International Journal of Information Security (2025) 24:202
https://doi.org/10.1007/s10207-025-01122-z

REGULAR CONTRIBUTION

Breaking the Ultralightweight RFID Mutual Authentication Protocol: Confidentiality Under Threat

Parsa Riaz[1] · Madiha Khalid[1] · Mehdi Hussain[1] · Naveed Riaz[1] · Umar Mujahid[2] · Muhammad Najam-ul-islam[3]

Received: 18 February 2025 / Accepted: 26 August 2025

Abstract

Radio Frequency Identification (RFID) tags collect real-time location information in IoT-based asset-tracking applications, rendering security and privacy crucial. Recently, an Ultralightweight RFID Mutual Authentication Protocol (URMAP) has been proposed, claiming to ensure confidentiality, integrity, and availability of tag/reader pair. This paper challenges the confidentiality claim of the protocol by demonstrating three secret-disclosure attack models. Probabilistic tango cryptanalysis extracts the tag's identification information with an average success rate of 84.375%. In addition, a functional attack is executed mainly by exploiting the unbalanced nature of encryption primitives, i.e., bit-wise AND and OR operators, to receive all of the tag's attributes deterministically. Grover's search-based brute force attack challenges the quantum resilience of the protocol by retrieving attributes associated with all tags within the identification network. To mitigate these vulnerabilities, an enhanced protocol, URMAP^{+}, is proposed, which retains the strengths of URMAP while addressing its security flaws. As a future direction, this article advocates for a paradigm shift toward post-quantum ultralightweight ciphers.

Keywords Authentication · Brute force · Confidentiality · Full disclosure attack · Grover's search · RFID · Security analysis

1 Introduction

The Internet of Things (IoT) is a comprehensive network of intelligent objects capable of auto-organizing, sharing information/ resources, and reacting to environmental changes under observation. As per the statistics of 2023, the IoT-based asset tracking applications market is worth USD 4.5 billion, and this application set is estimated to register a Compound Annual Growth Rate (CAGR) of over 12.5% between 2024 and 2032 [1]. Although the enabling technologies for such applications are bar codes, RFID tags, and QR-Codes, RFID tags have been heavily endorsed by organizations like IETF [2], ISO [3], and NIST [4].

The RFID system comprises the tag, the reader, and the database. The tag is attached to the asset and transmits a unique ID to the reader for identification. The database facilitates the reader with asset tracking as it contains information on all the tags associated with the system [5]. These tags collect real-time personal, sensor, and location information, making security a primary concern. Given the passive nature of commercial tags, access control is performed using a minimalist class of algorithms, the Ultralightweight Mutual Authentication Protocols (UMAPs).

In 2006, Petro Peris laid the foundation of ultralightweight cryptography. The main idea was to use triangular functions such as bitwise AND, OR, XOR, and modular addition as encryption primitives [6] [7] [8]. Cryptanalysis of these pioneering protocols proved that they were not viable due to the weaknesses of bitwise operators, which led to full disclosure, Denial of Service(DoS), and man-in-the-middle attacks [9] [10]. In response, Chien introduced the idea of non-triangular primitive, i.e., $Rotation$, as protocol operators [11]; however, it's cryptanalysis still highlighted weaknesses in protocol structure and operations [12][13][14]. Later, in 2017, Tewari and Gupta refined this approach using a combination of two primitives ($Rot(x, y)$, XOR) [15]. However, subsequent cryptanalysis revealed four full disclosure attacks and one desynchronization attack, demonstrating fundamental weaknesses in the design [16] [17].

To improve the cipher of the ultralightweight domain, variations of shuffle-based functions, i.e., permutation, scaled-

✉ Madiha Khalid
madiha.khalid@seecs.edu.pk

[1] National University of Sciences and Technology (NUST), Islamabad, Pakistan

[2] Georgia Gwinnett College (GGC), Lawrenceville, GA, USA

[3] Namal University, Mianwali, Pakistan

Published online: 02 September 2025

down traditional encryption primitives, and Physically Unclonable Functions (PUF), were introduced. For instance, in 2021, Sharqi et al. proposed a novel vector-space-based lightweight privacy-preserving RFID authentication protocol for IoT environments integrating vector space concepts [18]. However, subsequent cryptanalysis by Haradhan Ghosh et al. revealed vulnerabilities in the protocol, including susceptibility to tag anonymity and impersonation attacks [19].

Similarly, Wang et al. proposed a Lightweight RFID Security and Authentication Scheme (LRSAS+) protocol utilizing the SKINNY encryption algorithm [20]. The protocol consists of three phases: initialization, authentication, and update. During authentication, mutual verification between the tag and the server occurs based on the cryptographic keys, followed by an update phase that generates new session keys. Cryptanalysis revealed that LRSAS+ remains vulnerable to desynchronization attacks [21].

Recently, Hosseinzadeh et al. analyzed PUF-Based Secure Lightweight Authentication and Key Exchange (PLAKE) and Lightweight Security Protocol for Dynamic Charging System of Electric Vehicles Using Physical Unclonable Functions (EV-PUF), two Physically Unclonable Functions (PUF) based authentication protocols designed for constrained environments. PLAKE employs challenge response pairs with lightweight cryptographic operations [22], while EV-PUF integrates error correction techniques to enhance resilience against environmental variations [23]. Despite these enhancements, cryptanalysis revealed that both protocols remain vulnerable to impersonation and key leakage attacks [24]. Additionally, PLAKE has been proven susceptible to impersonation attacks, and weaknesses in the protocol have been highlighted through Scyther, an automated verification tool for security protocols [25].

Even though more than a thousand UMAPs using bitwise and shuffling operations have been proposed since 2006, cryptanalysis proves that these UMAPs are prone to disclosure attacks leading to the tag's privacy invasion, generalized desynchronization attacks resulting in DoS, and man-in-the-middle attacks affecting the integrity of the tag/ reader public messages. Therefore, balancing the ever-present yet conflicting goals, i.e., simplicity, security, and flexibility, in designing security mechanisms for passive RFID tags is critical for RFID security.

1.1 Motivation

Recently, a protocol named URMAP, short for Ultralight weight RFID Mutual Authentication Protocol, has been proposed [26]. URMAP is significant as it claims to offer a secure and efficient solution for RFID authentication, particularly emphasizing its resilience against replay attacks. The security analysis of URMAP lacks a detailed evaluation of the Confidentiality, Integrity, and Availability (CIA) claims. This oversight in the security analysis presents a critical research gap that needs to be addressed to ensure the protocol's robustness.

By proving URMAP's vulnerability to tag cloning through confidentiality breaches, the adversary can undermine the security and reliability of an identification network. A cloned tag allows an adversary unauthorized access/ tracking, bypassing security mechanisms, counterfeiting IoT sensing layer data, and disrupting normal operations via Denial of Service (DoS). The significance of addressing these vulnerability lies in the reliance of modern IoT networks on secure communication protocols for critical operations, such as healthcare monitoring, logistics, and industrial automation.

1.2 Contributions

This article presents novel insights into URMAP vulnerabilities by extending the structured cryptanalysis techniques to challenge the protocol's confidentiality claims. The highlighted weaknesses collectively suggest that URMAP may not be suitable for IoT applications that require guaranteed communication channel privacy.

The list of proposed cryptanalysis techniques is as follows:

1. *Tango Cryptanalysis:* The confusion capabilities of URMAP are evaluated through tango cryptanalysis, i.e., a structured probabilistic full disclosure attack model. A total of 15 linear combinations of URMAP's public messages that exhibit minimum hamming distance with tag's ID are identified, defining the average success probability of ID retrieval to be 84.37% with a single eavesdropped session.
2. *Functional Attack:* The encryption primitives of URMAP include AND, OR, XOR and $Rotation$. The biased behavior exhibited by URMAP's public messages due to the imbalanced nature of AND and OR functions is exploited to retrieve concealed tag identifiers (static and dynamic) deterministically with 100% success rate.
3. *Grover's Search Attack:* Quantum resilience of URMAP is evaluated by exposing the protocol to Grover's search algorithm, i.e., a computationally feasible brute force attack. When executed on a simulator, the quantum search algorithm's measurement results exhibit a distinct peak corresponding to the correct key value used in URMAP, given a plaintext/ciphertext pair. This peak enables the deterministic identification of the key.
4. *URMAP^{+}*: To mitigate these vulnerabilities, an enhanced protocol, URMAP^{+}, is proposed, which retains the strengths of URMAP while addressing its security flaws.

1.3 Paper Organization

The paper's organization is as follows: Section 2 briefly overviews the URMAP protocol. Section 3 presents the adversary model and attack assumptions for the proposed confidentiality breaches. Section 4 describes the tango attack, full disclosure attacks and grover search attack followed by section 5 elaborating the updated protocol that addresses the limitations of the URMAP. Finally, section 6 concludes the paper by summarizing the essential findings and contributions.

2 Ultralightweight RFID Mutual Authentication Protocol (URMAP)

The ultra-high frequency passive RFID-enabled EPC Network is a backbone of the IoT applications associated with tracking, identification, supply chain management, retail, and packaging [27]. The EPCglobal published a comprehensive air interface standard defining the passive tags' physical and logical requirements, i.e., EPC Generation-2 (Gen-2) [28]. The Gen-2 standard was accepted as an ISO standard, i.e., ISO 18000-6c, a part of the ISO 18000 standard series. Based on the memory and power limitations, the EPC Generation 2 (EPC Gen-2) tags can be classified as IETF C0-E0 constraint devices [29].

The Ultralightweight RFID Mutual Authentication Protocol (URMAP) is a minimalist security solution tailored for passive RFID systems' computational and energy constraints, specifically those aligned with EPC Gen-2 standards. The protocol has three primary entities. The reader authenticates the tag in coordination with the server. The defining feature of URMAP is its resilience to replay attacks, achieved through timestamps. The server stores the timestamp from the last successful session, i.e., T_t, and the preceding session's authentication takes place only if the current timestamp, i.e., T_c, is later than T_t.

2.1 Memory Architecture

URMAP's identity verification mechanism relies on five values associated with each tag of the RFID system, namely two static identifiers, (ID, SID), and three dynamic identifiers, $(IDS, K1, K2)$.

1. SID is unique to the tag/server network, i.e., all the tags connected to one server will have the same SID.
2. ID is unique for every tag in the network.
3. IDS acts as a pointer to the tag's data in the server's database.
4. The keys, i.e., $K1$ and $K2$, generate challenge-response pairs during the mutual authentication phase.

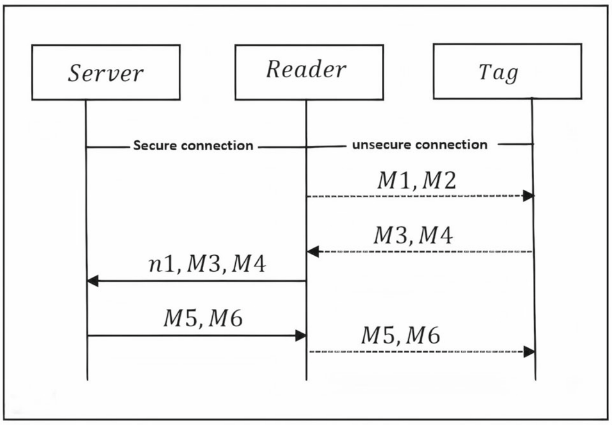


Fig. 1 High-level Description of URMAP Protocol

Additionally, the tag also stores a dynamic value, T_{s1}.

After every successful authentication session, the dynamic variables are updated using a private key $n1$. Both the server and the tag store the latest pairs of $(IDS_{new}, K1_{new}, K2_{new})$, $(IDS_{old}, K1_{old}, K2_{old})$ as a defense mechanism against DoS attacks. The server maintains a flag, i.e., st, to show if the database uses new or old values of pseudonyms for identity verification. Both the server and the tag use variables T_t to store the time stamps, i.e., T_c of authentication sessions in real-time.

