ESRAS
An efficient and secure ultra-lightweight RFID authentication scheme for low-cost tags
Shariq, Mohd; Singh, Karan; Lal, Chhagan; Conti, Mauro; Khan, Tayyab

DOI
10.1016/j.comnet.2022.109360

Publication date
2022

Document Version
Final published version

Published in
Computer Networks

Citation (APA)

Important note
To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.
ESRAS: An efficient and secure ultra-lightweight RFID authentication scheme for low-cost tags

Mohd Shariq a,c, Karan Singh a, Chhagan Lal b,c, Mauro Conti b,c, Tayyab Khan a

a School of Computer and Systems Sciences, JNU, New Delhi, India
b Department of Intelligent Systems, TU Delft, Netherlands
c Department of Mathematics, University of Padova, Italy

A B S T R A C T

Internet of Things (IoT) technologies rapidly evolve and are used in many real-life applications. One of the core technologies used in various IoT applications is Radio Frequency Identification (RFID) technology. RFID wirelessly and uniquely identifies the tagged objects without a direct line of sight. However, the research community had reported privacy and security-related concerns in RFID systems, where an adversary may tamper, eavesdrop, add, delete, or even modify the transmitted messages over an insecure communication channel.

To address the issues mentioned above, we propose an Efficient, Secure, and practical ultra-lightweight RFID Authentication Scheme (ESRAS), which uses rank operation for low-cost tags. We utilize a simplistic bitwise exclusive-OR, circular left rotation, and a newly proposed ultra-lightweight rank operation to provide high security at low cost. The analysis of ESRAS concerning its security and performance shows that it effectively resists several known attacks and is relatively superior to the existing schemes regarding the computational and storage costs. Moreover, ESRAS shows more suitability for low-cost RFID tags and can be executed in a multimedia big data environment.

1. Introduction

Internet-of-Things (IoT) is an umbrella keyword that covers several aspects associated with the extension of the Internet [1]. Over the past years, among various new technologies, IoT has emerged as a major topic researched in academia and industry, aiming to benefit society. IoT technology enables physical objects to exchange and collect data through the Internet [2]. The devices are embedded with various technologies and entities like RFID tags, actuators, sensors, and network connectivity for communicating with the external environment or other devices [3]. RFID technology is rapidly growing and becoming more popular with contactless technology. It has seen a tremendous increase in its applications used in the Automatic Identification and Data Capturing (ADIC) of any targeted object(s) with no direct line-of-sight or contact [4]. In addition to this, it can read the data remotely and automatically through radio frequency signals. One of the first uses of RFID was seen during World War Second in Identify Friend or Foe (IFF) aircraft systems [5]. Since then, several traditional automatic identification technologies have been used in various applications. These techniques include barcode recognition, biometric identification, optical character recognition, and magnetic card identification systems. However, these technologies are still unsatisfactory due to their several limitations. For example, the barcode system needs a direct line-of-sight contact between tags and readers for object identification and storing a small amount of data. While a biometric identification system is expensive for specific purposes, the optical character recognition cost is too high, and magnetic card identification needs close contact to identify an object. In contrast, RFID has numerous advantages over these traditional identification technologies. Therefore, RFID is considered one of the prominent technology in the twenty-first century [6,7]. RFID technology is widely used in various real-life applications such as e-libraries, e-payment, animal identification, personnel identification, smart healthcare systems, human implantation, access control, and Internet of Vehicles (IoV), etc [8–10].

RFID systems typically contain three main components: tags, readers, and backend servers. Each RFID tag has two key elements: a silicon microchip and an antenna. The microchip stores and processes data, while an antenna transmits and receives data to the RFID reader via radio signals [11]. The RFID tags consist of mainly three categories...
such as active, semi-active, and passive. Nowadays, primarily passive tags are being used because these are relatively cheaper as compared to active tags. Also, they have no battery and get charged through Radio Frequency (RF) waves from the RFID reader. Instead, the active tags get power, are charged with their battery or energy sources, and transmit signals periodically. The semi-active tags get power and are charged with internal energy sources [11]. Accordingly, the RFID tags can be operated at the three different types of frequency ranges, which includes the low-frequency range (LF: 125 kHz to 134 kHz, reading range: ~10 cm), high-frequency range (HF: 13.56 MHz, reading range: ~1 m), and ultra-high frequency range (UHF: 860 to 960 MHz, reading range: 10 to 15 m) [12]. The RFID reader consists of three components: RF signal generator, microcontroller, and receiver or signal detector. It is used to read/write the secret information of the tags over an insecure communication channel and further transmit it into the database of the backend server [13–15]. The RF signal generator produces radio waves, and an antenna is used to transmit them. Also, the reader receives feedback signals which come from the tag. The receiver/signal detector processes the information which is sent by the RFID tags. The backend server is connected with the reader and stores the associated secret information of the tags affixed to objects.

Typically, RFID authentication protocols can be classified into four categories depending upon the supported operations on the tags, the computational cost of the tags, and cryptographic primitives employed on the tags [16].

- **Full-fledged protocols** These support classical cryptographic primitives such as one-way hashing, symmetric/asymmetric encryption, and even private/public key algorithms [17,18].
- **Simple protocols** These provide support for one-way cryptographic hash and Random Number Generator (RNG) on the tags [19,20].
- **Lightweight protocols** These support RNG and Cyclic Redundancy Check (CRC) checksum on the tags. On the other hand, these protocols do not employ hash functions [21,22].
- **Ultra-lightweight protocols** These support bitwise AND, OR, and XOR operations on the tags. Besides, the RNG cannot be employed on the tags [23–25].

