjected to multiple cryptanalysis attacks. The unavailability of secure and reliable UMAP for RFID enabled IoT networks is one of the major challenges in the standardization of the secure architecture for the resource constraint IoT network perception layer. The subsequent sections present a comprehensive security analysis model to evaluate the strengths of the RFID node authentication protocols, the CIAA analysis of the existing UMAPs based on the presented model, and a secure and reliable UMAP termed as Extremely Good Privacy (EGP) protocol for the authentication of resource constraint IoT nodes.

3. Cryptanalysis Models for UMAPs

Since 2006, numerous UMAPs have been proposed for the EPC C1G2 identification system. However, most of these protocols were very weak and were found to be vulnerable within one year of their introduction [19, 42, 43]. The reason behind this hasty failure was lack of compact security analysis of the protocol at the design stage.

A comprehensive security analysis should perform the formal analysis of a protocol and the strength evaluation against at least three basic cryptanalysis models: desynchronization, traceability, and full disclosure attacks. This section

provides a brief description of the above stated security analysis model along with the cryptanalysis of UMAPs defined in previous section, to highlight the need of secure and reliable authentication protocol for RFID based IoT networks.

3.1. Formal Analysis. The formal analysis is performed to evaluate the protocol's ability to authenticate the communicating entities under multiple channel conditions. The sequence of challenge/response message exchange between the tag and the reader is examined by virtue of following methods.

3.1.1. Logic of Belief Analysis. This method analyzes the public message composition and sequence of interaction between the communicating parties to systematically evaluate the protocol's functionality on an abstract level. The objectives of logic of belief analysis are as follows:

(i) State what is accomplished by the protocol

(ii) Draw attention to unnecessary actions that can be removed from a protocol

(iii) Highlight any encrypted messages that could be sent in clear text

The prominent mathematical models used for the logic of belief analysis are Burrows–Abadi–Needham (BAN) logic model and Gong–Needham–Yahalom (GNY) logic model.

3.1.2. Automated Security Analysis. Automatic Security analysis verifies the ability of the protocol to achieve the designated security goals in the presence of malicious entities. The security analysis tools such as Casper-FDR and Avispa are mathematical frameworks which evaluate the protocol's behavior in multiple hostile environments with the set of axioms.

3.2. Desynchronization Attack Model. This attack model aims to disconnect a valid tag from an identification system by overwriting its dynamic attributes. However, a successful desynchronization attack does not reveal any tag's information to the adversary. The minimum requirements for the adversary to lunch a desync attack is the ability to eavesdrop and replay public messages. Based on the memory architecture of the RFID system, the execution of the attack can be defined for four different scenarios.

Scenario 1 (single copy of *IDS* stored on tag and reader's side). In this scenario the reader and the tag stores latest copy of identity pseudonym (*IDS*). The attack executes on such tag/reader pair in two steps.

(1) The adversary keeps track of an authentication session and blocks the challenge message Y from the tag to the reader. As a consequence of this step, the tag's memory updates whereas the *IDS* on the reader's side remains same.

(2) In the next session, the protocol fails at identification stage when the *IDS* provided by the tag is not found in the reader's dynamic memory.

Table 7: Status of dynamic memory for Scenario 1.

Sessions	Reader's memory (*IDS*)	Tag's memory (*IDS*)
1	IDS^1	IDS^1
2	IDS^1	IDS^2

Table 8: Status of dynamic memory for Scenario 2.

Sessions	Reader's memory (IDS^{NEW})	Tag's memory (IDS^{NEW}, IDS^{OLD})
1	IDS^1	IDS^2, IDS^1
2	IDS^3	IDS^3, IDS^1
3	IDS^3	IDS^2, IDS^1

Table 7 shows the status of system's dynamic memory for each step.

Scenario 2 (pair of latest *IDS* stored at tag's side). This scenario is defined for the identification system in which the tag stores a pair of latest identity pseudonyms (IDS^{NEW}, IDS^{OLD}) whereas the reader only stores the most recent copy of (IDS^{NEW}). The model executes in following steps [44]:

(1) Consider a synchronized pair of the tag and the reader. The adversary eavesdrops challenge message X (X^1) and blocks the challenge message Y (Y^1). As a result, the tag's dynamic memory updates (IDS^2, IDS^1) whereas the reader's database remains unchanged (IDS^1).

(2) The adversary allows the tag/reader pair to undergo an uninterrupted authentication session. The identity verification takes place on the basis of IDS^1. ($IDS_{reader} = IDS^3; IDS_{tag} = \{IDS^3, IDS^1\}$).

(3) In this step the attacker imitates as a valid reader and communicates with the tag based on IDS^1. The adversary replays the challenge message X^1. As a result, the tag's dynamic memory updates as $IDS^2$and IDS^1 whereas the reader's memory remains the same, i.e., IDS^3.

(4) Since the values of *IDS* do not match at the communicating ends, the tag fails in identification stage of preceding authentication sessions.

Table 8 shows the values of index pseudonyms at the end of each step.

Scenario 3 (pair of latest *IDS* stored at the reader's side). The scenario is defined for such protocols in which the reader stores two copies of dynamic memory (IDS^{NEW}, IDS^{OLD}) and the reader also sends a challenge message M to the tag as the last message of the session. The purpose of this message is to intimate the tag about successful mutual authentication so that its dynamic memory can be updated. The step by step execution of the attack is as follows [45]:

Table 9: Status of dynamic memory for Scenario 3.

Sessions	Reader's memory (IDS^{NEW}, IDS^{OLD})	Tag's memory (IDS^{NEW})
1	IDS^2, IDS^1	IDS^1
2	IDS^3, IDS^1	IDS^1
3	IDS^3, IDS^1	IDS^2

(1) The adversary sniffs the public messages IDS^1, X, Y, M from an ongoing authentication session and then blocks the message M. This prevents the tag to update its dynamic variables, i.e., $(IDS_{tag} = IDS^1; IDS_{reader} = \{IDS^2, IDS^1\})$.

(2) In the next session, the adversary allows the tag-reader pair to communicate on the basis of IDS^1 and blocks message M. This step again updates the reader memory whereas the tag's memory remains unchanged, i.e., $(IDS_{tag} = IDS^1; IDS_{reader} = \{IDS^3, IDS^1\})$.

(3) In the last session the adversary impersonates as a reader and replays message X *and* M recorded from step one. This replay attacks breaks the synchronization among the dynamic variables of the tag and the reader. The final values of dynamic variables at the tag and the reader's side are $(IDS_{tag} = IDS^2; IDS_{reader} = \{IDS^3, IDS^1\})$.

The step by step values of index pseudonyms are given in Table 9.

Scenario 4 (pair of latest *IDS* stored on both sides of the system). The last scenario is for the protocols that store the pair of latest dynamic variables on both communicating ends. The adversary requires five consecutive authentication sessions to completely disconnect a valid tag from the RFID system. The description of attack is as follows [34]:

(1) In step one, the adversary eavesdrops all the public messages (IDS^1, X^1, Y^1) from an authentication session between a completely synchronized tag/reader pair.

(2) In the next step, the adversary records IDS^2 and X^2 and block X^2 at the same time. The dynamic memory of both sides remains unchanged.

(3) In step three, the adversary forces the tag/reader pair authentication on the basis of IDS^1 by blocking the first response of the tag to the reader's *hello* message.

(4) In this step, the adversary imitates as the reader and communicates with the tag based on the messages eavesdropped in session one. This step makes the tag partially desynchronized.

