Multifactor Authentication and Key Management Protocol for WSN-assisted IoT Communication

Tabassum Ara and M. Prabhakar
School of Computing and Information Technology, Reva University, Bangalore, India

Abstract—In this paper a novel multi-factor authentication protocol for IoT applications, relying on enhanced Rabin-assisted elliptic curve cryptography, biometric features and time stamping methods, is developed. Furthermore, a fuzzy verification algorithm has been developed to perform receiver-level user verification, making computation efficient in terms of computational overhead as well as latency. An NS2 simulation-based performance assessment has revealed that the multifactor authentication and key management models we have proposed are capable of not only avoiding security breaches, such as smart card loss (SCLA) and impersonation attacks, but can also ensure the provision of maximum possible QoS levels by offering higher packet delivery and minimum latency rates.

Keywords—multifactor authentication, IoT security, ECC, timestamp, one-way bio-hashing, fuzzy verifier, WSN.

1. Introduction

Advancement of new technologies has given rise, over the past few years, to a new paradigm known as the Internet of Things (IoT), relying on machine-to-machine (M2M) communications performed on a large scale. Functionally, these communication paradigms exploit Wireless Sensor Networks (WSN) with augmented routing protocols to simultaneously transmit data between multiple users or machines. Typically, IoT systems require secure data transmission or communication protocols across peers and, hence, a proper security system becomes an inevitable need [1], [2].

In IoT communication, the Internet Engineering Task Force (IETF) recommended certain protocols and standards to incorporate WSN into the Internet [3], such as IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [4] and Routing Over Low Power and Lossy Networks (ROLL) [5]. Functionally, IoT may comprise multiple sensor nodes connected to the Internet via a gateway. In such a scenario, the connected sensors would be accessible by any authorized entity, thereby ensuring remote access to and control over major applications. As a consequence, IoT communication remains vulnerable to attacks [2], [6]–[11]. Considering the criticality of data security across IoT ecosystems, the transmitted data must be protected and secured across the sensor nodes and the entities connected to the network, by means of certain secure peer-to-peer channels. It is not feasible to apply Internet security measures directly to the IoT, due to WSN characteristics [12]. Nevertheless, efforts have been undertaken, such as IPsec [13] and IKEv2 [14], to ensure the security of IoT using Internet security models, with the resource-constrained nature of WSNs limiting their efficiency.

The classic uni-factor security models, such as password-based or bio-physical security systems are prone to being attacked, as the number of hacking techniques that exists is growing exponentially. Therefore, it becomes necessary to enable multi-factor assisted user (node) authentication to establish keys between nodes and the different authenticated stakeholders or entities. Furthermore, the existing efforts [15]–[17] have affirmatively stated that designing a secure IoT communication protocol is feasible. The objective may be achieved by enabling channel authentication and key management strategies requiring the remote entities to mutually authorize each other and to negotiate secret keys to assist the sensor nodes in avoiding active and passive attacks during the transmission [12]–[17]. Noticeably, even with certain security features deployed at the link layer of the IEEE 802.15.4 protocol stack, the openness of the Internet still causes significant vulnerability and, hence, demands certain robust key agreement and authentication schemes [12].

In this paper, a robust multi-factor authentication protocol has been developed that exploits efficacy of time-stamping, Rabin cryptosystem-assisted elliptic curve cryptography (ECC) and bio-hashing. To enable optimal user verification under real-time communication scenarios, a fuzzy-based verification model has been applied that learns the above mentioned features to provide access to a node for further communication. The proposed model has been implemented over a WSN with multiple cooperatively communicating or connecting nodes. NS-2 simulation has revealed that the proposed security model exhibits goods efficiency without incorporating any significant computational overheads or complexities.

The remaining sections of this paper are organized as follows. Section 2 presents the related work, which is fol-
ollowed by the discussion of key existing approaches to IoT data security, given in Section 3. Section 4 shows the proposed system and its implementation. Section 5 presents the results and discussion, while conclusions concerning the overall results are given in Section 6.