As one of the key technologies used in several IoT domains, the main idea behind RFID is to provide secure and reliable access to the information or messages exchanged over secure channels. The sensitive information associated with particular things or objects should be securely delivered over RFID systems. In an RFID system, mutual authentication is considered one of the primary security requirements that should be accomplished between tag and server over a secure communication channel. RFID authentication schemes must always be secure, efficient, and robust against well-known malicious security attacks [26]. However, several security attacks and privacy issues may occur while exchanging messages between tags and readers. Our proposed scheme satisfies various security and privacy requirements to ensure robust, efficient, and secure RFID communications. Besides, the RFID tags are resource-constrained devices with less computing power, limited computational resources, and low memory space. Therefore, an RFID system typically cannot withstand the traditional known cryptographic primitives/algorithms, namely symmetric key and asymmetric key cryptographic techniques. To overcome such issues, our proposed authentication scheme exploits bitwise exclusive-OR (⊕), circular left rotates, and newly proposed ultra-lightweight rank operations to encrypt data instead of hash functions, ensuring less computational and storage overheads on the tags while supporting a higher level of security. In addition, the proposed scheme meets essential security requirements and protects from various security attacks.

To establish a secure RFID authentication system, we present the key objectives, which can be regarded as the main criteria for designing our proposed scheme. These objectives are as follows.

1. Accomplish mutual authentication between communicating entities.
2. Achieve forward security.
3. Resist several known attacks like tag tracking, disclosure, and de-synchronization.
4. Minimize the storage and communication cost on tags with higher efficiency.

The key contributions of our proposed scheme are as follows.

1. We propose an efficient and secure ultra-lightweight RFID authentication scheme suitable in IoT scenarios. In particular, we exploit bitwise exclusive-OR (⊕), circular left rotates \( \text{Rot}(-,\cdot) \), and newly proposed ultra-lightweight rank \( \text{Rank}(-,\cdot) \) operations for minimizing the computational overhead on tags.
2. The proposed scheme accomplishes mutual authentication and provides resistance against several attacks, including de-synchronization and tag location tracking attacks.
3. We implement our proposed scheme, and a performance comparison has been made with the other similar existing schemes to show that our scheme greatly overcomes the storage and communication cost of tags while maintaining the key security requirements.

In Section 2, we describe previous schemes along with their limitations. The notations used and preliminaries are given in Section 3. The methodology of our proposed efficient and secure ultra-lightweight RFID authentication scheme using rank operation for low-cost tags is detailed in Section 4. Next, the informal security analysis is presented following the ESRAS’s performance evaluation in Section 5. Section 6 provides the results obtained using Sceythr simulation and the performance evaluation. Finally, Section 7 provides the concluding remark.

2. Related work

RFID is considered a promising technology for ubiquitous environments, allowing almost every object to be wirelessly identified via Radio Frequency (RF) waves. Over the past decades, researchers have proposed many schemes to achieve secure communication for safeguarding RFID systems from several known attacks and privacy concerns. However, it is difficult to ensure security and privacy requirements in low-cost RFID systems because of insecure communication among tags and readers. Next, we discuss relevant state-of-the-art RFID authentication schemes with their pros and cons as shown in Table 1.

Peris-Lopez et al. [41–43] presented a class of Ultra-lightweight protocols called UMAP. The protocols are chronological, Ultra-lightweight, lightweight, minimalistic, and efficient mutual authentication protocols namely, LMAP [41], \( M^2\text{AP} \) [42], and EMAP [43], respectively. These protocols utilized simple bitwise OR, exclusive-OR (⊕), and \( \text{sum mod} \) operations to achieve a low-cost computation cost. Also, all these operations were efficient and cost-effective for passive RFID tags. However, Wang-Li [24] and Li-Deng [25] showed that the families of UMAP protocols were insecure against the full disclosure and the de-synchronization attacks.

In [44], an enhancement of RAPP [31] to fix the security weaknesses of the de-synchronization attack on RAPP. The protocol utilizes XOR (⊕), permutation \( \text{Per}(-,\cdot) \), and build-in CRC-16 operations on the tag. Besides, the protocol utilizes a secret key backup mechanism to improve security functionalities with no increment in performance cost. The protocol provides security against tag tracing, replay, secret disclosure, and de-synchronization attacks. Later on, the protocol in [45] pointed out that the shared secret key between \( T \) and \( BS \) can be easily obtained by an adversary. Thus, the protocol [44] is insecure against de-synchronization attacks.
still vulnerable to de-synchronization attacks. However, the protocol fusion, irreversibility, and low complexity. In addition, the protocol is system associated with these properties, including sensibility, full con-

The reader remotely connects with the backend server by employing 4G and 5G technology to increase the mobility of RFID systems. Vulnerable to traceability attacks.

The Rabin algorithm only performs square and modulo operations to raise a higher level of security for the public key level. Vulnerable to tag impersonation attacks.

The scheme shows strong computational efficiency for RFID systems. The scheme does not take advantage of computation, communication, and storage costs.

The scheme is well-suitable for low-cost passive tags. Vulnerable to de-synchronization and secret disclosure attacks.

The scheme avoids unbalanced OR and AND operations. Insecure against de-synchronization attacks.

The scheme is well-suitable for low-cost RFID tags. Vulnerable to de-synchronization and replay attacks.

The scheme uses a sub-key key updating phase. The scheme improves efficiency and enhances a higher level of security for resource-constrained RFID systems.

The scheme is more suitable for low-cost RFIDs. The scheme uses fewer resources on tags which is better suited for low-cost RFID systems.

The scheme strongly avoids unbalanced AND, OR operations and proposes \( K_i(X) \). Vulnerable to impersonation and de-synchronization attacks.

The scheme ensures data confidentiality, integrity, and tag anonymity features. Vulnerable to secret disclosure and de-synchronization attacks.

The scheme provides stronger security also better suited for low-cost passive RFID tags. Vulnerable to impersonation and de-synchronization attacks.

The scheme is more suitable for low-cost RFID tags. The scheme is well-suitable for low-cost passive tags.

The scheme consumes high storage overhead on the tag. The scheme is highly secure against impersonation attacks.

In [38], a new succinct and Ultra-lightweight RFID authentication protocol called SLAP is presented. The protocol uses simple exclusive-OR, circular left rotate \( Rot(-, \cdot) \), and conversion \( Con(-, \cdot) \) operations. The passive tags were suitable for implementation using these operations. The conversion operation ensures a security guarantee of the RFID system associated with these properties, including sensibility, full confusion, irreversibility, and low complexity. In addition, the protocol is insecure against replay, traceability, and de-synchronization attacks. Later, Safrkhan and Bagheri [46] presented a protocol [38] that is still vulnerable to de-synchronization attacks. However, the protocol in [47] also pointed out that the protocol [38] is not secure under impersonation attacks.