(5) The last step comprises of the adversary's communication with the tag on the basis of IDS^2 and X^2. This step completely changes the values of identity pseudonyms stored in the tag's and the reader's memory.

Table 10: Status of dynamic memory for Scenario 4.

Sessions	Reader's memory (IDS^{NEW}, IDS^{OLD})	Tag's memory (IDS^{NEW}, IDS^{OLD})
1	IDS^1, IDS^0	IDS^1, IDS^0
2	IDS^1, IDS^0	IDS^1, IDS^0
3	IDS^3, IDS^1	IDS^3, IDS^1
4	IDS^3, IDS^1	IDS^2, IDS^1
5	IDS^3, IDS^1	IDS', IDS^2

The working example of the attack is presented in Table 10.

The scenarios covered in Tables 7, 8, 9, and 10 cover almost all the previous protocols. This proves that nearly every UMAP have been subjected to desynchronization attack which ultimately leads to Denial of Service (DoS). The basic theme of all the DoS attacks is to rewrite the tag's memory with such previous values of *IDS* that have been removed from the reader's memory. The generalized desynchronization attack proved that if the pair of latest dynamic variables are stored at the reader's side, the tag can be desynchronized in maximum five consecutive sessions, irrespective of its dynamic memory architecture [34].

An extended memory buffer for the tag's dynamic variables at the reader's database increases the number of sessions required by the adversary to overwrite the tag's memory. The increase in number of adversary administered session requirements for the execution of desynchronization attack strengthens the protocol's ability to withstand DoS attacks [46, 47].

3.3. Traceability Attack Model. One of the most prominent threats associated with the RFID system is traceability. In this model, the adversary gathers information related to the tag so that it can violate its location privacy at any point of time in future. The UMAPs can resist the traceability attacks by anonymizing the tag's response to the reader's queries.

According to the formal definition, the tag (T_0) is assumed traceable, if the adversary can correctly estimate the value of b when presented with $IDS_i^{T_b}$ from the set $\{IDS_i^{T_0}, IDS_i^{T_1}\}$ [48, 49]. Two basic models are available in literature to evaluate the strength of the protocol for preserving the anonymity of the tag.

3.3.1. Guess and Determine Model. In guess and determine model, the attacker has following capabilities:

(i) *Execute* (R, T, i). The attacker can snoop the communication between the tag (T) and the reader (R) during the i^{th} authentication session.

(ii) *Send* (X, Y, M, i). The adversary can block or alter the message M being communicated between X and Y entities during the identification session i.

The traceability attack executes as follows [50, 51]:

(i) *Phase 1 (Learning)*. The attacker gathers information related to the tag under attack by implementing *Execute* and *Send* command.

(ii) *Phase 2 (Challenge)*. The attacker is challenged to identify the tag being traced from the set of RFID identifiers.

(iii) *Phase 3 (Guess)*. The attacker continues to gather knowledge through learning phase until it can successfully trace the tag under consideration.

3.3.2. Metaheuristic Model. This model transforms the cryptanalysis of UMAP into a search problem solved with the help of metaheuristic algorithms. The main motivation behind using heuristic search algorithms is their ability to locate global maxima or minima efficiently. The step by step procedure for launching the metaheuristic traceability attack is presented as follows [52]:

(1) The adversary eavesdrops an authentication session between the tag ($Tag\ 0$) and the reader to obtain public messages X, Y.

(2) The attacker initializes the secret values associated with tag (K, n, ID) by using Mersenne Twister pseudorandom number generator. These initialized values act as a seed for simulated annealing algorithm. The adversary then calculates public messages (X', Y') based on assumed secret values.

(3) Simulated Annealing (SA) algorithm is used derive an estimate of (K, n, ID). The values obtained by implementing search algorithm produces public messages that are at minimum deviation from authentic X, Y.

(4) IDS^{i+1} is calculated by using output of simulated annealing algorithm.

(5) Repeat step (2)-(4) to obtain multiple approximations of IDS^{i+1}. Final estimate of dynamic pseudonym is obtained by taking majority vector of all approximations.

(6) For the traceability attack final execution, the attacker is presented with $IDS^{i+1}_{tag\ b} \epsilon \{IDS^{i+1}_{tag\ 0}, IDS^{i+1}_{tag\ 1}\}$. The successful cryptanalysis depends on correct guess of b by the attacker. In order to estimate the value of b, the attacker calculates a correlation function given in (95).

$$corr\left(IDS^{i+1}_{tag\ b}, IDS^{i+1}\right) = \cos\left(IDS^{i+1}_{tag\ b}, IDS^{i+1}\right) = \frac{IDS^{i+1}_{tag\ b} \cdot IDS^{i+1}}{\left|IDS^{i+1}_{tag\ b}\right| \cdot \left|IDS^{i+1}\right|} \tag{95}$$

If the correlation between two values is greater than 75%, the tag presented to the adversary is $tag\ 0$ otherwise it is $tag\ 1$.

3.4. Full Disclosure Attack Model. One of the primary features of a UMAP is provision of confidentiality services to the communicating parties. In this cryptanalysis model, the adversary intercepts the public messages to extract sensitive information related to the tag. The full disclosure attack models can be divided into two subcategories:

3.4.1. Ad Hoc Attacks. The ad hoc cryptanalysis also termed as unstructured attacks explore the protocol's equations to find the mathematical weaknesses. The unstructured attacks exploit the linear behavior of the protocol's operators to estimate the tag's *ID*. Table 11 presents a list of UMAPs primitives which exhibit linear behavior. These operators are not preferred for the UMAP design due to their inability to hide the tag's attributes in public messages.

3.4.2. Structured Attacks. In the structured cryptanalysis models, the adversary follows a predefined set of instructions to breach the confidentiality of an authentication session. The use of probabilistically imbalanced functions as protocol's primitives reveals the tag's information in public messages. Some of the common structured attack models are defined as follows:

(i) *Tango Attack*. The passive tango cryptanalysis is a probabilistic attack which is extremely efficient for recovering the tag's *ID* and other secret information related to a tag. The attack comprises two steps: (1) selection of good approximation (GA) equations and (2) manipulation of derived good approximation equations for disclosing the tag's *ID* under attack. The details of the attack are elaborated as follows [53, 56]:

(I) For selection of GA equation, the attacker locally initializes the tag's *ID* and dynamic variables, who then simulates x UMAP sessions based on the assumed data. The main aim of this step is to obtain a set of GA equations in terms of public parameters $\{IDS, X, Y\}$ for the tag's *ID* estimation. The combinations which exhibits poor diffusion of tag's *ID* are selected as GA equations.
Once a set of GA is derived, the tag *ID* of any identifier implementing the UMAP under consideration can be efficiently calculated using (II).

(II) The idea behind this step is to combine the results of GA equations of α eavesdropped sessions to obtain a single global estimation of *ID* which is highly correlated with tag's original *ID*. The detail procedure of step (II) is elaborated as follows:

(i) Define a matrix Z of size $r\ x\ m$, where

$$r = L\left(\#\ of\ bits\ in\ tag's\ ID\right)$$
$$m = \left(\#\ of\ GA\ equations\right) * \left(\#\ of\ evasdropped\ sessions\right) = x * \alpha \tag{96}$$

(ii) For each eavesdropped session, calculate the values of GA equations and store $L\ bit$

Table 11: Linearized approximations for UMAP's primitives.