2.2 Primitive

URMAP is a non-triangular protocol that uses three bitwise operators, i.e., AND, OR, XOR, and one shuffle-based operator, i.e., $Rotation(x, y)$. The rotation function gives a circular left shift to operand x based on the hamming weight of operand y.

2.3 Protocol Description

The URMAP protocol involves a series of interactions between the server and the tag. The process is illustrated in figure 1. The protocol description is as follows:

1. The reader initiates the session by sending public messages $M1$ and M_2 as defined in equation (1) and (2). $M1$ encrypts the session key $(n1)$ generated by the reader, and $M2$ encrypts the timestamp T_c.

$$M1 = SID \oplus n1 \quad (1)$$

$$M2 = (SID \oplus T_c) \wedge n1 \quad (2)$$

2. The tag extracts $n1$,T_c and verifies the freshness of the message. In case of success, the tag sends public messages $M3$ and $M4$ as defined in equation (4) and (5) for

the server via reader.

$$T'_{s1} = (n1 \oplus T_{s1}) \wedge (n1 \oplus SID) \quad (3)$$

$$M3 = SID \oplus T'_{s1} \quad (4)$$

$$M4 = IDS \oplus T'_{s1} \quad (5)$$

3. The server identifies the tag using IDS communicated through $M4$. The value of st is set to “new" if $IDS = IDS_{new}$, else st assumes the value “old". The server uses $M5$ as defined in equation (6) to re-encode the T_c timestamp and $M6$ as defined in equation (9) for authentication.

$$M5 = T_c \oplus IDS_{st} \quad (6)$$

$$K1' = Rot(K1_{st} \oplus n1, IDS_{st}) \quad (7)$$

$$K2' = Rot(K2_{st} \oplus T_{s1}, IDS_{st}) \quad (8)$$

$$M6 = (M5 \oplus K2') \wedge K1' \quad (9)$$

In the case of $st = new$, the dynamic variables of the server update at this point using equations (10-13).

$$IDS_{old} = IDS_{st}; K1_{old} = K1_{st} \quad (10)$$

$$K2_{old} = K2_{st} \quad (11)$$

$$IDS_{new} = (IDS_{st} \oplus ID) \vee (n1 \oplus T_{s1}) \quad (12)$$

$$K1_{new} = K1^{'}; K2_{new} = K2^{'} \quad (13)$$

4. The tag generates a response for the $M6$ challenge message after verifying the validity of T_c concealed in $M5$. After successful reader authentication, the tag's dynamic memory updates using equations (14-18)

$$IDS_{old} = IDS; K1_{old} = K1 \quad (14)$$

$$K2_{old} = K2 \quad (15)$$

$$IDS_{new} = (IDS \oplus ID) \vee (n1 \oplus T_{s1}) \quad (16)$$

$$K1_{new} = K1^{'}; K2_{new} = K2^{'} \quad (17)$$

$$T_{s1} = T'_{s1}; T_t = T_c \quad (18)$$

2.4 Security Claims

The URMAP proposal includes a complete functional and formal analysis, claiming its resilience to active attacks, especially replay. Formal analysis, e.g., Avispa and Scyther, verify a protocol's security properties by evaluating the algorithm's flow.

URMAP is based on the ISO-9798 three-pass entity authentication mechanism. The formal analysis of the general ISO-9798 standard, specifically URMAP, validates that the protocol flow ensures CIA [26] [30]. However, functional analysis is still required to evaluate the strength of the protocol primitives. In addition to this, the protocol's response to passive attacks has been overlooked, affecting confidentiality.

The subsequent section presents the adversary model that can potentially compromise the identification system using a series of passive full-disclosure attacks.

3 Adversary Model and Attack Assumptions

The proposed cryptanalysis techniques aim to compromise the tag's confidentiality by retrieving the tag's identifiers (static/dynamic), i.e., (SID, ID), $(IDS, K1, K2, T_{s1}, n1)$ from URMAP's public messages. Knowledge of all the attributes enables tag cloning, which can lead to IoT network poisoning.

Since URMAP aligns with the Dolev-Yao model as a two-party protocol claiming secrecy, the following are the adversary properties defined by the same model [31]:

1. Eavesdrop messages exchanged between the tag/reader pair.
2. Intercept, modify, or inject new or modified messages into the communication channel.
3. Replay public messages from previous authentication sessions.
4. Block public messages of an ongoing authentication session.
5. The adversary is computationally bounded, i.e., it can not perform computationally infeasible brute forcing of URMAP's keys. However, the adversary has a general-purpose computing resource equipped to perform bitwise operations and has Anaconda Python installed along with the Qiskit library to access and utilize the IBM Quantum simulator/fake backend.

The proposed attack model comprises two independent cryptanalysis techniques targeting the secrecy claims of URMAP. A brief description of the proposed techniques in the light of the Dolev-Yao adversary model is as follows:

1. *Tango Cryptanalysis:* The adversary's ability to eavesdrop and perform general-purpose computing enables a probabilistic full disclosure attack for the tag's ID retrieval.
2. *Functional Cryptanalysis followed by Grover's search Algorithm:* For functional cryptanalysis, the adversary blocks public messages to halt the update of dynamic identifiers, followed by the targeted tag's identifier retrieval through simple bitwise operations on the eavesdropped authentication session. Using the network identifier SID retrieved from the functional cryptanalysis and the adversary's capability to access the IBM simu-

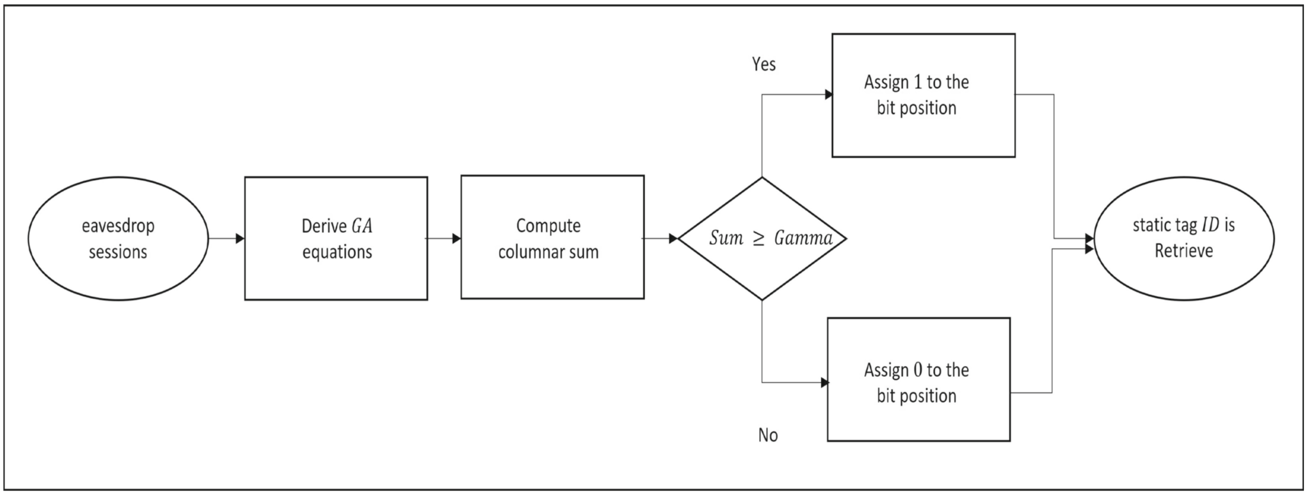

eavesdrop sessions
Derive GA equations
Compute columnar sum
$Sum \geq Gamma$
Yes
Assign 1 to the bit position
No
Assign 0 to the bit position
static tag ID is Retrieve

Fig. 2 Tango cryptanalysis flow diagram

lator, the Grover's Search is applied to the eavesdropped authentication sessions to retrieve the credentials of all the tags associated with the identification system under attack. It is important to highlight that the adversary uses the IBM Quantum simulator to model grover's search algorithm, which reduces the search space for keys but does not break cryptographic primitives directly. This aligns with the Dolev-Yao assumption of a computationally bounded adversary.

These attacks demonstrates that URMAP fails to provide confidentiality and integrity guarantees under Dolev-Yao model rendering it ineffective for IoT applications. Subsequent sections present the experimental results of the proposed attacks.

4 Security Analysis of URMAP

URMAP is an authenticated encryption mechanism that claims robustness against replay attacks through time stamps. This feature enables the protocol to claim CIA assurance to the tag/reader pair.

Despite the security claims, the protocol is vulnerable to full disclosure attacks that challenge the confidentiality offered by the cipher. The proposed attack model comprises three techniques, i.e., tango cryptanalysis, functional cryptanalysis, and Grover's search-based brute force attack. This section describes the execution of the above-listed methods on URMAP:

4.1 Tango Cryptanalysis

Tango cryptanalysis is a probabilistic full-disclosure attack that exploits the weak confusion and diffusion properties of URMAP. The nature of the protocol, i.e., $Rotation$, AND, and OR operations, makes it prone to the tango attack [17]. The attack execution comprises two steps, which are discussed in figure 2 and described as below:

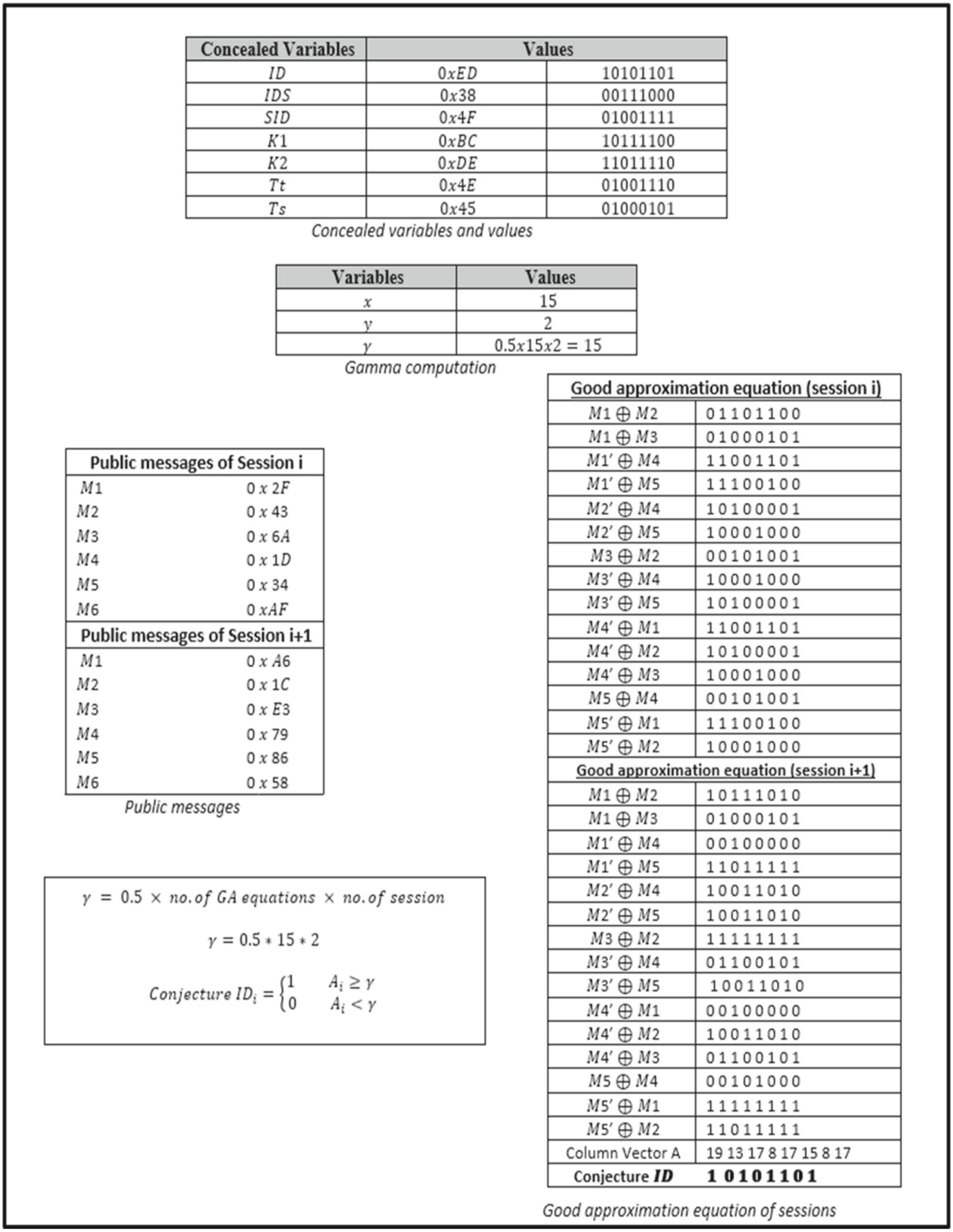
1. In this step, the protocol's public messages are analyzed by simulating the authentication sessions with hypothetical tag attributes. Linear combinations of public messages that give a minimum hamming distance from the hypothesized tag's static ID are selected. These combinations are termed Good Approximation (GA) equations. For URMAP, 1500 sessions were simulated with hypothesized 96-bit tag attributes. Equations with an average hamming distance of less than 48 were selected. Table 9 presented in Annex Appendix A provides a complete set of test equations.
2. This step involves extracting the ID of the targeted tag via eavesdropping. Public messages from authentication sessions are eavesdropped, GA equations are calculated, and the results are maintained in a $2D$ array. The ID is estimated in a bitwise fashion by (a) taking the columnar sum of bit values and (b) comparing the resultants at each position to gamma (γ):

$$\gamma = 0.5 \times \text{no. of GA equations} \times \text{no. of sessions} \qquad \text{(a)}$$