In [36], an Ultra-lightweight RFID protocol called KMAP. To prevent unbalanced logical AND and OR operations, the protocol uses simple exclusive-OR \((\oplus)\), circular left rotate \( Rot(-, \cdot) \), and pseudo-Kasami code \((K_i)\) operations. Besides, the protocol resists replay, de-

The table above provides a comparison among various RFID authentication schemes.

<table>
<thead>
<tr>
<th>References</th>
<th>Classes</th>
<th>Primitives</th>
<th>Strengths</th>
<th>Shortcomings</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>Lightweight</td>
<td>( XOR, Rot(-, \cdot), Perm(-, \cdot), E(k_0) \mod n )</td>
<td>The scheme consumes lower resource on tags which is better suited for low-cost RFID systems</td>
<td>Insecure against tag tracking, traceability, anonymity, disclosure, impersonation, de-synchronization, and replay attacks.</td>
</tr>
<tr>
<td>[27] SCHEME I</td>
<td>Lightweight</td>
<td>( XOR, PRNG(-, \cdot), h(k), g )</td>
<td>The protocol is secure against tag tracking, traceability, and de-synchronization attacks.</td>
<td>Vulnerable to traceability attacks.</td>
</tr>
<tr>
<td>[27] SCHEME II</td>
<td>Lightweight</td>
<td>Square and Modulo operations ((x^2 \equiv k \mod n/m))</td>
<td>The protocol provides stronger security also better suited for low-cost passive RFID tags.</td>
<td>Vulnerable to impersonation and de-synchronization attacks.</td>
</tr>
<tr>
<td>[26]</td>
<td>Lightweight</td>
<td>ECC-based ( E(k_0) \mod n/m, f(x), (k, n, SS) )</td>
<td>The protocol is more suitable for resource-constrained RFID systems.</td>
<td>Lack of security simulation tools including AVISPA, CryptoVerif, Scyther, etc.</td>
</tr>
<tr>
<td>[29]</td>
<td>Lightweight</td>
<td>( XOR, Rot(-, \cdot), Inv(-, \cdot), 2^m (+) )</td>
<td>The protocol is more suitable for resource-constrained RFID systems.</td>
<td>Vulnerable to tag impersonation attacks.</td>
</tr>
<tr>
<td>[30]</td>
<td>Lightweight</td>
<td>( XOR, PRNG(-, \cdot), En(X) )</td>
<td>The protocol is more suitable for resource-constrained RFID systems.</td>
<td>Vulnerable to impersonation and de-synchronization attacks.</td>
</tr>
<tr>
<td>[31]</td>
<td>Ultralightweight</td>
<td>( XOR, Rot(-, \cdot), Perm(-, \cdot) )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to de-synchronization and secret disclosure attacks.</td>
</tr>
<tr>
<td>[32]</td>
<td>Ultralightweight</td>
<td>( XOR, Mix(-, \cdot), Sep(-, \cdot) )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to de-synchronization and replay attacks.</td>
</tr>
<tr>
<td>[33]</td>
<td>Ultralightweight</td>
<td>( XOR, Perm(-, \cdot), build-in CRC = 16 function )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to de-synchronization attacks.</td>
</tr>
<tr>
<td>[34]</td>
<td>Ultralightweight</td>
<td>( XOR, modulu 2^l (+) ), Reverse Rotation ((RR))</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to secret disclosure attacks.</td>
</tr>
<tr>
<td>[35]</td>
<td>Ultralightweight</td>
<td>( XOR, CRC, Syndrome decoding )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to tag traceability attacks and tag impersonation attacks.</td>
</tr>
<tr>
<td>[36]</td>
<td>Ultralightweight</td>
<td>( XOR, Rot(-, \cdot), K_i(X) )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to impersonation and de-synchronization attacks.</td>
</tr>
<tr>
<td>[37]</td>
<td>Ultralightweight</td>
<td>( XOR, Rot(-, \cdot) )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to impersonation and de-synchronization attacks.</td>
</tr>
<tr>
<td>[38]</td>
<td>Ultralightweight</td>
<td>( XOR, PRNG(-, \cdot), Rot(-, \cdot), f(x), modulo 2^m (+) )</td>
<td>The protocol is more suitable for resource-constrained RFID systems.</td>
<td>Lack of security simulation tools including AVISPA, CryptoVerif, Scyther, etc.</td>
</tr>
<tr>
<td>[40]</td>
<td>Ultralightweight</td>
<td>( XOR, Rot(-, \cdot), Mix(-, \cdot) )</td>
<td>The protocol is more suitable for low-cost passive tags.</td>
<td>Vulnerable to de-synchronization and replay attacks.</td>
</tr>
</tbody>
</table>
In [37], a novel Ultra-lightweight protocol is introduced for IoT devices by employing RFID tags. The protocol utilizes exclusive-OR and circular left rotation. The protocol is secure under de-synchronization, tag tracking, and disclosure attacks and ensures data confidentiality, integrity, and tag anonymity properties. Later on, Wang et al. [48] pointed out that the shared secret key between T and BS can be obtained by an adversary. Hence, the protocol [37] is vulnerable to de-synchronization and disclosure attacks.

Zheng et al. [49] proposed a mobile RFID authentication scheme for the smart campus. The scheme employs one-way hash H(·), XOR (∧), and series operation (∧). It also guarantees known security features such as counterfeit, eavesdropping, tag location tracking, replay, Main-In-The-Middle (MITM), Denial-of-Service (DoS), and de-synchronization attacks. Later on, Safkhani and Vasilakos [50] pointed out that the protocol is susceptible to replay, tag tracking, and DoS attacks. Subsequently, the protocol utilizes operations of square and modulo such that

\[ f(x) \equiv x^2 \pmod{L}, \quad PRNG(\cdot), \quad \text{and} \quad Rot(\cdot) \]

in this section. To achieve higher security, simple bitwise XORing (∧), pseudo random number generation at reader, pre-shared secret keys of tags shared with the backend server, and SKINNY algorithm comprise three phases: setup, round function, and key updating. The protocol guarantees known security functionalities, including confidentiality, integrity, tag anonymity, data confidentiality, forward security, and can also resist several security attacks.