Equation	Triangular Approximation	Comments
$a \bigwedge b$	0 ($P = 0.75$)	[53, 54]
$a \vee b$	1 ($P = 0.75$)	[53, 54]
$a \oplus 1$	$\overline{a}$	[46]
$a \oplus 0$	a	[46]
$(1 \vee b) \oplus a$	$\overline{a}$	–
$(0 \wedge b) \oplus a$	a	–
$a + b$	$[a]_i \oplus [b]_i \oplus car_{i-1}$ $[0 < i < (L-1)]$	$car_i = Maj([a]_i, [b]_i, car_{i-1})$ given that $car_0 = 0$ [54]
$a + b$	$[a]_i \oplus [b]_i \oplus car(a, b, i-1)$ $[0 < i < (L-1)]$	$car(a, b, i) = ([a]_i \wedge [b]_i) \vee [([a]_i \oplus [b]_i) \wedge car(a, b, i-1)]$ given that $car(a, b, 0) = 0$ [55]
$a - b$	$[a]_i \oplus [b]_i \oplus bor(a, b, i-1)$ $[0 < i < (L-1)]$	$bor(a, b, i) = ([\overline{a}]_i \wedge [b]_i) \vee [([\overline{a}]_i \oplus [b]_i) \wedge bor(a, b, i-1)]$ given that $bor(a, b, 0) = 0$ [55]
$Rot(a, 0)$	a	[55]
$Rot(a, L)$	a	[55]
$b = Rot(a, c)$	$[b]_0 = [a]_{r(c)}$	$r(c) =$ hamming weight of c or mod of c [55]
$Rot(a \oplus b, a)$	$a \oplus b$	for modular rotation only [54]
$Rot(a, c) \oplus Rot(b, c)$	$Rot(a \oplus b, c)$	[54]

results as a row of matrix Z. Repeat this process for α authentication sessions.

(iii) The estimation of tag's ID is obtained by adding each of columns of matrix Z and returning a zero, if the sum of the said column is below a threshold δ. If the sum is greater than or equal to δ; then one is returned. Formula for calculating δ is as follows:

$$\delta = 0.5 * x * \alpha \tag{97}$$

The success probability of tango attack is directly proportional to the number of simulated session and the number of eavesdropped sessions.

(ii) *Recursive Linear Cryptanalysis*. The Recursive Linear Cryptanalysis (RLC) [54] is applicable to protocols in which the number of secret values associated with the tag under attack is less than or equal to the number of communicating messages per authentication session. This property of protocol makes RLC passive in nature. The attack executes by linearizing the public message encryption equations from a single authentication session in terms of attributes associated with the tag. The linear approximation defines the equation in terms of *XOR* function only. The rules of linear approximation can be derived from Table 11. If the coefficient matrix of linear system of equations is nonsingular and the system is over defined, full disclosure attack can be successfully executed in bit by bit fashion.

(iii) *Recursive Differential Cryptanalysis*. The Recursive Differential Cryptanalysis (RDC) [54] is similar to RLC and it is applied when the number of public messages from a single authentication session is less than the number of variables associated with the tag. The RDC is active in nature. The adversary forces the tag-reader set to communicate on the same set of dynamic variables for every session by blocking the *IDS* and the message Y along with eavesdropping these compromised sessions.
This expands the number of equations from which linearized system of equation can be formed. If the coefficient of resulting system of equations is nonsingular, the secrets related to the tag are successfully revealed. The success probability of RDC depends on the number of sessions that needs to be interfered by the adversary for successful execution.

The above stated attacks are the primary building blocks for the evaluation of CIAA services offered by the UMAPs.

Table 12: Comparative security analysis of UMAPs.

Protocol	Desynchronization Attack	Traceability Attack	Full Disclosure Attack
LMAP	[57]	[58, 59]	[57]
M^2AP	[57]	[60]	[57]
EMAP	[61]	[60]	[61]
SASI	[62]	[63]	[55, 64, 65]
Gossamer's Protocol	[66]	[60]	[67]
Yeh et al.'s Protocol	[34]	[60]	[64, 65, 68]
RAPP	[42, 69]	[45, 70]	[24, 67]
RAPLT	[34]	- -	[71]
RCIA	[34]	[72]	- -
SLAP	[34]	[73]	- -

Table 12 provides a compact security analysis of existing UMAPs and highlights the vulnerability of authentication protocols to multiple cryptanalysis models.

The results of comparative analysis presented in Table 12 emphasize the need to develop a protocol that is computationally efficient and are robust against structured and nonstructured attacks. The design principles for the development of secure authentication protocol are continuously evolving by virtue of weaknesses highlighted by the cryptanalysis reports of existing UMAPs. Following are the design principles that have been deduced through the cryptanalysis of UMAPs discussed in Section 2.

(i) The reader should store n latest values of dynamic variables associated with the tag. The value of n will be directly proportional to strength against desynchronization attack.

(ii) Introduction of an ultralightweight primitive with strong confusion and diffusion capabilities will improve the confidentiality offered by the UMAP.

By incorporating the above-mentioned principles, we can design a UMAP with strong confidentiality, integrity, and availability features. Section 4 presents a novel UMAP termed as Extremely Good Privacy protocol along with the detailed security analysis to prove its ability to provide security and privacy to low-cost IoT nodes.

4. Extremely Good Privacy Protocol

In this section, we propose a new UMAP which requires few on-chip resources and provides Extremely Good Privacy (EGP). The proposed protocol avoids all unbalanced logical operations (such as *AND*; *OR*) and involves only two extremely lightweight operations: *XOR*&*Per* − *XOR*. The new ultralightweight primitive "*Per* − *XOR*" inspired from permutation function (*Per*) introduced in [4], since the later primitive discloses the information of operands, therefore found unsuitable for UMAPs. Moreover, we have also introduced the concept of "inverse permutation" at the tag side which utilizes the permutation function efficiently and with the incorporation of inverse function now it does not require any other primitive to protect its contents. For better understanding of P_x primitive and P_x^{-1}, consider the following.

(A) Computation of Per-XOR (P_x). Suppose m&n are two l-bit strings, where

$$\begin{aligned} m &= m_1 m_2 \dots m_{l,} \\ m_i &\in \{0,1\}, \quad i = 1,2,\dots,l \\ n &= n_1 n_2 \dots n_{l,} \\ n_i &\in \{0,1\}, \quad i = 1,2,\dots,l \end{aligned} \tag{98}$$

The computation of $P_x(m,n)$ involves two following steps:

(1) Permute (transposition) the string $'m'$ according to the string $'n'$, by checking each i^{th} bit of the string $'n'$ (starting from LSB). If $n_i = 0$ then the bit stored at m_i will be placed at m_l location (LSB); otherwise it will be placed at the same position. In the next clock cycle, if $n_{i+1} = 0$ then the bit stored at m_{i+1} will be placed at m_{l-1} location; otherwise it will shifted-left (LSB side). This process will continue till we reach to MSB of string $'n'(n_l)$. After completion, we will have a new string $'m^{*'}$ which is the permuted version of string $'m'$.

(2) Take *XOR* between the new string $'m^{*'}$ and the string $'n'$.

Figure 14 shows the example of Per-XOR computation with reduced bit length.

(B) Computation of Inverse Per-XOR (P_x^{-1}). The tag uses $\boldsymbol{P_x^{-1}}$ primitive extensively in order to retrieve the concealed secrets. The computation of $\boldsymbol{P_x^{-1}}$ also involves two steps:

(1) Take XOR between the received $P_x(m,n)$ and preshared secret n.