A 1 is placed at a bit position if the columnar sum is greater than or equal to γ; otherwise, a 0 is placed.

Figure 3 demonstrates the execution of tango cryptanalysis on the 8-bit ID i.e, a scaled-down version of the protocol. The figure shows that the ID can be retrieved by eavesdropping just two sessions.

Additionally, figure 4 presents the trend of ID retrieval for 96-bit identification systems. According to the graph, single



Concealed Variables	Values	
ID	$0xED$	10101101
IDS	$0x38$	00111000
SID	$0x4F$	01001111
$K1$	$0xBC$	10111100
$K2$	$0xDE$	11011110
Tt	$0x4E$	01001110
Ts	$0x45$	01000101

Concealed variables and values

Variables	Values
x	15
y	2
γ	$0.5x15x2 = 15$

Gamma computation

Public messages of Session i	
$M1$	$0\ x\ 2F$
$M2$	$0\ x\ 43$
$M3$	$0\ x\ 6A$
$M4$	$0\ x\ 1D$
$M5$	$0\ x\ 34$
$M6$	$0\ xAF$
Public messages of Session i+1	
$M1$	$0\ x\ A6$
$M2$	$0\ x\ 1C$
$M3$	$0\ x\ E3$
$M4$	$0\ x\ 79$
$M5$	$0\ x\ 86$
$M6$	$0\ x\ 58$

Public messages

$$\gamma = 0.5 \times no.of\ GA\ equations \times no.of\ session$$

$$\gamma = 0.5 * 15 * 2$$

$$Conjecture\ ID_i = \begin{cases} 1 & A_i \geq \gamma \\ 0 & A_i < \gamma \end{cases}$$

Good approximation equation (session i)	
$M1 \oplus M2$	01101100
$M1 \oplus M3$	01000101
$M1' \oplus M4$	11001101
$M1' \oplus M5$	11100100
$M2' \oplus M4$	10100001
$M2' \oplus M5$	10001000
$M3 \oplus M2$	00101001
$M3' \oplus M4$	10001000
$M3' \oplus M5$	10100001
$M4' \oplus M1$	11001101
$M4' \oplus M2$	10100001
$M4' \oplus M3$	10001000
$M5 \oplus M4$	00101001
$M5' \oplus M1$	11100100
$M5' \oplus M2$	10001000
Good approximation equation (session i+1)	
$M1 \oplus M2$	10111010
$M1 \oplus M3$	01000101
$M1' \oplus M4$	00100000
$M1' \oplus M5$	11011111
$M2' \oplus M4$	10011010
$M2' \oplus M5$	10011010
$M3 \oplus M2$	11111111
$M3' \oplus M4$	01100101
$M3' \oplus M5$	10011010
$M4' \oplus M1$	00100000
$M4' \oplus M2$	10011010
$M4' \oplus M3$	01100101
$M5 \oplus M4$	00101000
$M5' \oplus M1$	11111111
$M5' \oplus M2$	11011111
Column Vector A	19 13 17 8 17 15 8 17
Conjecture ***ID***	**1 0101101**

Good approximation equation of sessions

Fig. 3 Tango cryptanalysis for 8-bit

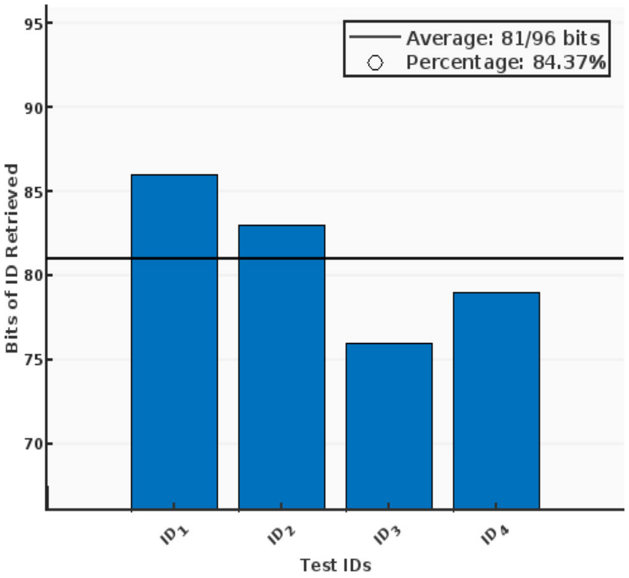


Fig. 4 96-bit key retrieval

session eavesdrop retrieves an average of 81 bits giving the success probability of 84.37%. This result reduces the brute force attack complexity for estimating ID.

4.2 Functional Cryptanalysis

This is a deterministic full-disclosure attack comprising the following steps:

4.2.1 Partial De-synchronization

A desynchronization attack disrupts the synchronization between a tag and a reader, preventing successful authentication.

The generalized de-synchronization attack does not apply to URMAP due to its robustness against replay attacks [32]. However, one can block public messages to alter the dynamic nature of the tag's attributes (IDS, $K1$, $K2$, T_{s1}), rendering them static.

In this scenario, the attacker blocks message $M5$ and $M6$, causing the RFID tag to keep the dynamic variable values unchanged. Consider a synchronized tag/reader pair, table 1 elaborates on the proposed partial desynchronization attack.

The adversary will first eavesdrop on the completely synchronized authentication session of the targeted tag. This session is labeled as *Session i* in table 1. For the subsequent sessions, public messages will be sniffed and message $M5$ and $M6$ will be blocked to keep the values of (IDS, $K1$, $K2$, T_{s1}) static at the tag's side. The database of public messages the adversary maintains will be used in the subsequent attack to reveal the tag's attributes.

Table 1 Partial desynchronization attack on URMAP

Server		**Tag**	
IDS_{old}	IDS_{new}	IDS_{old}	IDS_{new}
Session I: Adversary eavesdrops public messages			
IDS_0	IDS_1	IDS_0	IDS_1
Session i+1: Adversary blocks message $M6$			
IDS_1	IDS_2	IDS_0	IDS_1
Session i+2: Adversary blocks message $M6$			
IDS_1	IDS_2	IDS_0	IDS_1

4.2.2 Operation Imbalance Exploit Attack

This step analyzes the set of public messages eavesdropped from a synchronized session i, followed by x partially de-synchronized sessions. The partially desynchronized sessions are recorded and analyzed till all the attributes are completely calculated. The properties of bit-wise functions used in the proposed attack are:

$$\textbf{P1:}\quad a \oplus b = c \implies c \oplus b = a,$$
$$\textbf{P2:}\quad a \wedge b = 1 \implies a = 1;\ b = 1,$$
$$\textbf{P3:}\quad a \vee b = 0 \implies a = 0;\ b = 0,$$
$$\textbf{P4:}\quad \text{Rot}(a, b) = c \implies \text{Rot}^{-1}(c, b) = a.$$

the attack execution comprises the following steps, which are defined in the figure 5 and described below:

1. SID is retrieved iteratively by extracting $n1$ through **P2** and $M2$ equation (2) . The extracted $n1$ is then used to calculate SID using **P1** on $M1$ equation (1).
2. Once SID is retrieved, $n1$ for every session can be calculated using **P1** on $M1$ equation (1).
3. $T_{s1}^{'}$ and IDS are evaluated using the estimated SID, $M3$ equation (4), $M4$ equation (5) and **P1**.
4. $T_{s1}$is calculated using session i, the estimated SID, and equation M_{3i} equation (4). The $T_{s1}^{'}$ of synchronized session i i.e. $T_{s1i}^{'}$ will become T_{s1} of the subsequent partially de- synchronized session.
5. $K1^{'}$ and $K1$ are calculated iteratively using Rule **P2** applied on equation $M6$ equation (9) followed by **P4** applied on equation (7).
6. $K2^{'}$ is calculated iteratively using **P2** applied on equation $M6$ equation (9). After complete calculation of $K2^{'}$, **P4** is applied on equation (8) to calculate $K2$.

For the estimation of ID, the partial desynchronization should be reverted by allowing the tag's memory to update using message $M5$ and $M6$ as define in equation (6) and (9). With SID retrieved from previously elaborated strategy, $n1$, T'_{s1}, and IDS can be retrieved using $M1$, $M3$ and $M4$ as

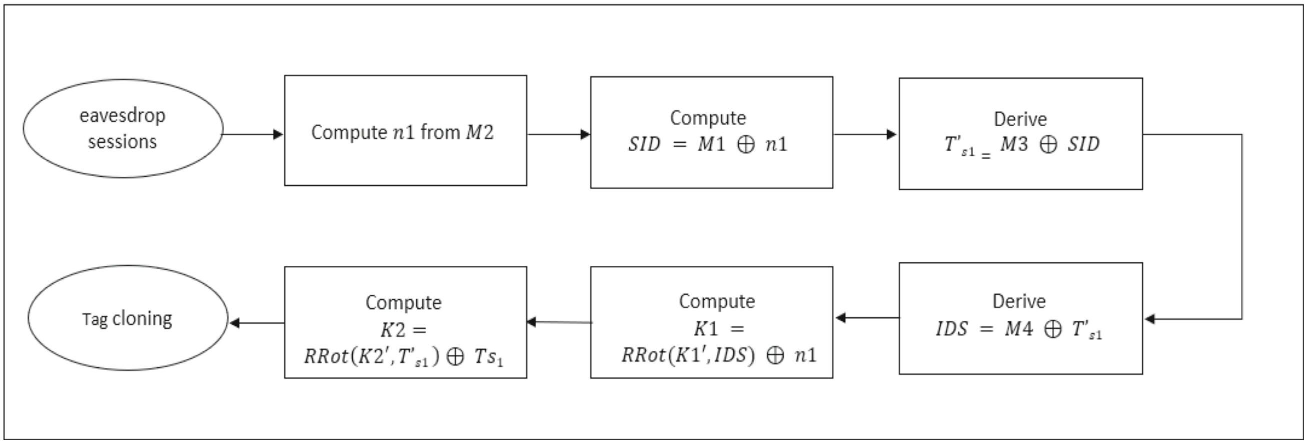

eavesdrop sessions
Compute $n1$ from $M2$
Compute $SID = M1 \oplus n1$
Derive $T'_{s1} = M3 \oplus SID$
Derive $IDS = M4 \oplus T'_{s1}$
Compute $K1 = RRot(K1', IDS) \oplus n1$
Compute $K2 = RRot(K2', T'_{s1}) \oplus Ts_1$
Tag cloning

Fig. 5 Full disclosure attack flow chart

define in equation (1), (4) and (5). Furthermore, T_{s1} can be evaluated; as for the given session j; T_{s1} assumes the value of $T^{'}_{s1}$ of the preceding session $j - 1$.