In [27], two RFID-based protocols are presented using one-way hash and Rabin Public Key (RPK) cryptosystem. The hashing protocol utilizes a secure one-way hash h(·), PRNG function, secret keys, and index grouping (g_j) on the RFID tag. In the hashing protocol, the index grouping number associated with the last successful authentication and the key groups associated with the previous three successful authentications are stored in the database. Later on, the hash-based protocol is susceptible to tag traceability attacks reported in [51]. In contrast, in RPK-based protocol, RPK is a symmetric cryptographic approach that utilizes operations of square and modulo such that (x^2 \equiv k \mod n/m). Such operations verify the Rabin encryption algorithm process and resist replay, tag tracking, and DoS attacks. Subsequently, the protocol is susceptible to tag impersonation attacks reported in [51].

In [29], a lightweight authentication protocol for passive RFID tags is introduced. The scheme uses inverse operations, XORing, circular shift, and addition modulo 2^{(+)}(+) operation. The protocol guarantees several security features, including mutual authentication, location privacy or untraceability, data integrity, tag anonymity, data confidentiality, forward security, and resisting known attacks.

Xiao et al. [30] presented a block cipher-based RF authentication protocol called LRSAS. The protocol uses some operations such as simple bitwise XORing (∧), PRNG(·), and SKINNY encryption algorithm. The SKINNY algorithm comprises three phases: setup, round function, and key updating. The protocol guarantees known security functionalities, including confidentiality, integrity, forward security, tag tracking, de-synchronization, and tag impersonation attacks. Later, Trinh et al. [52] reported that the protocol is susceptible to de-synchronization and secret disclosure attacks.

### 3. Preliminaries

We present the procedure of rank operation and circular rotation operations in this section. To achieve higher security, simple bitwise exclusive-OR, circular left and right, and newly proposed Ultra-lightweight Rank(X, Y) operation are used while designing an efficient and secure Ultra-lightweight RFID authentication scheme named ES-RAS. Table 2 shows all the necessary parameters with their description.

#### 3.1. Circular left rotate operation

The proposed scheme uses two circular left and right rotation operations. In general, the rotation operation can be denoted Rot(X, Y). However, Rot(X, Y) does left rotate string X by \( \text{rank}(y) \) mod L bits, where L denotes the bit length of X. The rank of X or Y is defined as the number of 1’s presented in string X or Y. For example, considering X and Y are two 8-bit length strings as follows:

\[
X = 01011100, \quad Y = 10101100
\]

Now, we compute the left rotate operation on X and Y as follows:

\[
\text{Rot}(X, Y) = \text{String } X \text{ is left rotated by } \text{rank}(Y), \text{where } \text{rank}(Y) = 4. \text{ Now, }\text{wrt}(Y)(\text{mod } 8) = 4 \mod 8 = 4. \text{ Hence, } \text{Rot}(X, Y) = \text{Rot}(X, 4) = 1100011011
\]

#### 3.2. Definition of rank and nullity

The rank and nullity of the string are defined as the number of 1’s and 0’s present in the given strings. Suppose that X and Y are the given strings, the rank and nullity of X and Y can be defined as

\[
\text{rank}(X \text{ or } Y) = \text{Number of } 1 \text{’s presents in string } X \text{ or } Y, \\
\text{nullity}(X \text{ or } Y) = \text{Number of } 0 \text{’s presents in string } X \text{ or } Y
\]

Considering X and Y are two n-bit length strings can be defined as,

\[
X = x_1x_2 \ldots x_n, \quad x_i \in \{0,1\}, \quad t = 1, 2, \ldots, n. \\
Y = y_1y_2 \ldots y_n, \quad y_j \in \{0,1\}, \quad t = 1, 2, \ldots, n.
\]

\[
\text{Rank}(X, Y) = Z = z_1z_2 \ldots z_k \\
= W''\oplus W', \quad z_k \in \{0,1\}, k = 1, 2, \ldots, n.
\]

To achieve full confusion, a new rank \( \text{Rank}(X, Y) \) operation is used for hiding the information presented in the X and Y. The new proposed Ultra-lightweight rank operation consists of four steps: rank and nullity, grouping, swapping, and composition. To better understand this operation, an example is illustrated below. We are assuming that X and Y are two 32-bit strings. Assuming the value of the threshold is \( T_h = 6 \).

\[
X = 110001110101110101100011101011110. \\
Y = 10111101110110001110110011
\]

All used operations are Ultra-lightweight. Thus these can be easily implemented on resource-constrained, low-cost RFID tags. The computational complexity of the rank operation depends on the threshold \( T_h \) value. If the value of \( T_h \) is small, the confusion will become greater. Therefore, we will suggest a larger value of \( T_h \) than 5.

**Step 1: Rank and nullity** Find the rank and nullity of both strings X and Y with respect to 1’s and 0’s that appear in the given strings.

\[
\text{rank}(X) = \text{Number of } 1 \text{’s presents in } X = 20 \text{ and}
\]

<table>
<thead>
<tr>
<th>Notations and their descriptions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notations</td>
</tr>
<tr>
<td>( T, R, BS )</td>
</tr>
<tr>
<td>ID</td>
</tr>
<tr>
<td>IDS</td>
</tr>
<tr>
<td>R_d, R_s</td>
</tr>
<tr>
<td>K_1, K_2</td>
</tr>
<tr>
<td>\text{Rank}(X, Y)</td>
</tr>
<tr>
<td>\text{Rank}(X \text{ or } Y)</td>
</tr>
<tr>
<td>\text{nullity}(X \text{ or } Y)</td>
</tr>
<tr>
<td>\text{Rot}(X, Y)</td>
</tr>
<tr>
<td>isb</td>
</tr>
<tr>
<td>msb</td>
</tr>
<tr>
<td>T_l</td>
</tr>
<tr>
<td>@</td>
</tr>
<tr>
<td>?</td>
</tr>
</tbody>
</table>
nullity($X$) = Number of 0's presents in $X = 12$.

nullity($Y$) = Number of 1's presents in $Y = 19$ and

nullity($Y$) = Number of 0's presents in $Y = 13$.