(2) Perform inverse permutation in a sequential manner to get the concealed string $'m'$. For inverse permutation, we use one pointer/indexer (X) for traversing over the result computed in step-1. If $n_i = 0$ then the pointer X moves to m_l position and bit stored on lth

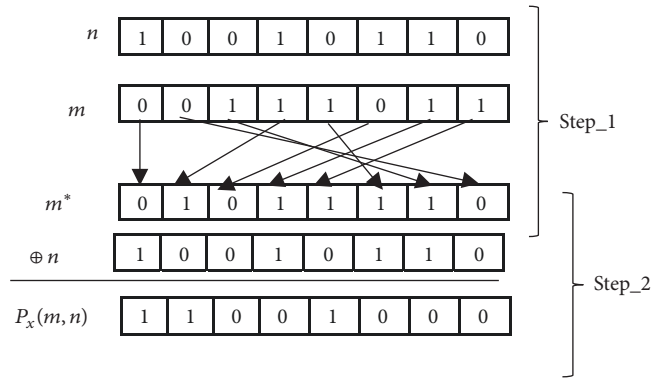


Figure 14: Computation (example) of Per-XOR (P_x).

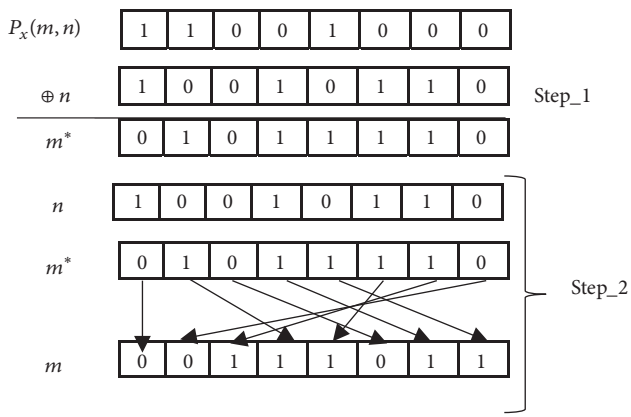


Figure 15: Computation (example) of inverse Per-XOR (P_x^{-1}).

Table 13: Notations used in EGP.

Symbol	Meaning
$P_x(a, b)$	Per-XOR b with a
$P_x^{-1}(a, b)$	Inverse Per-XOR b with a
$\bigoplus$	XOR
$\|$	Concatenation
K	Key
$'\mathscr{R}'$	Reader
$'\mathscr{T}'$	Tag

location on string m will be placed at m_1 location; otherwise the pointer X moves to m_i position and bit stored on ith location will be placed m_1 location. This process will continue till the last bit of the string $'n'(n_l)$.

For better understanding, Figure 15 shows the example of Inverse Per-XOR (P_x^{-1}) with reduced bit length.

4.1. Working of the EGP Protocol. Figure 16 shows the detailed working of the protocol. The EGP protocol involves three main components: Tag ($\mathscr{T}$), Reader ($\mathscr{R}$), and the backend database ($\mathscr{D}$). Each $'\mathscr{T}'$ contains the one static secret ID; two sets of IDS and keys (Old and new). To avoid the possible desynchronization attacks, the $'\mathscr{R}'$ uses the buffer-based security framework proposed in [46]. In the buffer-based security framework, the reader maintains a dynamic memory architecture and stores all previous pseudonyms and keys (depending upon buffer size). To avoid buffer overflow, a RTC (Real Time Clock) has also been integrated at the reader side that manages the storage of variable. The basic symbols and notations used in this protocol are presented in Table 13.

The specifications of the protocol are as follows:

(1) The $'\mathscr{R}'$ initiates the protocol session by sending "*hello*" message towards the $'\mathscr{T}'$.

(2) Upon receiving of this query, the $'\mathscr{T}'$ responds with its current IDS.

(3) The $'\mathscr{R}'$ looks for the received IDS in its database and if a match occurs then it computes A, B, and C messages and sends to the reader. Otherwise, it will send another "*hello*" towards $'\mathscr{T}'$ and repeat the same process of finding matched entry. If the $'\mathscr{R}'$ does not find the matched entry in this second round, it will terminate the protocol session.

(4) On receiving of A, B, and C messages, the $'\mathscr{T}'$ performs following three tasks:

(a) Extract random nonce (n_1, n_2) from messages $A\&B$:

$$n_1 = P_x^{-1}(A, K) \tag{99}$$

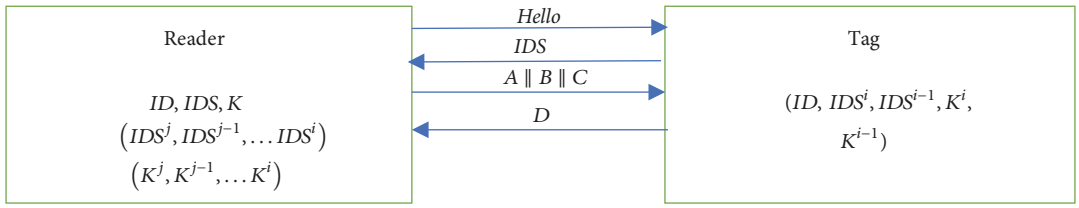
$$n_2 = P_x^{-1}(B, K \oplus n_1) \tag{100}$$

(5) Compute the local value of C and compare it with the received C. If both of the values coincide, only then the $'\mathscr{T}'$ authenticates $'\mathscr{R}'$; otherwise it will terminate the protocol session with the particular $'\mathscr{R}'$. After successful authentication, the $'\mathscr{T}'$ computes the message D and transmits towards $'\mathscr{R}'$.

(6) Update IDS and key (K^{next}).

(7) Upon receiving of message D, the $'\mathscr{R}'$ computes a local value of D and compares it with the received one. If both values coincide then the $'\mathscr{R}'$ authenticates $'\mathscr{T}'$ and updates IDS and *key* for the particular $'\mathscr{T}'$ in its database for future correspondence.

Figure 16 presents the block diagram of the EGP protocol.



$A = P_x(n_1, K)$	$C = P_x(K^* \oplus n_2 \oplus n_1, K)$
$B = P_x(n_2, K \oplus n_1)$	$D = P_x(K^* \oplus ID \oplus n_1, n_2)$
$K^* = P_x(K \oplus n_2, n_1)$	$K^i = K^*$
$IDS^i = P_x(IDS \oplus n_2, n_1 \oplus n_2)$	

Figure 16: Block diagram of EGP protocol.

4.2. Security Analysis of EGP Protocol. We analyze the security of the proposed protocol in two aspects: formal verification and resistance of the protocol against rudimentary attacks. The detailed description of the security analysis is presented as follows.

4.2.1. Formal Security Analysis. For formal security verification, we use GNY logic. The formal analysis tool uses abstract language and verifies the assumptions and objectives of the security algorithms. The detailed description of the analysis is as follows.

(1) GNY Logic Analysis. GNY (Gong–Needham–Yahalom) logic is a mathematical formal verification tool that verifies the security assumptions and goals of security algorithms. The GNY logic is a multistep process which first translates the assumptions and public messages in abstract language and then starts validating goals. For validation of the goals, the GNY logic uses three rules: Being Informed, Possession, and Freshness Rules.