Therefore, for each synchronized session, the associated attributes, i.e., $n1$, T_{s1}, and IDS, are used for the ID calculation. The step-by-step elaboration of the process is as follows:

1. With the values of $n1$, T_{s1}, and the IDS of the subsequent session i.e. IDS_{j+1}; $ID \oplus IDS_j$ can be calculated iteratively by applying **P3** to equation (16).
   - If a bit in IDS_{j+1} is 0, it confirms that both inputs of the OR operation were 0.
   - If a bit in IDS_{j+1} is 1 but the corresponding bit in $(n1 \oplus T_{s1})$ is 0, it means that $(IDS_j \oplus ID)$ must be 1.
2. Finally ID is retrieved iteratively using the the estimate of $(IDS_j \oplus ID)$, value of IDS_j and **P1** .

Figure 6 presents an example for 8-bit tag attribute retrieval. In the given example, eight sessions are eavesdropped. Variable j identifies the session being analyzed for the attribute retrieval. The first six sessions are partially desynchronized for the extraction of IDS, $K1$, $K2$, T_{s1}, $n1$,SID), and the last two are synchronised to receive the tag's ID.

After all the tag attributes are successfully retrieved, these values can be written on a blank tag to execute tag cloning. The subsequent section elaborates on a full disclosure attack on the rest of the tags associated with the server based on SID.

4.3 Grover's Search-Based Brute Force Attack

Grover's search algorithm accelerates brute force attacks on symmetric ciphers using a key search quantum circuit.

The key search algorithm requires a pair of Plain Text (PT), Cipher Text (CT), and encryption circuits. Under analysis, the encryption quantum circuit of block cipher takes all possible values of the key and the PT as input and compares the known CT with the generated CTs. In the case of a match, the diffusion operator amplifies the output to estimate the symmetric key [33]. The block diagram of Grover's search based brute force attack is presented in figure 8a .

The search can be applied individually to all URMAP equations to extract dynamic values associated with the targeted tag; however, in the proposed model, quantum circuits are specifically designed only for equations that are not classically reversible i.e., equations (3) and (9). For the quantum implementation of these equations, consider the generalized form:

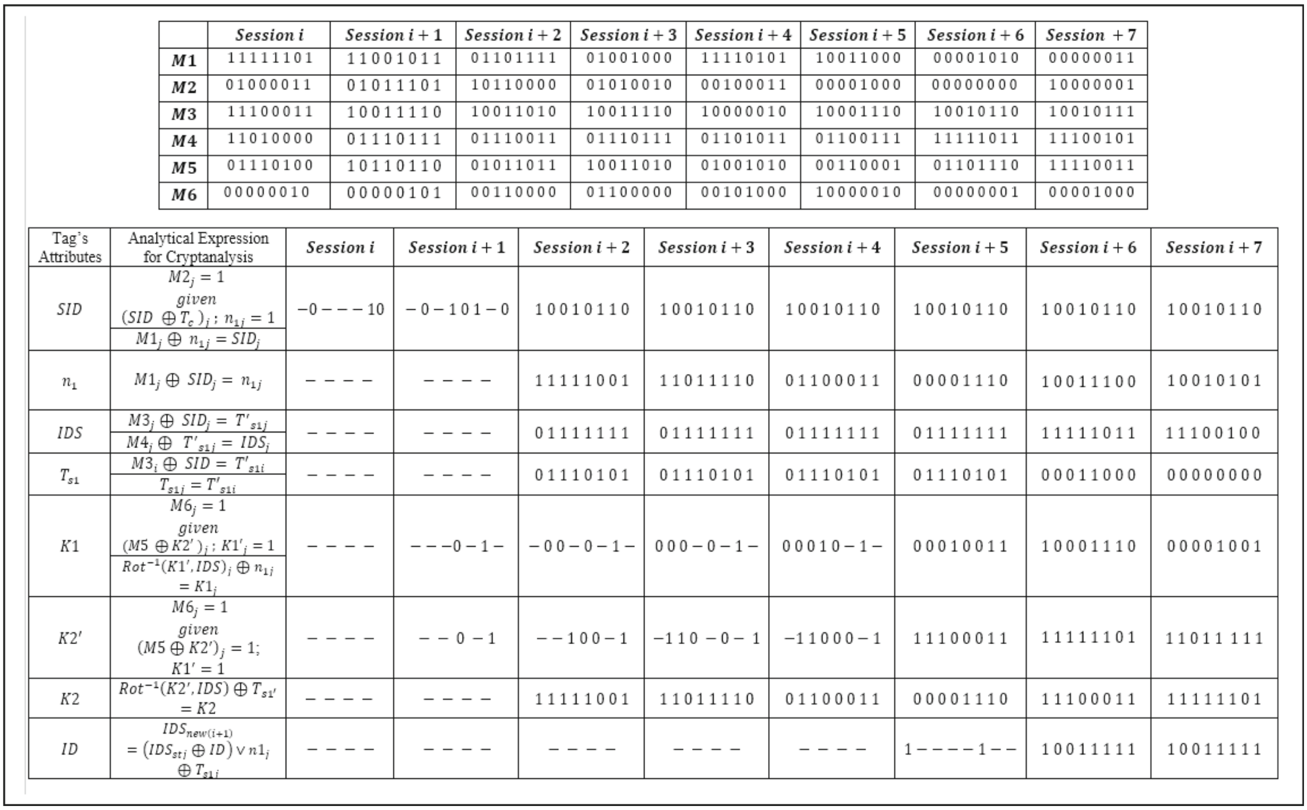
$$CT = (PT \oplus Key) \wedge Const \tag{19}$$

Where:

- $CT \rightarrow T'_{s1}$; $CT \rightarrow M6$
- $PT \rightarrow n1$; $PT \rightarrow M5$
- $Key \rightarrow T_{s1}$; $Key \rightarrow K2'$
- $Const \rightarrow (n1 \oplus SID)$; $Const \rightarrow K1'$

In quantum arithmetic, the OR and AND gates correspond to the $controlled-NOT$ and $Toffoli$ gates, respectively. The proposed quantum circuit for equation (19) is designed using the IBM Quantum Platform. Figure 7 presents the circuit diagram of the bit-level quantum implementation of URMAP.

In this circuit, the $qubit$ representing the Key value is $Xored$ with PT using a $Cnot$ gate. Key and PT are marked as



	$Session\ i$	$Session\ i+1$	$Session\ i+2$	$Session\ i+3$	$Session\ i+4$	$Session\ i+5$	$Session\ i+6$	$Session\ +7$
$M1$	11111101	11001011	01101111	01001000	11110101	10011000	00001010	00000011
$M2$	01000011	01011101	10110000	01010010	00100011	00001000	00000000	10000001
$M3$	11100011	10011110	10011010	10011110	10000010	10001110	10010110	10010111
$M4$	11010000	01110111	01110011	01110111	01101011	01100111	11111011	11100101
$M5$	01110100	10110110	01011011	10011010	01001010	00110001	01101110	11110011
$M6$	00000010	00000101	00110000	01100000	00101000	10000010	00000001	00001000

Tag's Attributes	Analytical Expression for Cryptanalysis	$Session\ i$	$Session\ i+1$	$Session\ i+2$	$Session\ i+3$	$Session\ i+4$	$Session\ i+5$	$Session\ i+6$	$Session\ i+7$
SID	$M2_j = 1$ $given$ $(SID \oplus T_c)_j ; n_{1j} = 1$ $M1_j \oplus n_{1j} = SID_j$	−0−−−10	−0−101−0	10010110	10010110	10010110	10010110	10010110	10010110
n_1	$M1_j \oplus SID_j = n_{1j}$	−−−−	−−−−	11111001	11011110	01100011	00001110	10011100	10010101
IDS	$M3_j \oplus SID_j = T'_{s1j}$ $M4_j \oplus T'_{s1j} = IDS_j$	−−−−	−−−−	01111111	01111111	01111111	01111111	11111011	11100100
T_{s1}	$M3_i \oplus SID = T'_{s1i}$ $T_{s1j} = T'_{s1i}$	−−−−	−−−−	01110101	01110101	01110101	01110101	00011000	00000000
$K1$	$M6_j = 1$ $given$ $(M5 \oplus K2')_j ; K1'_j = 1$ $Rot^{-1}(K1', IDS)_j \oplus n_{1j} = K1_j$	−−−−	−−−0−1−	−00−0−1−	000−0−1−	00010−1−	00010011	10001110	00001001
$K2'$	$M6_j = 1$ $given$ $(M5 \oplus K2')_j = 1; K1' = 1$	−−−−	−−0−1	−−100−1	−110−0−1	−11000−1	11100011	11111101	11011 111
$K2$	$Rot^{-1}(K2', IDS) \oplus T_{s1'} = K2$	−−−−	−−−−	11111001	11011110	01100011	00001110	11100011	11111101
ID	$IDS_{new(i+1)} = (IDS_{stj} \oplus ID) \vee n1_j \oplus T_{s1i}$	−−−−	−−−−	−−−−	−−−−	−−−−	1−−−−1−−	10011111	10011111

Fig. 6 Full disclosure attack on URMAP's 8-bit variant

control and target $qubits$, respectively. The output obtained as a transition on the target $qubit$ is further $Anded$ with Constant value using $Toffoli$ gate to get the final output as a transition on the $qubit$ representing CT.

This circuit is transformed into a phase oracle that compares the calculated CT (sequentially using all possible keys) with the given CT. The oracle is then connected to the diffusion operator to amplify the correct key per the block diagram in figure 8a . This basic structure is replicated to enhance the desired key value.

Figure 8b shows the IBM Qiskit implementation of URMAP Grover search algorithm that gives the value of 2-bit key in 1 iterations using a simulated environment. In this diagram, the first layer is for initializing (PT,CT) pair, followed by one iterations of Grover's search. Finally, the last layer gives the searched value of the key as a measured output of $qubit$ representing the key values. The given circuit is abstracted by representing $2-bit$ variables(Key,PT,$Const$, CT_given, $CT\_Calculated$) as single circuit line to improve logical clarity.

Grover's search algorithm demonstrates the unstructured search for all possible combinations of the key that generates a given (PT, CT) pair. Therefore, for reversible cipher primitives, a precise value of the key is retrieved.

Equation (19) is irreversible due to the AND function. However, if the value of $Const$ is restricted to 1, the function's response to key search becomes targeted. Figure 8b shows that $Const$ $qubit$ is initialized to 11 by applying $pauli-X$ gate to $|0>|0>$. $Const = 11$ is the underlying assumption of URMAP quantum cryptanalysis. PT and CT_given are initialized to retrieve Key value by comparing the $CT\_calculated$ and CT_given.