**Step 2: Grouping** First, all the given strings (i.e., $X$ and $Y$) are divided into several small blocks by using the rule of segmentation, which is based on the $rank(X)$ and $rank(Y)$, respectively. In particular, a parameter threshold $T_h$ is used to limit the size of each substring. Let us consider that $rank(X)$ is $m$ such that $m \leq n$, so the two substrings of $X$ after division are given as $X_1 = x_n x_{n-1} \ldots x_{m+2} x_{m+1}$, $X_2 = x_m x_{m-1} \ldots x_2 x_1$. Then, the division operation on substrings $X_1$ and $X_2$ will be continued by using the same segmentation rule based on their $rank(X)$ until the length of all the substrings $X_j (j \leq n)$ are less than the threshold $T_h$. Similarly, the same rule is performed on string $Y$ divided into $Y_i (k \leq n)$ by using the same segmentation rule. The computations on strings $X$ and $Y$ are shown in Figs. 1 and 2.

**Step 3: Swapping** In Table 3, we divide the above-obtained strings $X'$ and $Y'$ into two blocks according to nullity($X$) and nullity($Y$) given in Step 1, respectively. After that, we perform a swap operation on both parts to obtain $X''$ and $Y''$ as shown in Table 4.

**Step 4: Composition** To compute the final computation of the rank operation, we first perform $not(\sim)$ operation on $X''$ and $Y''$. Thereafter, the simple bitwise exclusive-OR operation is performed between them. Then, we have

$\sim X'' = 100001100100100110010011011$

$\sim Y'' = 00110010011010011001010001011101$

$R = \sim X'' \oplus \sim Y''$

$\therefore rank(R) = 17$ and nullity($R$) = 15.

The aforementioned rank operation is well-suitable and can be easily realized in low-cost RFID tags. The proposed rank operation has four major properties, which are as follows:

- **Full confusion** For each given input, they are confused by the other one and have no predicted or fixed bit produced by the rank operation. Therefore, an adversary cannot obtain one bit of useful or even sensitive information from the output to predict the input.
M. Shariq et al.

4. Proposed scheme

The proposed scheme consists of two phases: initialization and authentication.

4.1. Valid assumptions considered

The underlying assumptions on which the proposed scheme operates are as follows:

(1) An adversary can mimic a genuine tag or reader.
(2) An adversary can tamper, modify, intercept, add, and/or even delete messages during the messages exchanged among RFID components.
(3) The communication channel is considered to be secure between $\mathcal{R}$ and $\mathcal{BS}$ as well as insecure between $\mathcal{T}$ and $\mathcal{R}$, i.e., all attacks are possible.
(4) The backend server is a fully trusted entity; an adversary cannot reveal the tags’ information.

4.2. Initialization phase

This phase defines some underlying statements, which are as follows:

(1) Initially, the manufacturer stores a unique static identification number $ID$ and an index $IDS_{old}$ in each tag’s internal memory.
(2) For each $\mathcal{R}$, the manufacturer stores two secret keys $K_1$ and $K_2$, a new and old pseudonym $IDS_{new}$ and $IDS_{old}$ in $\mathcal{BS}$. Initially, $IDS_{old} = 0$ and $IDS_{new} = Null$.
(3) Each reader consists of two $\mathcal{PRNG}$. In contrast, the backend server has no such limitations.
(4) The other relevant tag’s information is stored at $\mathcal{BS}$.

4.3. Authentication phase

Fig. 3 shows the detailed process of the authentication phase of ESRAS scheme which is executed in the following steps.

**Step 1:** $M_1 : \mathcal{R} \rightarrow \mathcal{T} : \{"Hello"\}$

The reader $\mathcal{R}$ initiates an authentication session by sending a “Hello” message to $\mathcal{T}$.

**Step 2:** $M_2 : \mathcal{T} \rightarrow \mathcal{R} : \{IDS\}$

After receiving, $\mathcal{T}$ sends an index pseudonym $IDS$ to $\mathcal{R}$.

**Step 3:** $M_3 : \mathcal{R} \rightarrow \mathcal{T} : \{A \parallel B_{L or R}\}$

Upon receiving $IDS$, $\mathcal{R}$ uses the received $IDS$ as an index to search the secrets of tags in the backend server database. If it finds a match in the database, then $\mathcal{R}$ generates a $L$-bit pseudo-random number $R_1$ and computes $A$ and $B$. After that, $\mathcal{R}$ sends the messages $A \parallel B_{L or R}$ to $\mathcal{T}$, where $B_L$ and $B_R$ represent the left and right half of the string $B$.
respectively. The $B_L$ or $B_R$ of string $B$ is transmitted to the tag based on the rank of $B$ (if $\text{Rank}(B)$ is odd, then sent $B_L$, else sent $B_R$).

- Computes $A = \text{Rank}(\text{Rot}(K_1, K_2), K_1 \oplus R_1)$.
- Computes $B = \text{Rank}(\text{Rot}(K_1, R_1), K_1 \oplus K_2) \oplus \text{Rot}(\text{Rank}(K_1, R_1) \oplus K_2, K_2)$.

**Step 4: $M_4$: $T \rightarrow R : \{C_L \lor R\}$**

After receiving $A \parallel B_L \parallel R$, the tag extracts $R_1$ from $A$ by XORing $\text{Rank}(\text{Rot}(K_1, K_2), K_1)$ with $A$. After that, the tag computes a local value of $B'$ and verifies whether $B'_L \lor R ? = B_L \lor R$ if so, then $T$ authenticates $R$ as a legitimate reader and computes the response message $C$. $T$ sends the response message $C_L \lor R$ to $R$, where $C_L$ and $C_R$ represent the left and right half of the string $C$, respectively.