(a) *Being Informed Rules.* Any formula that $\mathcal{M}$ receives is considered as "being informed".
$\mathcal{I}_1$: if $\mathcal{M}$ is informed the formula $\mathcal{N}$, which he did not convey in this run, then $\mathcal{M}$ is informed $\mathcal{N}$

$$\mathcal{I}_1: \frac{\mathcal{M} \triangleleft {}^*\mathcal{N}}{\mathcal{M} \triangleleft \mathcal{N}} \tag{101}$$

$\mathcal{I}_2$: if $\mathcal{M}$ is informed an encrypted formula with symmetric key (K) then $\mathcal{M}$ is informed the formula.

$$\mathcal{I}_2: \frac{\mathcal{M} \triangleleft \{\mathcal{N}\}_K, \mathcal{M} \ni \mathcal{N}}{\mathcal{M} \triangleleft \mathcal{N}} \tag{102}$$

(b) *Possession Rules.* If $\mathcal{M}$ possesses a formula then it can possess other associated formulae as well.
$\mathcal{P}_1$: $\mathcal{M}$ can possess any variable which is being informed

$$\mathcal{P}_1: \frac{\mathcal{M} \triangleleft \mathcal{N}}{\mathcal{M} \ni \mathcal{N}} \tag{103}$$

$\mathcal{P}_2$: if $\mathcal{M}$ possess two different formulae then it can possess their concatenation and functions as well.

$$\mathcal{P}_2: \frac{\mathcal{M} \ni \mathcal{N}, \mathcal{M} \ni \mathcal{O}}{\mathcal{M} \ni (\mathcal{N}, \mathcal{O}), \mathcal{M} \ni \mathcal{F}(\mathcal{N}, \mathcal{O})} \tag{104}$$

Table 14 enlists the notations used in GNY logic analysis.

(c) *Freshness Rules.* On the basis of belief, $\mathcal{M}$ determines the freshness of messages.
$\mathcal{F}_1$: if $\mathcal{M}$ believes that a formula is fresh then he also believes that any concatenation and function will be fresh.

$$\mathcal{F}_1: \frac{\mathcal{M}| \equiv \#(\mathcal{N})}{\mathcal{M}| \equiv \#(\mathcal{N}, \mathcal{O}), \mathcal{M}| \equiv \#\mathcal{F}(\mathcal{N}, \mathcal{O})} \tag{105}$$

(2) Formal Proof of EGP Using GNY Logic. The first step in GNY analysis is to describe the assumptions of the protocols which are followed by the formalization of the exchanged messages. Finally, the goals of the protocols are verified using formal analysis postulates.

The authentication process mainly depends upon the pseudorandom numbers; therefore we apply analysis on first two messages only.

The messages can be formulated as follows:

$$\begin{aligned} &\mathcal{T} \triangleleft * \{n_1\}_K \sim> \mathcal{R}| \equiv \mathcal{T} \overset{K}{\longleftrightarrow} \mathcal{R} \\ &\mathcal{T} \triangleleft * \{n_2\}_{n_1,K} \sim> \mathcal{R}| \equiv \mathcal{T} \overset{n_1,K}{\longleftrightarrow} \mathcal{R} \end{aligned} \tag{106}$$

The goal of sending such messages is

$$\mathcal{G}_1: \mathcal{T}| \equiv \mathcal{R} \ni (n_1, n_2) \tag{107}$$

By applying the verification postulates, we can validate EGP's goal as follows:

$$\begin{aligned} &\mathcal{T} \triangleleft * \{n_1\}_K \sim> \mathcal{R}| \equiv \mathcal{T} \overset{K}{\longleftrightarrow} \mathcal{R} \\ &\mathcal{IS}_1: \mathcal{T} \triangleleft {}^* \{n_1\}_K \\ &\left(\text{Since } \mathcal{R}| \equiv \mathcal{T} \overset{K}{\longleftrightarrow} \mathcal{R} \text{ is already assumed to be satisfied}\right) \end{aligned} \tag{108}$$

Table 14: Notations used in GNY logic analysis.

S.No.	Statement/Notations	Interpretation
1	$\mathcal{M}$	Manager/Administrator
2	$\mathcal{N}$	Message/variable
3	$\mathcal{R}$	Reader
4	$\mathcal{T}$	Tag
5	$\mathcal{IS}$	Intermediate Step
6	$\mathcal{M} \lhd *\mathcal{N}$	$\mathcal{M}$ is informed $\mathcal{N}$, $\mathcal{M}$ didn't convey in current session
7	$\mathcal{M} \lhd \mathcal{N}$	$\mathcal{M}$ is informed $\mathcal{N}$
8	$(\mathcal{N} \parallel \mathcal{O})$	Concatenation of $\mathcal{N}$&O
9	$\mathcal{M}\vert \equiv \#(\mathcal{N})$	$\mathcal{M}$ believes that $\mathcal{N}$is fresh
10	$\mathcal{M}\vert \equiv \varphi(\mathcal{N})$	$\mathcal{M}$ believes that $\mathcal{N}$is computable/recognizable
11	$\mathcal{M}\vert \equiv \mathcal{N}$	$\mathcal{M}$ believes that $\mathcal{N}$ holds
12	$\mathcal{M}\vert \equiv \mathcal{M} \overset{K}{\leftrightarrow} \mathcal{Z}$	$\mathcal{M}$ believes that $K$is an appropriate secret for $\mathcal{Z}$
13	$\mathcal{M} \ni \mathcal{N}$	$\mathcal{M}$ can possess $\mathcal{N}$
14	$\mathcal{M}\vert \sim \mathcal{N}$	$\mathcal{M}$ once conveyed $\mathcal{N}$
15	$\{\mathcal{N}\}_K$	$\mathcal{N}$is encrypted using symmetric key K

By considering $\mathcal{I}_1, \mathcal{S}_4$, and $\mathcal{I}_2$ we can have

$$\begin{aligned} &\mathcal{IS}_2\colon \mathcal{T} \lhd \{n_1\}_K \\ &\mathcal{IS}_2\colon \mathcal{T} \lhd n_1 \end{aligned} \tag{109}$$

According to rule $\mathcal{P}_1$, we get

$$\mathcal{IS}_3\colon \mathcal{T} \ni n_1 \tag{110}$$

Now for second message as we know,

$$\begin{aligned} &\mathcal{T} \lhd * \{n_2\}_{n_1,K} \sim> \mathcal{R}\vert \equiv \mathcal{T} \overset{n_1,K}{\longleftrightarrow} \mathcal{R} \\ &\mathcal{IS}_4\colon \mathcal{T} \lhd^* \{n_2\}_{n_1,K} \quad \left(\text{Since } \mathcal{IS}_3 \text{ and } \mathcal{R}\vert \equiv \mathcal{T} \overset{K}{\leftrightarrow} \mathcal{R} \text{ is already assumed to be satisfied}\right) \end{aligned} \tag{111}$$

If we consider $\mathcal{I}_1, \mathcal{S}_4$, and $\mathcal{I}_2$ then $\mathcal{IS}_4$ can be represented as

$$\begin{aligned} &\mathcal{IS}_5\colon \mathcal{T} \lhd \{n_2\}_{n_1,K} \\ &\mathcal{IS}_6\colon \mathcal{T} \lhd n_2 \end{aligned} \tag{112}$$

Further $\mathcal{IS}_6$ can be interpreted as

$$\mathcal{IS}_7\colon \mathcal{T} \ni n_2 \tag{113}$$

Hence from $\mathcal{IS}_3$ and $\mathcal{IS}_7$, it can be observed that EGP optimally achieves its goal:

$$\mathcal{T}\vert \equiv \mathcal{R} \ni (n_1, n_2) \tag{114}$$

If adversary tries to modify n_1, then effect of this alteration directly transfers to n_2 as well. The tag will not verify message C, hence abort such protocol sessions, and will remain synchronized.