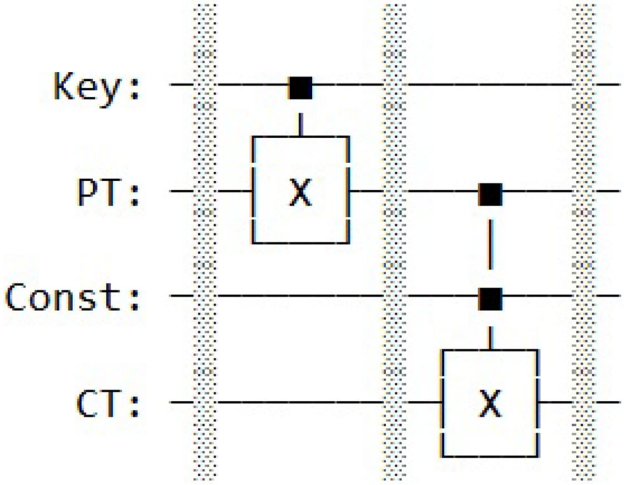
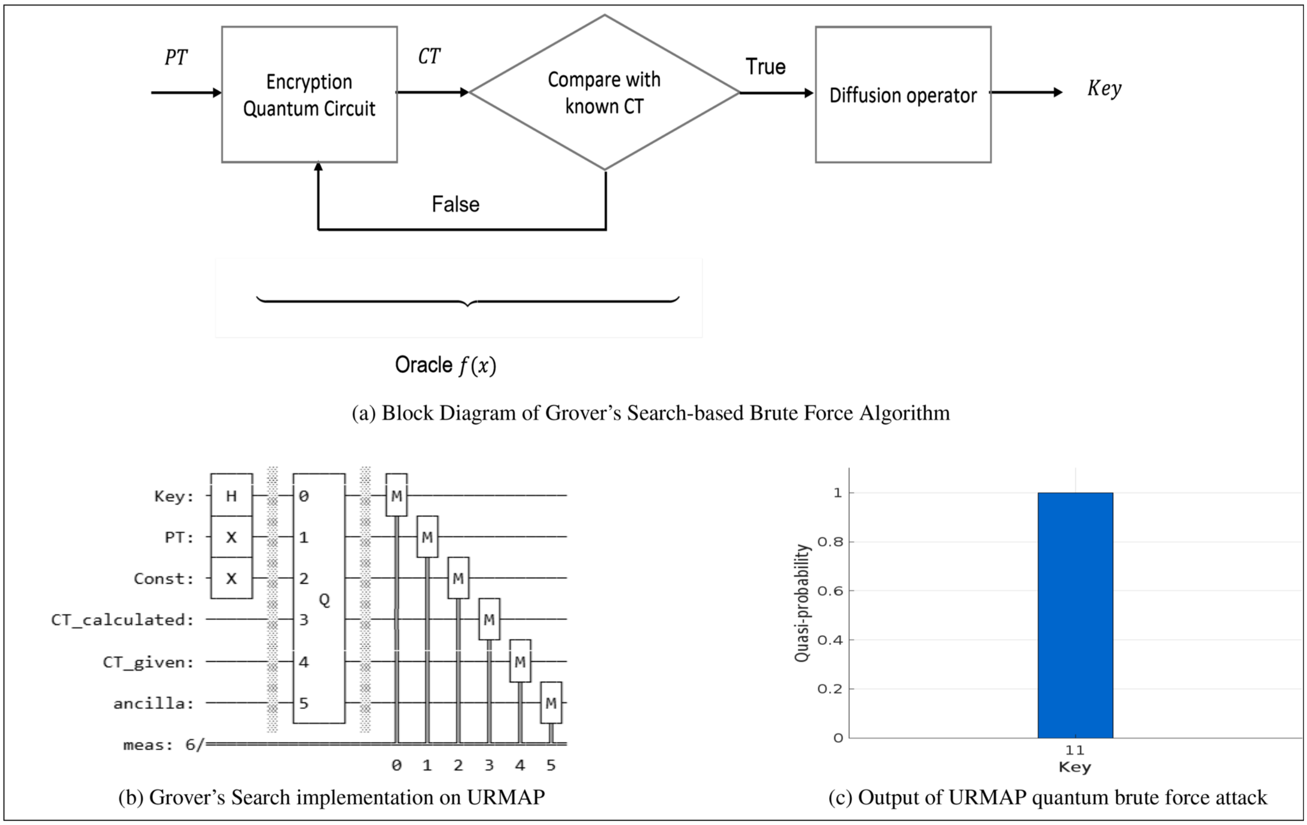


Fig. 7 Generalised quantum circuit for URMAP equations

The given circuit is initialized with $PT = 11$,$CT\_given = 00$, and the expected output of the search algo-



(a) Block Diagram of Grover's Search-based Brute Force Algorithm

(b) Grover's Search implementation on URMAP

(c) Output of URMAP quantum brute force attack

Fig. 8 Grover's Search Algorithm

rithm, i.e., $Key = 11$, is shown in figure 8c as the only probable outcome.

By the compact size of the circuit defined in Figure 8b , the output of Grover's search is a well-defined peak when IBM Qiskit *Aer Simulator* is utilized. It is important to highlight that the IBM simulator is accessible through an open source Python library, qiskit, and requires the resources equivalent to a general-purpose computer.

On the contrary, the algorithm's results are unreliable on IBM quantum computing resources due to the inherent challenges of maintaining *qubits* and their operations in a stable, noise-free environment. This impact is experimentally proven by executing the circuit in figure 8b on ibm_Kyiv an open source IBM quantum computer. The measurement result produced 2^{11} probabilistic outputs instead of one defined peak.

Given the server SID is retrieved through functional analysis section 4, the detailed description of URMAP quantum cryptanalysis on tag$_\alpha$ is as follows:

1. First, tag$_\alpha$ is partially desynchronized as per the model given in Section 4.2.1. The attacker will record public messages from the partially desynchronized session to execute full disclosure.
2. Since SID of the tag/server model of the targeted network was calculated deterministically in section 4, SID, $M1$, and $M3$ will be used to calculate $n1$ and T'_{s1}, respectively.

$$n1 = M1 \oplus SID \tag{20}$$

$$T'_{s1} = M3 \oplus SID \tag{21}$$

3. IDS and T_c are calculated using $M4$ and $M5$, respectively.

$$IDS = M4 \oplus T'_{s1} \tag{22}$$

$$T_c = M5 \oplus IDS \tag{23}$$

4. T_{s1} will be calculated in a bit-wise manner. For every eavesdropped session, bit positions where $SID \oplus n1 = 1$; equation (19) and figure 8b are used to estimate the corresponding bit values of T_{s1}.
5. By the imbalanced nature of AND, a portion of $K1'$ is retrieved in every session i.e., at bit positions where $M6 = 1$; $K1' = 1$. Concurrently, $K1$ is partially estimated using equation (24).

$$RRot(K'_1, IDS) \oplus n1 = K_1 \tag{24}$$

This process is repeated over multiple sessions to recover $K1$.

6. With the value of $K1$, $K1^{'}$ can be calculated for every session. Using Grover's search algorithm, a portion of $K2^{'}$ is retrieved in every session i.e., at bit positions where $K1^{'} = 1$. Once $K2^{'}$ is fully recovered equation (25) is used to calculate $K2$.

$$RRot(K_2', IDS) \oplus T_{s1}' = K_2 \tag{25}$$

7. Finally tag$_\alpha$ ID is retrieved by the description given in section 4.2.2.

This attack can reveal the attributes of all tags associated with the identification network under threat. The adversary can then clone the entire identification network, ultimately compromising the integrity of the IoT application sensing layer.

The attack model presented in the preceding subsections nullifies URMAP's confidentiality claims. The tango attack estimates the tag's ID with an average precision of 84.37%, and the proposed functional attack deterministically retrieves tag attributes (ID, (SID, IDS, $K1$, $K2$, $n1$, T_{s1}) by exploiting the operational imbalance of the AND and OR functions. Therefore, the adversary can compromise the IoT application by creating a cloned sensing layer identification network by extracting tag identifiers.

Finally, Grover's search-based brute force attack is effective on URMAP for the following reasons:

- AND, OR and XOR are bitwise operations. Therefore, instead of designing an oracle of 96-bit block cipher, the problem is simplified to a two-bit encryption circuit. The down-scaling in the Grover search phase oracle enables efficient algorithm execution even in the simulation environment. This enables the adversary with basic computing capabilities to execute the brute force attack.
- The imbalance nature of AND function gives one to one correspondence to (CT, Key) provided $Const = 1$. This property enables the output of a unique Key estimate, which will have the highest probability of occurrence.

Given the level of vulnerability to full-disclosure attacks, URMAP should be rendered ineffective for IoT security. Therefore, the subsequent section proposes a full-fledged protocol to bridge the gaps identified in URMAP [26].

5 Ultralightweight RFID Authentication Protocol Plus (URMAP$^+$)

URMAP$^+$ is an enhanced ultralightweight mutual authentication protocol designed to address critical vulnerabilities in URMAP, particularly its susceptibility to functional cryptanalysis and quantum attacks. By replacing weak primitives (e.g., $AND/Rotation$) with XOR and Double $Rotation$ in an SPN structure, URMAP+ significantly improves confusion and diffusion while maintaining low computational overhead. The protocol also introduces dynamic session identifiers to resist tracking and desynchronization attacks, making it a secure solution for IoT and RFID applications. Following is the description of the protocol:

5.1 Memory Architecture

The memory architecture of URMAP introduced the concept of a timestamp and a buffer-based dynamic memory to store the latest pairs of IDS, $K1$, and $K2$. This architecture makes the protocol resilient to active replay and desynchronization attacks. Therefore, table 2 showcases the memory architecture of URMAP$^+$, derived from the original URMAP design.

The protocol associates a set of static and dynamic tag identifiers, i.e, (ID) and (IDS, $K1$, $K2$, SID) with each tag, in addition to timestamp represented by value T_c stored as variable T_t at both sides.

5.2 Primitives

A literature review shows that non-triangular operators' confusion and defusion capabilities, i.e., shuffle base functions, are better than bitwise functions, i.e., AND and OR. The primary reason for this trend is the imbalanced nature of these triangular functions.

In light of this observation, URMAP$^+$ is defined as a single non-triangular cipher with only two primitives, i.e., XOR and $Rotation$. Where $Rot(x, y)$ refers to the circular rotation of x by hamming weight of y.

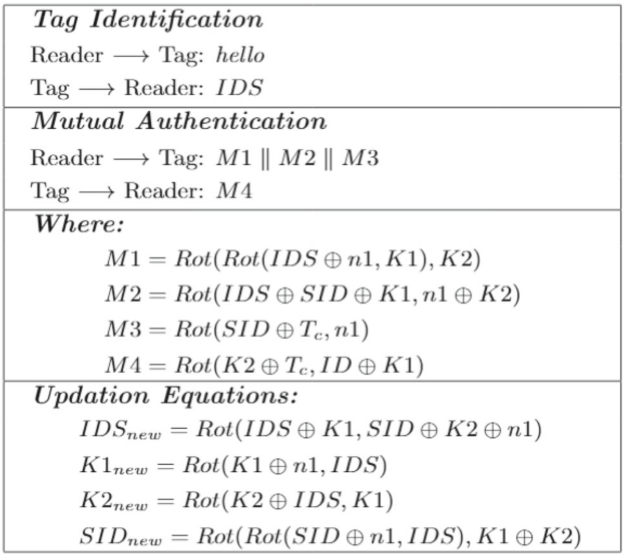
5.3 Protocol Description

URMAP$^+$ authenticates session comprises three phases. Description of these phases are elaborated as follows Figure 9:

1. *Tag Identification:* In this phase, the tag is identified in the reader database through pseudonym IDS.
   - The reader sends a $ping$ message to the passive tag for the session initiation.
   - The tag replies with the latest IDS as plain text. The reader uses this value to locate the tag in its database. In case of a match, the protocol moves to the next step; otherwise, the reader requests the tag for the old IDS.
   - If none of the IDS is found in the reader's database, the protocol terminates.