- Extracts $R_1 = \text{Rank}(\text{Rot}(K_1, K_2), K_1) \oplus A$.
- Compute $B' = \text{Rank}(\text{Rot}(K_1, K_2), K_1 \oplus K_2) \oplus \text{Rot}(\text{Rank}(K_1, R_1) \oplus K_2, K_2)$.
- Verify $B'_L \lor R = B_L \lor R$.
- Computes $C = \text{Rank}(K_1 \oplus K_2, R_1), \text{Rank}(R_1 \oplus K_2) \parallel \text{ID}$.

**Step 5: $M_4$: $R \rightarrow T : \{D \parallel E_L \lor R\}$**

Upon receiving $C_L \lor R$, $R$ computes a local value of $C'$ and verifies whether $C'_L \lor R ? = C_L \lor R$, if so, then $R$ authenticates $T$ as a legitimate tag and updates its index pseudonyms $\text{IDS}_{\text{old}}$ and $\text{IDS}_{\text{new}}$ in the database. After that, the reader generates $L$-bit pseudo-random number $R_1$ and computes the response messages $D$ and $E$. The reader $R$ sends the response messages $D \parallel E_L \lor R$ to the tag $T$ where $E_L$ and $E_R$ represent the left and right half of the string $E$, respectively.

- Computes $C' = \text{Rank}(\text{Rot}(K_1 \oplus K_2, R_1), \text{Rank}(R_1 \oplus K_2) \parallel \text{ID})$.
- Verify $C'_L \lor R \parallel C_L \lor R$ and updates the index $\text{IDS}_{\text{old}} = \text{IDS}$ and $\text{IDS}_{\text{new}} = \text{Rank}(\text{Rot}(\text{IDS} \parallel R_1, K_1, K_2))$.
- Computes $D = \text{Rank}(R_1, K_1 \oplus K_2) \oplus \text{Rank}(K_1, K_2) \parallel R_2$.
- Computes $E = \text{Rank}(\text{Rot}(R_1, K_2, K_2), \text{IDS}_{\text{new}} \parallel \text{Rot}(\text{Rank}(R_1 \oplus K_2, R_2) \parallel K_2)$.
- Verify $E'_L \lor R ? = E_L \lor R$ and updates the index $\text{IDS} = \text{IDS}'$.

**6. Analysis and evaluation**

The security analysis and the performance evaluation of ESRAS are illustrated in Tables 5 and 6. We have compared our proposed scheme with various existing Ultra-lightweight RFID authentication schemes. These schemes are URASP [8], IOLAS [29], LRSAS [30], RAPP [31], RAPLT [32], KMAP [36], Tewari–Gupta [37], SLAP [38], LMAP [41], M²AP [42], EMAAP [43], LPCP [44], SASI [53], David–Prasad [54], Gossamer [55], RRAP [56], RCIA [57], and URMAP [58], respectively.

**5.1. Informal security analysis**

The comparative informal security and privacy analysis of ESRAS are illustrated in Table 5.
Security and privacy comparison among various Ultra-lightweight authentication schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Mutual authentication</th>
<th>Forward security</th>
<th>Resistance to tag tracking</th>
<th>Resistance to de-synchronization attacks</th>
<th>Resistance to disclosure attacks</th>
<th>Security of the diffusion function</th>
</tr>
</thead>
<tbody>
<tr>
<td>URASP [8]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>IOLAS [29]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>LRSAS [30]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>RAPP [31]</td>
<td>No</td>
<td>📙</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>📙</td>
</tr>
<tr>
<td>RAPLT [32]</td>
<td>No</td>
<td>📙</td>
<td>No</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>KMAP [36]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>Tewari-Gupta [37]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>LMAP [41]</td>
<td>📙</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>M2AP [42]</td>
<td>📙</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>EMAP [43]</td>
<td>📙</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LPCP [44]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>SASI [53]</td>
<td>📙</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>David-Prasad [54]</td>
<td>📙</td>
<td>No</td>
<td>No</td>
<td>📙</td>
<td>📙</td>
<td>📙</td>
</tr>
<tr>
<td>Gossamer [55]</td>
<td>📙</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RAP [56]</td>
<td>Yes</td>
<td>📙</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RCIA [57]</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>URMAP [58]</td>
<td>Yes</td>
<td>📙</td>
<td>📙</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ours</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Acronyms: Yes: Denotes satisfied; No: Denotes not satisfied; 📙: No discussion.