4.3. Desynchronization Attack. The desynchronization attacks presented in [34] force the legitimate readers and the tags to update different pair of pseudonyms and keys and therefore make the resources unavailable for the legitimate parties. To avoid such availability and desynchronization attacks, the EGP protocol uses dynamic memory architecture at the reader's side. This memory architecture involves RTC (Real Time Clock) and Shift Registers to store the current and previous values of pseudonyms and keys of each associated tag. The memory architecture is located at the reader's side and therefore does not increase the cost of the tag. If the adversary tries to block some genuine authentication sessions and uses replay attack models to desynchronize the EGP' tag and the reader then this will be impossible for the adversary, since the reader keeps the records of n authentication sessions. The size of dynamic variable buffer at the reader's side primarily depends on the architecture of the database associated with the network. The increase in buffer size enhances the synchronization of the tag/reader pair at the cost of increased memory requirement at the reader's side.

4.4. Traceability Attack. In EGP if an adversary tries to find the conjuncture ID through publically disclosed messages then because of optimal messages structure, she will get only ambiguous equation:

$$ID = P_x^{-1}(D, P_x^{-1}(C, K) \oplus p_x^{-1}(B, K \oplus P_x^{-1}(A, K)) \tag{115}$$

By keeping in view of computational complexity of (115), it will be almost impossible for an adversary to track the individual tag by resolving this equation. Moreover, most of the variables involved in EGP will get update after each authentication session; therefore EGP proves to be secure against all existing traceability attack models.

4.5. Full Disclosure Attack. The full disclosure attacks exploit the inherent weaknesses of the *T-functions*. The attackers usually perform different computational operations on public messages and try to obtain conjecture secret values. However, the inclusion of nontriangular primitive *Per − XOR* and *inverse Per − XOR* makes EGP protocol almost impossible to retrieve the concealed secret from public messages. The protocol's performance to unstructured and structured full disclosure attack is described as follows:

(A) *Ad Hoc Attack*. The ad hoc attacks target the lack of randomness in public messages and the linear behavior of the protocol's primitives. In EGP protocol, the structure of the public messages is designed to avoid the previously presented unstructured full disclosure attacks. Every public message increments the degree of randomness by one; i.e., message A consists of random number n_1, message B consists of n_2 and n_1, and message C consists of K^*, n_2, and n_1. Even if the adversary keeps the dynamic variables constant for multiple sessions by blocking the message D, the values of n_1, n_2, and K^* change for every session and hence the values of public messages vary making it theoretically impossible to derive the tag's ID by just eavesdropping the authentication sessions.

In addition to this, the *Per − XOR* operator provides improved confusion and diffusion services to the public messages due to the following features:

(1) The *Xor* operation masks the result of permutation making it impossible to reveal the LSB or MSB of the first operand of *Per − Xor* function without complete information of the second operand.

(2) The operands of the *Per − Xor* function in the EGP protocol are the irreversible combination of dynamic variables. The analysis of message $D(P_x(K^* \oplus ID \oplus n_1, n_2))$ shows that even if the adversary obtains $K^* \oplus ID \oplus n_1$ by exploiting the reversable nature of *Per − Xor*, the ID cannot be retrieved without the knowledge of K^* and n_1. This enables the presented primitive to effectively conceal the secret values associated with the session in public messages.

In [45], an ad hoc full disclosure attack on the RAPP protocol is presented. In the proposed attack, the dynamic variables are kept constant by blocking the last message from the $\mathscr{R}$ to the $\mathscr{T}$ and then the weakness of permutation primitive is exploited to obtain the random number n_1 that eventually leads to disclosure of the tag's ID. For the estimation of single random number, the adversary generates a database of two public messages consisting of constants IDS, K_1, K_2, and K_3.

In EGP protocol, the dynamic variables on the tag's side can be kept constant by blocking message D from the $\mathscr{T}$ to the $\mathscr{R}$. This leaves the adversary with only three public messages (A, B, C) based on constant IDS and K. Since the EGP protocol uses two random numbers (n_1, n_2), the number of public messages is not sufficient to estimate the private keys generated by the $\mathscr{R}$.

Therefore, due to nonlinear behavior of *Per − Xor* function and small number of public messages, the ad hoc attacks proposed for RAPP protocol are not applicable to the EGP protocol.

(B) *Tango Attack*. The tango attack proves to be unsuccessful against nontriangular based UMAPs. The inventors of the tango attack also highlighted this inherent weakness of the attack model. In EGP, we have extensively used nontriangular primitives (*Per − XOR* and *Inverse Per − XOR*) in its design, which requires extensive computational complexity $O(8 \times 2^K \times log_2(K!) \times log_2(n_1!) \times \log(n_2!)$ to retrieve ID. Therefore, it is almost impossible for an adversary to find the optimal GA equations and apply tango attack model on EGP.

(C) *Recursive Linear and Differential Cryptanalysis*. The RLC model exploits the weak diffusion properties of the protocols and uses the public messages to construct the set of linear equations for each individual bit of the concealed secrets. After constructing the sufficient equations, the adversary solves the equations recursively and tries to get the concealed secrets bit by bit. However, the incorporation of (optimal) nontriangular primitives such as recursive hash [41], Psuedo-Kasami codes [31], and *Per − XOR* in protocol messages makes it almost impossible for an adversary (with RLC) to construct enough equations that may disclose the concealed secrets.

On the other hand, RDC model is more powerful and requires an active attacker which can block the genuine authentication sessions (between the reader and the tags) and hence both the legitimate readers and the tag communicate with the previous pseudonyms and keys. The attacker then tries to find the differential relationship between the random nonce and finds conjecture secrets. However, RDC also fails to disclose the concealed secrets of nontriangular based UMAPs. This inventor of RLC and RDC also highlighted this inherent limitation of the models.

The cryptanalysis proves that the EGP protocol is robust against all the attack models presented in Section 3. None of the previous UMAPs (discussed here) can withstand all types of existing adversarial models discussed in the security analysis model which make them unsuitable for real world applications. On the other, the evaluation of EGP protocol

based on robustness, reliability, and security proves that the presented authentication protocol is most suitable for the authentication of resource constraint IoT network perception layer.

5. Conclusion

The 5th generation mobile communication systems are envisioned to offer high-speed broadband service which is a key enabling factor for the development in the field of the IoT networks. The security and privacy of the IoT network are of utmost concern since a large amount of user-specific data is being generated on a real-time basis. The identity verification of the communicating parties is a primary part of the secure perception layer. The resource constraint IoT networks use ultralightweight protocols for the node authentication. This paper presents a brief survey on the existing UMAPs and their cryptanalysis models. The UMAPs can be broadly classified into three categories based on the primitives used for the challenge/response message calculation, i.e., UMAPs with triangular functions, UMAPs with single nontriangular function, and UMAP with hybrid nontriangular functions. The hybrid nontriangular functions provide enhanced confidentiality, integrity, availability, and authentication (CIAA) services at the cost of increased gate equivalents. However, the literature review shows that almost all the existing UMAPs are vulnerable to multiple cryptanalysis attacks, i.e., desynchronization attack, full disclosure attack, and traceability attacks. In this paper, we have proposed a new ultralightweight authentication protocol named EGP (Extremely Good Privacy) for IoTs. The proposed protocol introduced a new ultralightweight primitive, Per-XOR which is composed of two extremely lightweight operations: XOR and permutation. This newly proposed primitive increases the confusion and diffusion properties of the public messages optimally and avoids all the existing adversarial models. The performance comparison of the EGP protocol shows that it outperforms compared to its contending UMAPs in terms of security. This remarkable feature makes EGP the best choice for extremely low-cost IoTs sensors and RFID tags.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Singh, G. Tripathi, and A. J. Jara, "A survey of internet-of-things: future vision, architecture, challenges and services," in *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 287–292, IEEE, Seoul, South Korea, March 2014.