Table 2 Memory Architecture

Component	Static Memory	Dynamic Memory
Tag	ID	IDS_{old}, $K1_{old}$, $K2_{old}$,SID_{old} IDS_{new}, $K1_{new}$, $K2_{new}$, SID_{new} T_t
Server / Reader	ID	IDS_{old}, $K1_{old}$,$K2_{old}$, SID_{old} IDS_{new}, $K1_{new}$, $K2_{new}$, SID_{new} T_t



Tag Identification

Reader $\longrightarrow$ Tag: *hello*

Tag $\longrightarrow$ Reader: IDS

Mutual Authentication

Reader $\longrightarrow$ Tag: $M1 \parallel M2 \parallel M3$

Tag $\longrightarrow$ Reader: $M4$

Where:

$$M1 = Rot(Rot(IDS \oplus n1, K1), K2)$$
$$M2 = Rot(IDS \oplus SID \oplus K1, n1 \oplus K2)$$
$$M3 = Rot(SID \oplus T_c, n1)$$
$$M4 = Rot(K2 \oplus T_c, ID \oplus K1)$$

Updation Equations:

$$IDS_{new} = Rot(IDS \oplus K1, SID \oplus K2 \oplus n1)$$
$$K1_{new} = Rot(K1 \oplus n1, IDS)$$
$$K2_{new} = Rot(K2 \oplus IDS, K1)$$
$$SID_{new} = Rot(Rot(SID \oplus n1, IDS), K1 \oplus K2)$$

Fig. 9 URMAP+ description

2. *Mutual Authentication:* In this phase, the tag/reader pair gets authenticated through a challenge-response mechanism and verifies tag identifiers.

   - The reader generates a random number ($n1$) that acts as session private key. This value is encrypted using equation (26).

   $$M1 = Rot(Rot(IDS \oplus n1, K1), K2) \tag{26}$$

   - The reader generates the reader authentication challenge message $M2$ given in the equation (27).

   $$M2 = Rot(IDS \oplus SID \oplus K1, n1 \oplus K2) \tag{27}$$

   - To ensure the timeliness of the authentication session, the reader generates $M3$, an encrypted version of timestamp T_c.

   $$M3 = Rot(SID \oplus T_c, n1) \tag{28}$$

   The messages $M1$, $M2$, $M3$ are concatenated and sent to the tag's side. The steps performed by the tag are:

   - The tag extract $n1$ from message $M1$, validates reader by generating local copy of $M2$ and comparing it with the received value and extract timestamp T_c from $M3$.
   - If the extracted T_c is greater than T_t, timeliness is verified, and T_t assumes the received values of T_c else, the protocol identifies an active replay attack and gets terminated.
   - Next, the tag generates message $M4$, which has the dual purpose of tag authentication and timeliness verification. Equation (29) defines messages $M4$.

   $$M4 = Rot(K2 \oplus T_c, ID \oplus K1) \tag{29}$$

   Only a valid tag has possession of static ID and knowledge of current T_c; therefore, only a valid tag can generate $M4$ to prove its identity and message timeliness.

3. *Dynamic Memory Update:* The final step involves the update of the tag's dynamic identifiers to enhance the freshness of public messages of consecutive authentication sessions.

   - Tag's dynamic memory updates after processing messages $M1$, $M2$, $M3$, i.e., after verification of the reader and timeliness of public messages.
   - Reader's dynamic memory updates after tag verification through message $M4$.

   The dynamic memory update equations are as follows:

   $$IDS_{old} = IDS;\ K1_{old} = K1;$$
   $$K2_{old} = K2;\ SID_{old} = SID \tag{30}$$
   $$IDS_{new} = Rot(IDS \oplus K1, SID \oplus K2 \oplus n1) \tag{31}$$
   $$K1_{new} = Rot(K1 \oplus n1, IDS) \tag{32}$$
   $$K2_{new} = Rot(K2 \oplus IDS, K1) \tag{33}$$
   $$K2_{new} = Rot(K2 \oplus IDS, K1) \tag{34}$$
   $$SID_{new} = Rot(Rot(SID \oplus n1, IDS), K1 \oplus K2) \tag{35}$$

The subsequent discussion presents the security and performance analysis of URMAP^{+}.

Table 3 Result for 96-bit GA's of URMAP+

Equation	Value
$M_1 \oplus M_2$	47.89
$M_1 \oplus M_3$	47.9633
$M_1' \oplus M_4$	47.906
$M_2' \oplus M_1$	47.986
$M_2' \oplus M_3$	47.84
$M_2' \oplus M_4$	47.704
$M_3' \oplus M_1$	47.933
$M_4' \oplus M_1$	47.92

5.4 Security Analysis

In this section, functional, formal, and logic-based security analysis of URMAP+ is presented to evaluate the CIA services in general and assess the updated protocol's response to attacks applied on URMAP in particular.

5.4.1 Informal Analysis

This analysis includes the protocol's response to adhoc and structured attacks to verify CIA services.

1. *Confidentiality Analysis:* The following discussion proves the URMAP$^+$'s resilience to confidentiality breaches proposed in this article for URMAP.
   - Tango cryptanalysis: In the proposed update, the number of public messages is reduced by 33%, i.e., URMAP's session comprises six public messages compared to four public messages in URMAP$^+$. Reduced communication overhead implies reduced linear combinations of public messages and limited GAs. For comparison purposes, URMAP$^+$'s GAs are derived on 96-bit dataset of 1500 sessions and presented in table 3. These GAs are then used to estimate the tag's ID by eavesdropping on a single authentication session. Figure 10 shows that for values of tag identifiers as defined in URMAP tango attack in figure 4, the ID retrieval rate has reduced to 48%, proving URMAP$^+$ robustness to tango cryptanalysis.
   - Functional cryptanalysis: The analysis of URMAP reveals that the use of the AND function in $M2$ and $M6$ as define in equation (2) and (9) is a crucial factor contributing to the compromise of its confidentiality. As an unbalanced bitwise operator, the AND function creates a direct relationship between ciphertext and the secret key when a constant input is used, reducing the search complexity for adversaries. Furthermore, though lightweight, the $Rotation$ function in URMAP is vulnerable to cryptanalysis. Studies highlight its susceptibility to pattern recognition and inversion attacks [34] [35]. URMAP$^+$ replaces the AND and $Rotation$ functions with XOR and $Double, Rotation$ that forms a Substitution-Permutation Network (SPN). The update provides substitution through reversible function XOR and $Rotation$ to improve the level of confusion, contrary to irreversible substitution defined in equation (2), (9) and (12) of URMAP. Double rotation enhances diffusion, keeping implementation as lightweight as URMAP since no new primitive has been introduced. Finally, the static nature of the SID in the RFID system contributes to confidentiality breaches by creating a domino effect that compromises all tags in the identification network. Static identifiers enable tracking and correlating tags across sessions, leading to the full disclosure of associated tags. To prevent this, the static SID is replaced with dynamic identifiers that change in each session using equation (35). This approach limits the impact of a single compromised session, thereby improving the overall security of the protocol.
   - Grover's search-based brute force attack: URMAP weakness to Grover's cryptanalysis were triangular operation as defined in equation (9) i.e., AND and XOR. These operators offered limited confusion and diffusion since single bit change in PT or Key affected only the respective bit of CT. The vulnerability enabled iterative Grover search application using an IBM simulator that offers limited computational capability of only 30 qubits. In URMAP, double rotation significantly enhances CT's dependency on PT and Key, making bit-by-bit retrieval of Key impossible using a bounded adversary enabled with the simulator. This makes the protocol secure as per the Dolev-Yao model.
2. *Integrity Analysis:* In the proposed protocol, the public messages are cryptographically linked to ensure the integrity of random number $n1$ and the time stamp T_c. Moreover, because of the optimal design of the protocol's structure and messages, the adversary cannot retrieve confidential information from the public messages.
3. *Availability Analysis:*The memory architecture of the proposed protocol is inspired by URMAP, which uses timestamps to combat replay attacks. Therefore, the protocol is resilient to generalized desynchronization attacks [32]. For elaboration of the concept, consider a synchronized tag/reader pair. The adversary eavesdrops on one successful session. In the subsequent session, it blocks the tag's challenge message $M4$ for partial desynchronization. Typically adversay replaying message $M1_i$, $M2_i$,

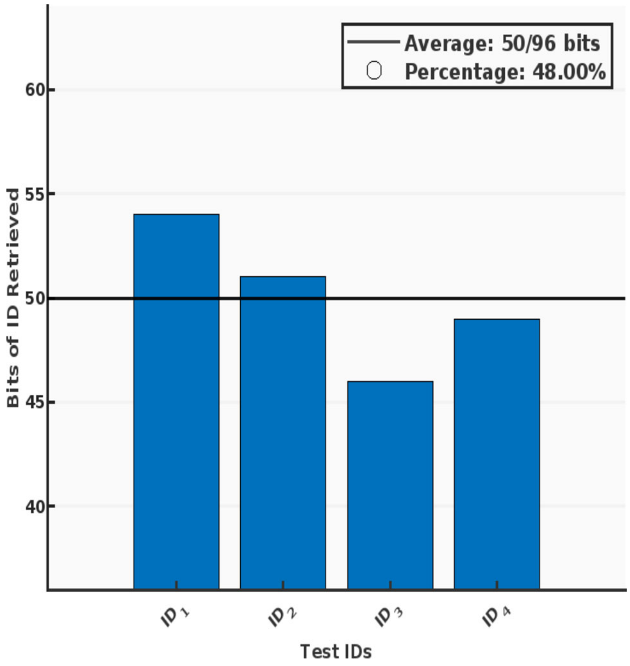


Fig. 10 URMAP$^+$ response to Tango cryptanalysis

Table 4 URMAP$^+$'s resilience to desynchronization attack

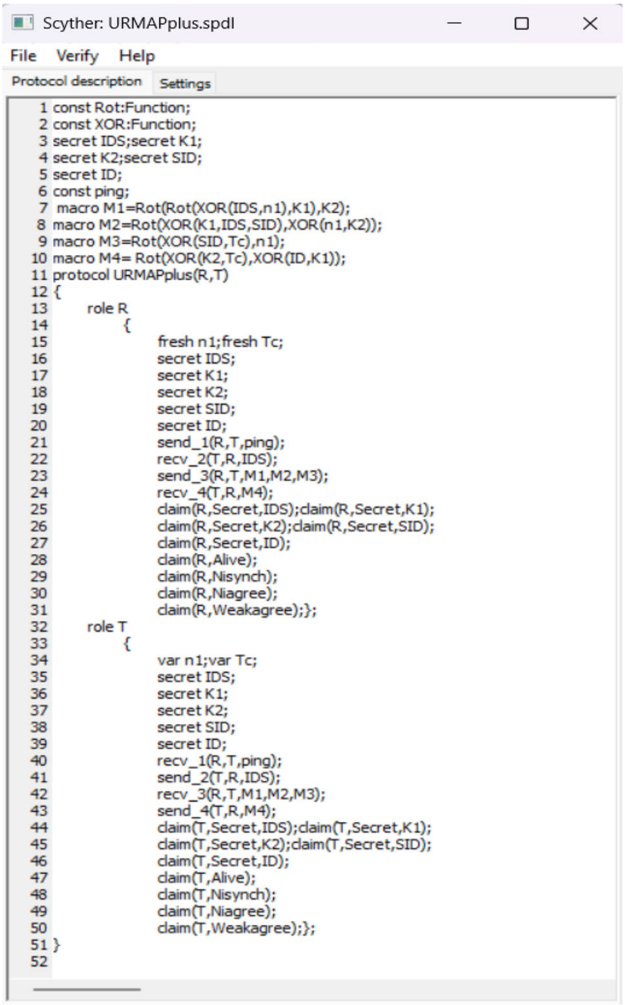
Reader		Tag	
IDS_{old}	IDS_{new}	IDS_{old}	IDS_{new}
Session i:Adversary eavesdrops public messages			
IDS_0	IDS_1	IDS_0	IDS_1
Session i+1: Adversary blocks message $M4$			
IDS_0	IDS_1	IDS_1	IDS_2
Session i+2: Adversary replays message $M1_i$,$M2_i$,$M3_i$			
IDS_0	IDS_1	IDS_1	IDS_2
Session i+3: Successful tag/reader authentication			
IDS_1	IDS_3	IDS_1	IDS_3

$M3_i$ would desynchronize the tag but in URMAP$^+$ n_{1i} will be extracted by $M1_i$, $M2_i$ will be verified. However, the outdated timestamp will terminate the session, keeping the tag/reader pair partially connected. Table 4 demonstrates the above elaborated scheme.

Complete synchronization will be regained with an uninterrupted successful authentication session between legit tag/reader pair, thus ensuring resilience to DoS.