Performance comparison among various Ultra-lightweight authentication schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computational operations on the tag side</th>
<th>Total no. of messages for mutual authentication</th>
<th>Communication messages generated on tag</th>
<th>Storage cost on tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>URASP [8]</td>
<td>@, Per, Rot</td>
<td>4L</td>
<td>1.5L</td>
<td>4L</td>
</tr>
<tr>
<td>IOLAS [29]</td>
<td>@, Rot, Inc.,+</td>
<td>4L</td>
<td>2L</td>
<td>5L</td>
</tr>
<tr>
<td>LRSAS [30]</td>
<td>@, Em, X</td>
<td>5L</td>
<td>2L</td>
<td>3L</td>
</tr>
<tr>
<td>RAPP [31]</td>
<td>@, AND, O.R., Rot, Per</td>
<td>5L</td>
<td>2L</td>
<td>5L</td>
</tr>
<tr>
<td>RAPLT [32]</td>
<td>@, Mer, Sep</td>
<td>4L</td>
<td>3L</td>
<td>5L</td>
</tr>
<tr>
<td>KMAP [36]</td>
<td>@, Rot, K</td>
<td>4L</td>
<td>2L</td>
<td>7L</td>
</tr>
<tr>
<td>Tewari-Gupta [37]</td>
<td>@, Rot</td>
<td>4L</td>
<td>2L</td>
<td>7L</td>
</tr>
<tr>
<td>SASI-PR [38]</td>
<td>@, Rot, Con</td>
<td>4L</td>
<td>1.5L</td>
<td>7L</td>
</tr>
<tr>
<td>LMAP [41]</td>
<td>@, OR, +</td>
<td>4L</td>
<td>2L</td>
<td>6L</td>
</tr>
<tr>
<td>M2AP [42]</td>
<td>@, AND, O.R., +</td>
<td>4L</td>
<td>3L</td>
<td>6L</td>
</tr>
<tr>
<td>EMAP [43]</td>
<td>@, AND, O.R.</td>
<td>4L</td>
<td>3L</td>
<td>6L</td>
</tr>
<tr>
<td>LPCP [44]</td>
<td>@, Rot, CRC-16</td>
<td>5L</td>
<td>2L</td>
<td>5L</td>
</tr>
<tr>
<td>SASI [55]</td>
<td>@, OR, Rot, +</td>
<td>4L</td>
<td>2L</td>
<td>7L</td>
</tr>
<tr>
<td>David-Prasad [54]</td>
<td>@, AND</td>
<td>5L</td>
<td>3L</td>
<td>5L</td>
</tr>
<tr>
<td>Gossamer [55]</td>
<td>@, Rot, MisBits, +</td>
<td>4L</td>
<td>3L</td>
<td>7L</td>
</tr>
<tr>
<td>RAP [56]</td>
<td>@, Rot, Rec</td>
<td>5L</td>
<td>2L</td>
<td>5L</td>
</tr>
<tr>
<td>RCIA [57]</td>
<td>@, Per, R</td>
<td>4L</td>
<td>2L</td>
<td>7L</td>
</tr>
<tr>
<td>URMAP [58]</td>
<td>@, Per = XOR</td>
<td>4L</td>
<td>2L</td>
<td>5L</td>
</tr>
<tr>
<td>Ours</td>
<td>@, Rot, Rank</td>
<td>5L</td>
<td>1.5L</td>
<td>5L</td>
</tr>
</tbody>
</table>

Acronyms: L: Number of bits stored in each parameter.

- **Mutual authentication**: It implies that genuine tags and readers need to authenticate each other. In the ESRAS scheme, the reader-server unit authenticates the genuine tag by comparing the left or right half of the transmitted messages that are $E'_L \oplus R = R'_L \oplus L$. Likewise, the genuine tag authenticates the server by comparing the left or right half of the transmitted message $C'_L \oplus R = C'_L \oplus L$. Therefore, the property of mutual authentication is established between tag and server.

- **Resistance to tag tracking**: The tag’s ID or its secrets are not revealed in the ESRAS scheme. Moreover, it uses the index pseudonym $IDS$. The $IDS$ and the shared secret keys $K_1$ and $K_2$ are updated during each successful protocol run. Besides, the update operations also involve random numbers, so the tag will be anonymous to the adversary. This way, the adversary cannot track the response messages $A, B, C, D, E$ involving random numbers. In addition, we are not using any unbalanced operations during the update process, so the adversary cannot track the location of the tags through $IDS$.

- **Forward security**: It ensures securing past communications from a tag if the tag gets compromised by an adversary. Considering an adversary can access the tag, it cannot determine the previous secrets such as random numbers, index pseudonyms, and keys from the tag. Now, considering an adversary can determine the $IDS$ and keys of the tag. However, the adversary still cannot determine the previous secrets as different values of the $IDS$ and keys are used after a successful protocol run. Hence, the adversary cannot compromise the previously communicated information from the same tag.

- **Resistance to de-synchronization attacks**: The shared secret cannot be de-synchronized among tag and reader through an adversary. The tag and the readers will use different random numbers to update the shared secret in each authentication phase. Therefore, the adversary cannot modify or even change the response messages to change the values of $R_1$ and $R_2$. Hence, the proposed scheme successfully prevents the de-synchronization attack.

- **Resistance to disclosure attacks**: The adversary can slightly modify the messages from the reader and send them to the tag to verify the correctness of modifications. It is quite difficult to disclose the secrets (i.e., $K_1$ and $K_2$) even if the adversary knows $Rank(Rot(K_1, K_2), K_1)$. In our proposed scheme, the tag performs
masking (XORing operation) between the tag’s unique identification number $ID$ and $Rank(Rank(K_1 \oplus K_2, R_1), Rank(K_1, K_2))$. Therefore, it is impossible to disclose without knowing secrets. Hence, ESRAS resists the disclosure attack.

5.2. Performance evaluation

The performance evaluation of ESRAS is done in terms of computation, communication, and storage cost for each tag. The tags have limited computation capability and memory in comparison to the reader and server in RFID systems. However, the tags’ performance can be evaluated with no limitations in the hardware environments of the backend database and reader. Therefore, any authentication scheme considers the computational complexity of an RFID system. Table 6 illustrates the performance comparison of ESRAS with several considerable Ultra-lightweight RFID authentication schemes.

- **Computation cost:** It consists of the types of Ultra-lightweight primitives that are required for each tag. The proposed scheme does not utilize CRC, hash function, pseudo-Kasami code $K_c(X)$, and encryption $E(\cdot)/$decryption $D(\cdot)$ algorithm. In ESRAS scheme, only simple bitwise exclusive-OR ($\oplus$), and circular left rotation $Rot(X,Y)$, and a newly proposed Ultra-lightweight rank $Rank(X,Y)$ operations have been utilized.

- **Communication cost:** The numbers of transmitted messages by the tag in each authentication session. The tag transmits a total of one and a half communication messages. All the used parameters have $L$ bits. Thus, the communication cost generated on the tag is $1.5L$ bits.

- **Storage cost:** The shared elements and the static tag $ID$ are stored in the tag’s memory space. In the ESRAS scheme, a total of five strings, including its unique $ID$, the two entries for index pseudonym $IDS$, and two shared secret keys (i.e., $K_1$ and $K_2$) are stored on the tag. Hence, the storage of each tag is $5L$ bits.