[2] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[3] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on internet of things: architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[4] Y. Tian, G. Chen, and J. Li, "A new ultralightweight RFID authentication protocol with permutation," *IEEE Communications Letters*, vol. 16, no. 5, pp. 702–705, 2012.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[6] A. P. Hsu, W. Lee, A. J. Trappey, C. V. Trappey, and A. Chang, "Using system dynamics analysis for performance evaluation of IoT enabled one-stop logistic services," in *Proceedings of the 2015 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 1291–1296, IEEE, Kowloon, October 2015.

[7] L. B. Campos, C. E. Cugnasca, A. R. Hirakawa, and J. S. Martini, "Towards an IoT-based system for smart city," in *Proceedings of the 2016 IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 129-130, Sao Paulo, Brazil, September 2016.

[8] A. F. Harris, V. Khanna, G. S. Tuncay, and R. H. Kravets, "Smart LaBLEs: Proximity, Autoconfiguration, and a constant supply of gatorade(TM)," in *Proceedings of the 1st IEEE/ACM Symposium on Edge Computing, SEC 2016*, pp. 142–154, IEEE, USA, October 2016.

[9] M. Darianian and M. P. Michael, "Smart home mobile RFID-based internet-of-things systems and services," in *Proceedings of the International Conference on Advanced Computer Theory and Engineering (ICACTE '08)*, pp. 116–120, IEEE, December 2008.

[10] J. Dedy Irawan, E. Adriantantri, and A. Farid, "RFID and IOT for attendance monitoring system," in *Proceedings of the 3rd International Conference on Electrical Systems, Technology and Information, ICESTI 2017*, EDP Sciences, Indonesia, September 2017.

[11] A. M. Wicks, J. K. Visich, and S. Li, "Radio frequency identification applications in hospital environments," *Hospital Topics*, vol. 84, no. 3, pp. 3–9, 2006.

[12] M. McGee, "Health-care IT has a new face," *Information Week*, vol. 988, p. 16, 2004.

[13] A. Aguilar, W. Van Der Putten, and F. Kirrane, "Positive patient identification using RFID and wireless networks," in *Proceedings of the in HISI 11th Annual Conference and Scientific Symposium*, 2006.

[14] J. Dalton, C. Ippolito, I. Poncet, and S. Rossini, "Using RFID technologies to reduce blood transfusion errors," White Paper by Intel Corporation, Autentica, Cisco systems and San Raffaele Hospital, 2005.

[15] A. G. Kulkarni, A. K. N. Parlikad, D. C. McFarlane, and M. Harrison, "Networked RFID systems in product recovery management," in *Proceedings of the 2005 IEEE International Symposium on Electronics and the Environment, 2005*, IEEE, New Orleans, LA, USA, 2005.

[16] Z. Zhang, "Hierarchical multi-reader RFID systems for Internet-of-Things, 2010, US-AB".

[17] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and near-Field Communication*, John Wiley & Sons, 2010.

[18] C. Bolan, "A review of the electronic product code standards for RFID technology," in *Proceedings of the 7th International Network Conference, INC 2008*, pp. 171–178, UK, July 2008.

[19] M. Safkhani and N. Bagheri, "Passive secret disclosure attack on an ultralightweight authentication protocol for Internet of Things," *The Journal of Supercomputing*, vol. 73, no. 8, pp. 3579–3585, 2017.

[20] R. Baashirah and A. Abuzneid, "Survey on prominent RFID authentication protocols for passive tags," *Sensors*, vol. 18, no. 10, p. 3584, 2018.

[21] H.-Y. Chien, "SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 337–340, 2007.

[22] G. Gódor and S. Imre, "Elliptic curve cryptography based authentication protocol for low-cost RFID tags," in *RFID-Technologies and Applications (RFID-TA), 2011 IEEE International Conference on*, vol. 14, IEEE, 2011.

[23] Y.-l. Liu, C. Wang, X. l. Qin, and B. Li, "A lightweight RFID authentication protocol based on elliptic curve cryptography," *Journal of Computers*, vol. 8, no. 11, 2013.

[24] W. Shao-hui, H. Zhijie, L. Sujuan, and C. Dan-wei, *Security Analysis of RAPP an RFID Authentication Protocol Based on Permutation*, College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, 2012.

[25] C. Jin, C. Xu, X. Zhang, and J. Zhao, "A secure RFID mutual authentication protocol for healthcare environments using elliptic curve cryptography," *Journal of Medical Systems*, vol. 39, no. 3, pp. 1–8, 2015.

[26] Z. Zhang and Q. Qi, "An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography," *Journal of Medical Systems*, vol. 38, no. 5, pp. 1–7, 2014.

[27] Y. K. Lee, L. Batina, and I. Verbauwhede, "EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol," in *Proceedings of the 2008 IEEE International Conference on RFID (Frequency Identification), IEEE RFID 2008*, pp. 97–104, IEEE, USA, April 2008.

[28] Y. P. Liao and C. M. Hsiao, "A Secure ECC-Based RFID Authentication Scheme Using Hybrid Protocols," in *Advances in Intelligent Systems and Applications - Volume 2*, vol. 2 of *Smart Innovation, Systems and Technologies*, pp. 1–13, Springer, Berlin, Heidelberg, Germany, 2013.

[29] A. Tewari and B. B. Gupta, "Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags," *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1085–1102, 2017.

[30] H. Luo, G. Wen, J. Su, and Z. Huang, "SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system," *Wireless Networks*, vol. 24, no. 1, pp. 69–78, 2018.

[31] U. Mujahid, M. Najam-ul-Islam, and S. Sarwar, "A new ultra-lightweight RFID authentication protocol for passive low cost tags: KMAP," *Wireless Personal Communications*, vol. 94, no. 3, pp. 725–744, 2017.

[32] G. EPCglobal, *EPC radio-frequency identity protocols generation-2 UHF RFID; specification for RFID air interface protocol for communications at 860 MHz960 MHz*, EPCglobal Inc, November 2013.

[33] C. Rolfes et al., "Security for 1000 Gate Equivalents".

[34] M. Safkhani and N. Bagheri, "Generalized desynchronization attack on UMAP: application to RCIA, KMAP, SLAP and SASI+ protocols," *IACR Cryptology ePrint Archive*, p. 905, 2016.

[35] P. Peris-Lopez, J. Hernandez-Castro, J. E. Tapiador, and A. Ribagorda, "LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags," in *Proceedings of the 2nd Workshop on RFID Security*, 2006.

[36] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "M^2AP: a minimalist mutual-authentication protocol for low-cost RFID tags," in *Proceedings of the in International conference on ubiquitous intelligence and computing*, vol. 2, Springer, 2006.

[37] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "EMAP: an efficient mutual-authentication protocol for low-cost RFID tags," in *Proceedings of the OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, vol. 4277 of *Lecture Notes in Computer Science*, pp. 352–361, Springer, 2006.

[38] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda, "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol," in *Proceedings of the in International Workshop on Information Security Applications*, Springer, 2008.

[39] K.-H. Yeh, N. Lo, and E. Winata, "An efficient ultralightweight authentication protocol for RFID systems," *Radio Frequency Identification System Security*, vol. 4, no. 10, pp. 49–60, 2010.

[40] B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 140–147, ACM, April 2008.

[41] U. Mujahid, M. Najam-Ul-Islam, and M. A. Shami, "RCIA: a new ultralightweight RFID authentication protocol using recursive hash," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 642180, 2015.