5.4.2 Formal Analysis

In this subsection, URMAP$^+$ is formally analyzed to prove the authenticity and secrecy between the tag/reader pair using Scyther, i.e., an automated security analysis tool that uses the Dolev-Yao model [31] [36]. URMAP$^+$ is first described in the high-level descriptive format as elaborated in figure 11.



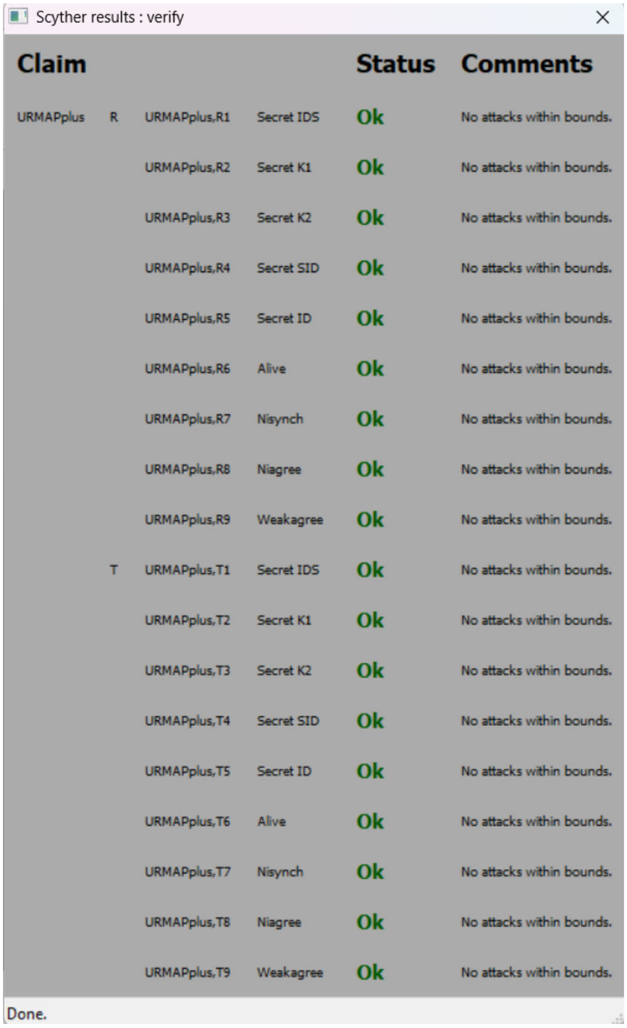
```
const Rot:Function;
const XOR:Function;
secret IDS;secret K1;
secret K2;secret SID;
secret ID;
const ping;
 macro M1=Rot(Rot(XOR(IDS,n1),K1),K2);
macro M2=Rot(XOR(K1,IDS,SID),XOR(n1,K2));
macro M3=Rot(XOR(SID,Tc),n1);
macro M4= Rot(XOR(K2,Tc),XOR(ID,K1));
protocol URMAPplus(R,T)
{
        role R
                {
                        fresh n1;fresh Tc;
                        secret IDS;
                        secret K1;
                        secret K2;
                        secret SID;
                        secret ID;
                        send_1(R,T,ping);
                        recv_2(T,R,IDS);
                        send_3(R,T,M1,M2,M3);
                        recv_4(T,R,M4);
                        claim(R,Secret,IDS);claim(R,Secret,K1);
                        claim(R,Secret,K2);claim(R,Secret,SID);
                        claim(R,Secret,ID);
                        claim(R,Alive);
                        claim(R,Nisynch);
                        claim(R,Niagree);
                        claim(R,Weakagree);};
        role T
                {
                        var n1;var Tc;
                        secret IDS;
                        secret K1;
                        secret K2;
                        secret SID;
                        secret ID;
                        recv_1(R,T,ping);
                        send_2(T,R,IDS);
                        recv_3(R,T,M1,M2,M3);
                        send_4(T,R,M4);
                        claim(T,Secret,IDS);claim(T,Secret,K1);
                        claim(T,Secret,K2);claim(T,Secret,SID);
                        claim(T,Secret,ID);
                        claim(T,Alive);
                        claim(T,Nisynch);
                        claim(T,Niagree);
                        claim(T,Weakagree);};
}
```

Fig. 11 URMAP+ elaboration in the high level descriptive language

The tool then automatically verifies secrecy, synchronization, and agreement between the tag/reader pair, given that the adversary capabilities are restricted to properties defined in the Dolev-Yao model and elaborated in section 3. The results of the automated analysis are presented in figure 12.

5.4.3 GNY Logic Analysis

Gong Needham Yahalom (GNY) logic proof verifies that the protocol achieves its goals, i.e., key distribution and mutual authentication, based on the logical rules. To implement the systematic mechanism of GNY logic analysis, terms such as formulas, statements, and postulates require elaboration. A brief description of GNY logic attributes is presented for reference [37].



Claim				Status	Comments
URMAPplus	R	URMAPplus,R1	Secret IDS	Ok	No attacks within bounds.
		URMAPplus,R2	Secret K1	Ok	No attacks within bounds.
		URMAPplus,R3	Secret K2	Ok	No attacks within bounds.
		URMAPplus,R4	Secret SID	Ok	No attacks within bounds.
		URMAPplus,R5	Secret ID	Ok	No attacks within bounds.
		URMAPplus,R6	Alive	Ok	No attacks within bounds.
		URMAPplus,R7	Nisynch	Ok	No attacks within bounds.
		URMAPplus,R8	Niagree	Ok	No attacks within bounds.
		URMAPplus,R9	Weakagree	Ok	No attacks within bounds.
	T	URMAPplus,T1	Secret IDS	Ok	No attacks within bounds.
		URMAPplus,T2	Secret K1	Ok	No attacks within bounds.
		URMAPplus,T3	Secret K2	Ok	No attacks within bounds.
		URMAPplus,T4	Secret SID	Ok	No attacks within bounds.
		URMAPplus,T5	Secret ID	Ok	No attacks within bounds.
		URMAPplus,T6	Alive	Ok	No attacks within bounds.
		URMAPplus,T7	Nisynch	Ok	No attacks within bounds.
		URMAPplus,T8	Niagree	Ok	No attacks within bounds.
		URMAPplus,T9	Weakagree	Ok	No attacks within bounds.

Done.

Fig. 12 URMAP+ claim verification results

1. *Formula:* A formula refers to a variable assuming a particular value in a session. The universal formulas used in the encryption algorithms are S: shared secret and K: encryption key. Table 6 defines a selected list of formulae along with their description.
2. *Statements:* These notions reflect some properties of the formulae. Let P and Q be communicating parties, i.e, principals, and X be a formula; table 7 elaborates some primary statements.
3. *Logical Postulates:* There are five categories of logical postulates. Table 5 defines a selection of logical postulates relevant to URMAP+ analysis.

The logic of the belief analysis process consists of six steps. The description of these steps in the context of the security analysis of URMAP$^+$ is as follows:

Table 5 GNY logic postulate description

S.no.	Postulate	Description
$T1$	$\frac{P \lhd *X}{P \lhd X}$	Being told a "not originated here" formula is a special case of being told formula
$T2$	$\frac{P \lhd (X,Y)}{P \lhd X}$	Being told a concatenate formula refers to the transmission of all the individual components i.e. X, Y
$T3$	$\frac{P \lhd [X]_K, P \ni K}{P \lhd X}$	If P is told an encrypted version of X and P possess the key K then P is being told X
$P1$	$\frac{P \lhd X}{P \ni X}$	If X is communicated to P then P possess X
$R1$	$\frac{P \mid\equiv \phi(X), P \ni K}{P \mid\equiv \phi([X]_K), P \mid\equiv \phi([X]_K^{-1})}$	If P believes that X is recognizable and P possess the key K then P also believes that $[X]_K$ and $[X]_K^{-1}$ are recognizable

Table 6 GNY logic formula descriptions

Formula	Description
$X \| \| Y$	X is concatenated with Y
$[X]_K$	bit string X is encrypted using symmetric key K
$Y = F(X)$	reversible computationally feasible function

Table 7 GNY logic statement descriptions

Statement	Description
$P \lhd X$	P is told X
$P \ni X$	P possesses X
$P \mid\equiv C$	P believes that condition C holds
$P \mid\equiv \#(X)$	P believes that X is fresh
$P \lhd (*X)$	$*$:"not originated here", P believes that it has never conveyed X which it has received

1. Formal definition of communicating parties and URMAP$^+$ public messages

$$\begin{aligned}
&Principals = Reader \Rightarrow R; Tag \Rightarrow T\\
&Msg1.\, R \Rightarrow T : hello\\
&Msg2.\, T \Rightarrow R : *IDS\\
&Msg3.\, R \Rightarrow T : M1 = *[n1]_{K1,K2}||\\
&M2 = *\, F(n1, SID, IDS, K1, K2)||M3 = *\, [T_c]_{SID,n1}\\
&Msg4.\, T \Rightarrow R : M4 = *F(T_c, K1, K2, ID)
\end{aligned} \tag{36}$$

Table 8 Performance Analysis of URMAP

	EMAP [7]	Tewari & Gupta Protocol [15]	SLAP [38]	LRSAS+ [20]	PLAKE [22]	URMAP+
Tag's Memory	$6L$	$5L$	$7L$	$3L$	$5L$	$8L$
Tag's Communication Cost	$2L$	$3L$	$1.5L$	$2L$	$3L$	$2L$
Formal Verification	No	No	No	No	Yes	Yes
Confidentiality Resilience	No[10]	No[16]	Yes	Yes	No [24]	Yes
Integrity Resilience	No [10]	No [17]	No[39]	No [21]	No [24]	Yes
Availability Resilience	No[10]	No [16]	[32]	No [21]	No[24]	Yes
Quantum Resilience	...	...	...	...	...	Yes

2. Definition of URMAP$^+$'s security goals.

$$\begin{aligned}&1.Symmetric\,key\;(n1)distribution.\\&2.Mutual\,authentication\end{aligned}\tag{37}$$

3. Define the Possession Set (PS), i.e, everything that a principal receives and generates during a session, and the Belief Set (BS), i.e, contents of the principal's memory.

$$\begin{aligned}&\quad\textbf{Reader:}\\&BS:[SID, ID, IDS, K1, K2, T_t];\\&PS:[n1, M1, M2, M3, M4]\\&\quad\textbf{Tag:}\\&BS:[SID, ID, IDS, K1, K2, T_t];\\&PS:[M1, M2, M3, M4]\end{aligned}\tag{38}$$

4. Parse every message in terms of statements defined in table 7.

$$\begin{aligned}&Msg1.\,T\triangleleft hello\\&Msg2.\,R\triangleleft *IDS\\&Msg3_1.\,T\triangleleft *[n1]_{K1,K2}\\&Msg3_2.\,T\triangleleft *F(n1, SID, IDS, K1, K2)\\&Msg3_3.\,T\triangleleft *[T_c]_{SID,n1}\\&Msg4.\,R\triangleleft *F(T_c, K1, K2, ID)\end{aligned}\tag{39}$$

5. The statements defined in step 4 are then analyzed based on the postulates defined in table 5.

$$\begin{aligned}&Msg1.\,T\ni hello\,(T1)\\&Msg2.\,R\triangleleft IDS\,(T1)\\&Msg3_1.\,T\ni n1\,(P1\,,\;T1\,and\,T3)\\&Msg3_2.\,T\mid\equiv *F(n1, SID, IDS, K1, K2)\\&\qquad(T1\,and\,R1(with\,SID, IDS, K1, K2\\&\qquad being\,recognizable\,from\,T's\,BS))\\&Msg3_3.\,T\ni T_c\,(P1\,,\;T1\,and\,T3)\\&Msg4.\,R\mid\equiv *F(T_c, K1, K2, ID)\\&\qquad(T1\,and\,R1with\,K1, K2, ID\\&\qquad being\,recognizable\,from\,R's\,BS))\end{aligned}\tag{40}$$

6. At the end of the session, the security goals of URMAP$^+$ are achieved by following the logic.

$$\begin{aligned}&a.Symmetric\,key\;(n1)\,is\\&\quad successfully\,distributed\,as\,T\ni n1.\\&b.Authentication\,accomplished\,since\\&\quad T\mid\equiv \#F(n1, SID, IDS, K1, K2)\\&\quad and\;R\mid\equiv \#F(T_c, K1, K2, ID).\end{aligned}\tag{41}$$

The GNY logic analysis proves that URMAP$^+$ securely transfers n_1 to the tag, as the tag possesses the random number n_1 at the end of the protocol. Also, the reader/tag authentication challenge messages are recognized by the intended parties, ensuring successful mutual authentication.