6. Scyther simulation of our proposed scheme

Scyther is a GUI-based automatic tool to verify security protocols [59]. The Scyther tool checks whether the proposed protocol is secure under security attacks. The experimental setup is simulated by using Scyther installed on the Linux platform with Ubuntu v20.04. The command “scyther-gui.py” is used to open the window for opening and editing files. The Scyther input language is Security Protocol Description Language (SPDL), used for writing the protocol specification or description. The SPDL is case sensitive, where xyz.spdl is not the same as XYZ.spdl. Programs or SPDL descriptions of protocols are saved with a .spdl extension. A pre-defined set of claim events are in Scyther, such as Secret, session – keyreveal, Niagree, Nisynch, Alive, and Weakagree. Moreover, a sequence of events is used, which includes sending or receiving. Figs. 4 and 5 show the SPDL specification of the roles of tag and reader. From Fig. 6, the result status OK indicates that there are no possible attacks within bounds which means that the proposed ESRAS scheme strongly resists all possible active and passive (i.e., replay and Man-In-The-Middle) attacks.

7. Conclusion

Since security and privacy are two key concerns in RFID systems, several issues such as eavesdropping, tampering, and modifications may occur over a communication channel while transmitting messages. Considering these security flaws, we proposed an efficient and secure Ultra-lightweight RFID authentication scheme (ESRAS). We have utilized the simple bitwise exclusive-OR, circular left rotate, and newly proposed Ultra-lightweight rank operations to provide higher security...
with fewer computation and communication costs of tags. The rank operation has four major security properties: full confusion, irreversibility, sensibility, and low complexity. The security analysis demonstrates that our ESRAS scheme is resistant to possible security attacks. The performance evaluation demonstrates that ESRAS requires less storage cost and computation overhead than existing schemes. The Scyther results show that our scheme has no possible attacks within bounds. Furthermore, our scheme shows superiority for low-cost RFID systems and can be realized in several real-world domains.

**CRediT authorship contribution statement**

**Mohd Shariq**: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

**Karan Singh**: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

**Chhagan Lal**: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

**Mauro Conti**: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

**Tayyab Khan**: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

**Declaration of competing interest**

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Chhagan Lal reports was provided by Delft University of Technology.

**Data availability**

No data was used for the research described in the article.

**Acknowledgments**

All authors approved the version of the manuscript to be published.

**References**


Masoumeh Saffkhani, Nasour Bagheri, Generalized desynchronization attack on UMAP: Application to RCIA, KMAP, SLAP and SASI+ protocols, IACR Cryptol. ePrint Arch. 2016 (2016) 905.


MOHD SHARIQ received his B.Tech. degree in Computer Science and Engineering from Gautam Buddha Technical University, India, and M.Tech. degree in Computer Science and Technology from the School of Computer and Systems Sciences, JNU, New Delhi, India in 2017. He has submitted his Ph.D. thesis in the School of Computer and Systems Sciences, JNU, New Delhi. His primary research interests include RFID security protocols design, security and privacy, lightweight RFID authentication, and information security. He has published several research papers in reputed journals.

KARAN SINGH received the Engineering degree (Computer Science & Engineering) from Kamala Nehru Institute of Technology, Sultanpur, UP, India. He is the M.Tech. (Computer Science & Engineering) and Ph.D. (Computer Science & Engineering) from Motilal Nehru National Institute of Technology, UP, India. He worked at Gautam Buddha University, UP, India. Currently, he is working with the School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi. His primary research interests are in Computer network, Network security, Multicast communication, IoT, and Body Area network. He is the supervisor of many research scholars. He is a reviewer of Springer, Taylor & Francis, Elsevier Journals and IEEE Transactions. He is an Editorial Board Member of Journal of Communications and Network (CN), USA. He published 70+ research papers in refereed journals and good conferences. He organized the workshops, conference sessions, and training. Dr. Singh worked as General Chair of the international conference (Qbine 2013) at Gautam Buddha University, India. Recently he organized a conference ICCGT 2018 at Dronacharya College of Engineering, Gurgoan and a special session in 2nd ICGEIT 2018 at Denmark.

CHHAGAN LAL received the Ph.D. degree in computer science and engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. He is currently working as a Postdoctoral Research Fellow of the Delft University of Technology, The Netherlands. Previously, he was a Postdoctoral Fellow of the Department of Mathematics, University of Padua, Italy, where he was part of the SPURITZ Research Group. He was a Postdoctoral Research Fellow of Simula Research Laboratory, Norway. His current research areas include applications of blockchain technologies, security in software-defined networking, and Internet-of-Things networks. During his Ph.D., he was awarded the Canadian Commonwealth Scholarship under the Canadian Commonwealth Scholarship Program to work in the University of Saskatchewan, Saskatoon, SK, Canada.

MAURO CONTI received the Ph.D. degree from the Sapienza University of Rome, Italy, in 2009. He is a Full Professor with the University of Padua, Italy. He is also affiliated with TU Delft and the University of Washington, Seattle. After his Ph.D., he was a Postdoctoral Researcher with Vrije Universiteit Amsterdam, The Netherlands. In 2011, he joined as an Assistant Professor with the University of Padua, where he became an Associate Professor in 2015, and a Full Professor in 2018. He has been Visiting Researcher with GMU, UCL, UCL, TU Darmstadt, UF, and FIU. His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of security and privacy. In this area, he published more than 400 papers in topmost international peer-reviewed journals and conferences. He has been awarded the Marie Curie Fellowship by the European Commission in 2012, and a Fellowship by the German DAAD in 2013. He is an Area Editor-in-Chief for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. He was a Principal Investigator for the National Science Foundation Grant 1420696, and for the Special Session on Blockchain Applications and Security at the 2020 IEEE Global Communications Conference. He co-organized a conference ICCCS 2018 at Dronacharya College of Engineering, Gurgoan, and a special session in 2nd ICGEIT 2018 at Denmark.

TAYYAB KHAN received the Engineering degree in Computer Science & Engineering from Gautam Buddha Technical University, UP, India. He has completed his M.Tech. and Ph.D in Computer Science & Engineering from the school of computer and systems sciences, JNU New Delhi, India in OCT 2016 from the school of computer and systems sciences, JNU New Delhi, India. He has published various research papers in reputed journals. His primary research interests are in the Trust modeling, wireless sensor network, Body sensor networks, Network security, Multicast communication. He has published many research papers in refereed journals and good conferences. He organized the workshops, conference sessions, and training.

11