[42] U. Mujahid and M. Najam-ul-islam, "Pitfalls in ultralightweight RFID authentication protocol," *International Journal of Communication Networks and Information Security*, vol. 7, no. 3, pp. 169–176, 2015.

[43] G. Avoine and X. Carpent, "Yet another ultralightweight authentication protocol that is broken," in *Proceedings of the International Workshop on Radio Frequency Identification: Security and Privacy Issues*, Lecture Notes in Computer Science, Springer, 2012.

[44] H.-M. Sun, W.-C. Ting, and K.-H. 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.

[45] N. Bagheri, M. Safkhani, P. Peris-Lopez, and J. E. Tapiador, "Weaknesses in a new ultralightweight RFID authentication protocol with permutation-RAPP," *Security and Communication Networks*, vol. 7, no. 6, pp. 945–949, 2014.

[46] M. Khalid and U. Mujahid, "Security framework of ultra-lightweight mutual authentication protocols for low cost RFID tags," in *Proceedings of the 2017 International Conference on Communication, Computing and Digital Systems, C-CODE 2017*, pp. 26–31, IEEE, Pakistan, March 2017.

[47] M. Khalid, U. Khokhar, and M. Najam-ul-Islam, "Advance strong authentication strong integrity (ASASI) protocol for low cost radio frequency identification," in *Proceedings of the 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, pp. 1–6, IEEE, Shah Alam, July 2018.

[48] R. C.-W. Phan, "Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 4, pp. 316–320, 2009.

[49] A. Juels and S. A. Weis, "Defining strong privacy for RFID," in *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pp. 342–347, White Plains, NY, USA, March 2007.

[50] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and J. C. A. van der Lubbe, "Security flaws in a recent ultralightweight RFID protocol," https://arxiv.org/pdf/0910.2115.pdf, 2009.

[51] S. S. S. GhaemMaghami, A. Haghbin, and M. Mirmohseni, "Traceability Improvements of a New RFID Protocol Based On EPC C1G2," *IACR Cryptology ePrint Archive*, vol. 2015, Article ID 872, 2015.

[52] J. C. Hernandez-Castro, J. M. Estevez-Tapiador, P. Peris-Lopez, J. A. Clark, and E. Talbi, "Metaheuristic traceability attack against SLMAP, an RFID lightweight authentication protocol," *International Journal of Foundations of Computer Science*, vol. 23, no. 02, pp. 543–553, 2012.

[53] J. C. Hernandez-Castro, P. Peris-Lopez, R. C.-W. Phan, and J. M. E. Tapiador, "Cryptanalysis of the David-Prasad RFID ultralightweight authentication protocol," in *Proceedings of the 6th International Workshop on Radio Frequency Identification: Security and Privacy Issues (RFID '10)*, pp. 22–34, Springer, Istanbul, Turkey, 2010.

[54] Z. Ahmadian, M. Salmasizadeh, and M. R. Aref, "Recursive linear and differential cryptanalysis of ultralightweight authentication protocols," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1140–1151, 2013.

[55] G. Avoine, X. Carpent, and B. Martin, "Strong authentication and strong integrity (SASI) is not that strong," in *International workshop on radio frequency identification: security and privacy issues*, Lecture Notes in Computer Science, pp. 50–64, Springer, Berlin, Germany, 2010.

[56] D. F. Barrero, J. C. Hernández-Castro, P. Peris-Lopez, D. Camacho, and M. D. R-Moreno, "A genetic tango attack against the David-Prasad RFID ultra-lightweight authentication protocol," *Expert Systems with Applications*, vol. 31, no. 1, pp. 9–19, 2014.

[57] T. Li and G. Wang, "Security analysis of two ultra-lightweight RFID authentication protocols," in *Proceedings of the IFIP International Information Security Conference*, vol. 232, pp. 109–120, Springer, New York, NY, USA, 2007.

[58] K. Ouafi and R. C.-W. Phan, "Privacy of recent RFId authentication protocols," in *Information security practice and experience*, vol. 4991 of *Lecture Notes in Comput. Sci.*, pp. 263–277, Springer, Berlin, Germany, 2008.

[59] M. Safkhani, N. Bagheri, M. Naderi, and S. K. Sanadhya, "Security analysis of LMAP ++, an RFID authentication protocol," in *Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions, ICITST 2011*, pp. 689–694, IEEE, UAE, December 2011.

[60] M. Umar, *Khokhar, Ultralightweight Cryptography for low cost passive RFID [Ph.D. thesis]*, Bahria University, July 2016, Tags PhD dissertation.

[61] L. Tieyan and D. Robert, "Vulnerability analysis of EMAP-an efficient RFID mutual authentication protocol," in *Proceedings of the 2nd International Conference on Availability, Reliability and Security (AReS '07)*, pp. 238–245, IEEE, Vienna, Austria, April 2007.

[62] T. Cao, E. Bertino, and H. Lei, "Security analysis of the SASI protocol," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 1, pp. 73–77, 2009.

[63] J. C. Hernandez-Castro, J. M. E. Tapiador, P. Peris-Lopez, and J.-J. Quisquater, "Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations," https://arxiv.org/pdf/0811.4257.pdf.

[64] G. Avoine, X. Carpent, and B. Martin, "Privacy-friendly synchronized ultralightweight authentication protocols in the storm," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 826–843, 2012.

[65] U. Mujahid, "probabilistic recursive cryptanalysis of ultralightweight mutual authentication protocols for passive RFID systems," *Pakistan Journal of Engineering and Applied Sciences*, 2016.

[66] Z. Bilal, A. Masood, and F. Kausar, "Security analysis of ultralightweight cryptographic protocol for low-cost RFID tags: gossamer protocol," in *Proceedings of the 12th International Conference on Network-Based Information Systems*, pp. 260–267, IEEE, Indianapolis, Ind, USA, August 2009.

[67] E. Taqieddin and J. Sarangapani, "Vulnerability analysis of two ultra-lightweight RFID authentication protocols: RAPP and Gossamer," in *Proceedings of the 7th International Conference for Internet Technology and Secured Transactions, ICITST 2012*, pp. 80–86, IEEE, UK, December 2012.

[68] P. Peris-Lopez, J. C. Hernandez-Castro, R. C.-W. Phan, J. M. E. Tapiador, and T. Li, "Quasi-linear cryptanalysis of a secure RFID ultralightweight authentication protocol," in *International Conference on Information Security and Cryptology*, pp. 427–442, Springer, Berlin, Heidelberg, 2010.

[69] X. Zhuang, Z.-H. Wang, C.-C. Chang, and Y. Zhu, "Security analysis of a new ultra-lightweight RFID protocol and its improvement," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 3, pp. 166–177, 2013.

[70] N. Bagheri, M. Safkhani, P. Peris-Lopez, and J. E. Tapiador, "Cryptanalysis of RAPP, an RFID authentication protocol," *IACR Cryptology ePrint Archive*, p. 702, 2012.

[71] S. Wang, S. Liu, and D. Chen, "Security analysis and improvement on two RFID authentication protocols," *Wireless Personal Communications*, vol. 82, no. 1, pp. 21–33, 2015.

[72] S. A. Yasear, N. H. Zakaria, and M. N. Omar, "Enhancing the security of RCIA ultra-lightweight authentication protocol by using Random Number Generator (RNG) technique," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 1-2, pp. 77–80, 2017.

[73] K. Baghery, B. Abdolmaleki, S. Khazaei, and M. R. Aref, "Breaking anonymity of some recent lightweight RFID authentication protocols," *Wireless Networks*, vol. 25, no. 3, pp. 1235–1252, 2019.