5.5 Performance Analysis

Performance analysis of the proposed protocol compared to prominent URMAPs is presented in table 8, which shows that URMAP$^+$ is a lightweight cipher offering adequate CIA service portfolio to the IoT track and trace applications. Prominent features supporting the CIA claims of the proposed protocol are:

1. Evaluation of UMAP on quantum cryptanalysis model.
2. Evaluation of protocol's response to generalized desynchronization attack by timestamp T_c.
3. Formal security analysis using Scyther.

Table 8 highlights that URMAP+ is the only protocol with verified quantum resistance. The dimension of security analysis has not been explored entirely. In addition, most protocols have not been verified through formal analysis, such as Scyther, rendering their security analysis incomplete.

UMAPs that fail to provide CIA services to tag/reader identification channel has the following practical implications for IoT track and trace applications:

1. Confidentiality breach described in functional analysis of URMAP may lead to counterfeit products or unauthorized access to restricted areas.
2. System vulnerable to data tempering leads to incorrect inventory management, causing disruptions and delays.
3. Systems prone to desynchronization or DoS attacks can halt critical operations, increasing the application's downtime.
4. Addressing the security breaches after the protocol has been deployed impacts the operational cost of the application.

Given the impact of vulnerable authentication protocols, designing and implementing robust UMAP is crucial.

6 Conclusion

This paper presents a technique to compromise the IoT sensing layer identification network that uses Ultralightweight RFID Mutual Authentication Protocol (URMAP) for authenticated encryption. The imbalanced nature of the protocol's primitive (AND and OR) led to the deterministic retrieval of all seven RFID tag identity attributes, i.e., (SID, ID, IDS, $K1$, $K2$, $n1$, T_{s1}). In addition to exploiting operational imbalance, Grover's search brute force model for URMAP is also presented to retrieve the exact attributes. The efficient implementation of the proposed model facilitates deterministic information retrieval even on a quantum simulator. The tag's ID is also proven vulnerable to tango cryptanalysis with an average success rate of 84.37%. Given the level of vulnerability to full disclosure attacks, URMAP is rendered ineffective, and an updated version of the protocol, i.e., URMAP$^+$, is proposed for IoT security. Security analysis demonstrates that URMAP$^+$ provides robust security in passive RFID systems. Furthermore, in light of the demonstrated feasibility of quantum cryptanalysis on UMAPs, future work must focus on developing quantum-safe authentication algorithms to preserve the security attributes in identification systems.

Appendix A Good approximation equations

Table 9 Results of 96-bit GA's of URMAP

GA's	Result 96 bits
$M1 \oplus M2$	**15.39**
$M1 \oplus M3$	**14.56**
$M1' \oplus M4$	**16.36**
$M1' \oplus M5$	**17.30**
$M2 \oplus M3$	**19.12**
$M2 \oplus M4$	82.65
$M2' \oplus M1$	75.04
$M2' \oplus M4$	**19.84**
$M2' \oplus M5$	**17.04**
$M2' \oplus M6$	49.15
$M3 \oplus M4$	80.97
$M3' \oplus M4$	**15.08**
$M3' \oplus M5$	**18.09**
$M4 \oplus M5$	**15.90**
$M4' \oplus M1$	**18.20**
$M4' \oplus M2$	**19.43**
$M4' \oplus M3$	**18.14**
$M5' \oplus M1$	**17.40**
$M5' \oplus M2$	**15.55**
$M1 \oplus M2 \oplus M3$	51.38
$M2 \oplus M3 \oplus M4 \oplus M5$	59.87
$M1 \oplus M2 \oplus M3 \oplus M4 \oplus M5 \oplus M6$	52.46

Author Contributions P.R. and M.K. wrote the manuscript text, M.H. and U.M. prepared the experimental results, and N.R. and M.N. prepared figures and tables. All authors reviewed the manuscript.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Global Market Insights. Iot-based asset tracking and monitoring market size report - (2032)
2. Krishna, P., Husak., D.: Simple lightweight rfid reader protocol. *SLRRP Working Group*, (2005)
3. Iso/iec 15963-1:2020. https://www.iso.org/standard/73195.html. Accessed: 2021-12-05
4. Robshaw, M., Williamson, T.: Rain rfid and the internet of things: Industry snapshot and security needs. In *NIST Lightweight Cryptography Workshop. NIST*, 1–13, (2015)

5. Ahson, SA., Ilyas, M.: *RFID handbook: applications, technology, security, and privacy*. CRC press, (2017)
6. Peris-Lopez, P., Hernandez-Castro, J., Tapiador, J., Ribagorda, A.: Lmap: A real lightweight mutual authentication protocol for low-cost rfid tags. 01 (2006)
7. Huang, H.: (2007) An efficient mutual authentication protocol on rfid tags. volume 4809, 550–556, 12
8. Peris-Lopez, P., Hernandez-Castro, JC., Estevez-Tapiador, JM., Ribagorda, A.: M2ap: A minimalist mutual-authentication protocol for low-cost rfid tags. In Jianhua Ma, Hai Jin, Laurence T. Yang, and Jeffrey J.-P. Tsai, editors, *Ubiquitous Intelligence and Computing*, pages 912–923, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg
9. Ticyan Li and Guilin Wang. Security analysis of two ultra-lightweight rfid authentication protocols. In Hein Venter, Mariki Eloff, Les Labuschagne, Jan Eloff, and Rossouw von Solms, editors, *New Approaches for Security, Privacy and Trust in Complex Environments*, pages 109–120, Boston, MA, 2007. Springer US
10. Tieyan Li and Robert Deng: Vulnerability analysis of emap-an efficient rfid mutual authentication protocol. **238–245**, 01 (2007)
11. Chien, H.-Y.: Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity. IEEE Trans. Dependable Secure Comput. **4**(4), 337–340 (2007)
12. Ain, QU., Mahmood, Y., Mujahid, U., islam, MN.: Cryptanalysis of mutual ultralightweight authentication protocols: Sasi and rapp. In *2014 International Conference on Open Source Systems and Technologies*, 136–145 (2014)
13. Avoine, G., Carpent, X., Martin, B.: Strong authentication and strong integrity (sasi) is not that strong. (6370) 50-64, 06 (2010)
14. Cao, T., Bertino, E., Lei, H.: Security analysis of the sasi protocol. IEEE Trans. Dependable Secure Comput. **6**(1), 73–77 (2009)
15. Tewari, A., Gupta, B.B.: Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for iot devices using rfid tags. J. Supercomput. **73**(03), (2017)
16. Khalid, M., Mujahid, U., Islam, MU.: Cryptanalysis of ultra-lightweight mutual authentication protocol for radio frequency identification enabled internet of things networks. *International Journal of Distributed Sensor Networks*, 14:155014771879512, 08 (2018)
17. Khalid, M., Mujahid, U., Islam, MN., Tran, B.: Probabilistic full disclosure attack on iot network authentication protocol. In *Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 2*, 728–738. Springer, (2020)
18. Shariq, M., Singh, K.: A novel vector-space-based lightweight privacy-preserving rfid authentication protocol for iot environment. J. Supercomput. **77**, 08 (2021)
19. Ghosh, H., Maurya, P.K., Bagchi, S.: Cryptanalysis of an rfid-enabled authentication protocol for healthcare. Wirel. Pers. Commun. **138**(4), 2613–2635 (2024)
20. Khorasgani, A., Sajadieh, M., Yazdani, M.: Novel lightweight rfid authentication protocols for inexpensive tags. Journal of Information Security and Applications **67**(103191), 06 (2022)
21. Kumar, S., Banka, H., Kaushik, B., et al.: An ultra-lightweight secure rfid authentication protocol for low-cost tags. Journal of Computer Virology and Hacking Techniques **20**, 803–818 (2024)
22. Roy, S., Das, D., Mondal, A., Mahalat, M.H., Sen, B., Sikdar, B.: Plake: Puf-based secure lightweight authentication and key exchange protocol for iot. IEEE Internet Things J. **10**(10), 8547–8559 (2023)
23. Babu, P.R., Reddy, A.G., Palaniswamy, B., Das, A.K.: Ev-puf: Lightweight security protocol for dynamic charging system of electric vehicles using physical unclonable functions. IEEE Transactions on Network Science and Engineering **9**(5), 3791–3807 (2022)
24. Lee, S.-W., Safkhani, M., Le, Q., Ahmed, O., Hosseinzadeh, M., Rahmani, A., nasour bagheri na.bagheri@gmail.com.: Designing secure puf-based authentication protocols for constrained environments. Sci. Rep. **13**(12), 11–11 (2023)
25. Sun, D.-Z., Gao, Y.-N., Tian, Y.: On the security of a puf-based authentication and key exchange protocol for iot devices. Sensors **23**(6559), 07 (2023)
26. Alhasan, A.Q.A., Rohani, M.F., Abu-Ali, M.S.: Ultra-lightweight mutual authentication protocol to prevent replay attacks for low-cost rfid tags. IEEE Access **12**, 50925–50934 (2024)
27. Chen, X.-Y., Jin, Z.-G.: Research on key technology and applications for internet of things. Phys. Procedia **33**, 561–566 (2012)
28. EPC P.C Global. Epc radio-frequency identity protocols generation-2 uhf rfid specification for rfid air interface protocol for communications at 860 mhz–960 mhz. *Version 1.0*, 23, (2013)
29. Bormann, C., Ersue, M., Keranen, A.: Terminology for constrained-node networks. *Internet Engineering Task Force (IETF): Fremont, CA, USA*, pages 2070–1721, (2014)
30. Islam, MN., Khalid, M., Mujahid, U.: Formal security analysis of generalized ultralightweight mutual authentication protocol. In *Proceedings of the Future Technologies Conference*, pages 566–573. Springer, (2023)
31. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)
32. Safkhani, M., Bagheri, N.: Generalized desynchronization attack on umap: Application to rcia, kmap, slap and sasi+ protocols. IACR Cryptol. ePrint Arch. **2016**, 905 (2016)
33. Bathe, B., Anand, R., Dutta, S.: Evaluation of grover's algorithm toward quantum cryptanalysis on chacha. Quantum Inf. Process. **20**(12), 394 (2021)
34. Avoine, G., Carpent, X., Martin, B.: Strong authentication and strong integrity (sasi) is not that strong. In *Radio Frequency Identification: Security and Privacy Issues: 6th International Workshop, RFIDSec 2010, Istanbul, Turkey, June 8-9, 2010, Revised Selected Papers 6*, pages 50–64. Springer, (2010)
35. Cao, T., Bertino, E., Lei, H.: Security analysis of the sasi protocol. IEEE Trans. Dependable Secure Comput. **6**(1), 73–77 (2008)
36. JF Cremers, C., The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper. In *International conference on computer aided verification*, pages 414–418. Springer, (2008)
37. Gong, L., Needham, RM., Yahalom, R.: Reasoning about belief in cryptographic protocols. In *S&P*, pages 234–248. Citeseer, (1990)
38. Hanguang Luo, G., Wen, J.S., Huang, Z.: Slap: Succinct and lightweight authentication protocol for low-cost rfid system. Wireless Netw. **24**(01), (2018)
39. Ciampi, G.: Cryptanalysis of the SLAP authentication protocol. *CoRR*, arXiv:abs/1906.03228 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.