2. Related Work

Realizing the shortcomings of single parameter-based security systems, Shin et al. [18] developed a two-factor authentication model where the prime focus was on employing an authenticated key agreement paradigm between users and IoT devices. However, this model was found to be limited due to its inability to address stolen or lost smart card attacks (SCLA), as well as offline password guessing and/or retrieval attacks relying on Brute Force, etc. To deal with these shortcomings, Wazid et al. [19] designed a new secure lightweight multi-factor remote user authentication scheme for hierarchical IoT networks (HIoTN). They proposed a user-authenticated key management protocol (UAKMP). However, these approaches are complex. To reduce computational complexities, Saravbhatla et al. [20] used a biometric feature authentication model for heterogeneous WSNs. However, personalized biometric traits preserved in a memory chip can be compromised due to smart card loss.

Challa et al. [21] developed a secure signature-based authenticated key establishment scheme for future IoT applications. Porambage et al. [22] developed a group key establishment protocol for secure multicast communications between resource-constrained sensor devices in IoT. However, the ambiguity of information shared between the interconnected nodes confines its efficiency in large networks. Ning et al. [23] developed a proof-based hierarchical authentication scheme for IoT. This model focused primarily on developing a U2IoT architecture (i.e. unit IoT and ubiquitous IoT), and eventually recommended an aggregated proof-based hierarchical authentication scheme (APHA). The authors combined directed path descriptors, homomorphism functions and Chebyshev chaotic maps for mutual authentication to ensure hierarchical access control.

Mick et al. [24] developed a lightweight authentication and secured routing (LASeR) method for named data networks (NDN) used in smart city IoT applications. To enable computationally efficient routing, He et al. [25] developed an ECC-based RFID authentication scheme for the healthcare sector. Being an energy-constrained network, WSN-assisted IoT requires energy efficient and reliable transmission. Hence, Mohd et al. [26] developed lightweight block ciphers enhancing IoT security. The authors focused on both security and computational efficiency. Similar work was performed by Lu et al. [27], who developed a lightweight privacy-preserving data aggregation scheme, known as lightweight privacy-preserving data aggregation (LDPA). They relied on homomorphic Paillier encryption, Chinese remainder theorem and one-way hash chain techniques to ensure efficient data gathering and to achieve a reduction in the false rate.

Jakalan et al. [28] designed a model called network security situation awareness (NSSA), where the focus was on assessing security situation-related elements and information originating from a multi-source heterogeneous networks. The authors considered four IoT security-related variables, such as context, attack, vulnerability, and network flow, which were then processed using the ontological concept to obtain the best security solution. In order to augment computational complexity, Diro et al. [29] recommended a lightweight cryptographic model – ECC. A similar effort was made by Yuan et al. [30], who developed a lightweight trust mechanism for IoT edge devices based on the fusion of multi-source feedback information. The use of multi-source feedback information for global trust estimation enabled avoiding a scenario in which a malicious node becomes a part of the network and, hence, enhance the security of the solution. This approach may be helpful in exploiting different node features to isolate the unauthorized node. However, this is done at the cost of increased computational overheads and latency.

Zahra et al. [31] employed Shibboleth, also known as the security and cross-domain access control protocol between fog a client and a fog node, to achieve secure communication between nodes, even under uncertain network conditions. Zheng et al. [32] utilized attribute-based encryption to enable data sharing over the network. In addition to the removal of attribute matching function, the use of the attribute bloom filter enabled hiding all attributes in the access control structure. However, the efficacy of their model could not be assessed in terms of node performance parameters. A similar effort was made by Want et al. [33]. Lei et al. [34] derived a closed-form model for assessing the probability of security-outage and its impact on throughput. They assessed their model to achieve better trade-offs between secure communication and energy-efficient data transmission over IoT. Lai et al. [35] derived a novel pairing-free data access control scheme by exploiting cipher text policy, attribute-based encryption (CP-ABE), where ECC was applied as encryption the method. Elhoseny et al. [36] designed a hybrid-cryptosystem using AES and RSA algorithms for diagnostic text data security in medical images. The authors applied 2-dimensional discrete wavelet transform and steganography concepts to ensure secure data transmission. In [37] Ruan et al. designed a leakage-resilient (LR) eCK security model for the password-based authenticated key exchange (PAKE) protocol. The authors applied the LR PAKE protocol with Diffie-Hellman key exchange LR storage (LRS).

3. Proposed Method

Unlike major existing methods, the proposed solution exploits multiple authorization schemes to design a robust security model, where each factor may function as a supplementary security layer, to ensure seamless communi-
3.1. Bio-Hashing

Over the past few years, the use of bio-traits as a security feature has increased significantly. Considering the increased threat of security breaches in the classic approaches, biometric features, such as fingerprint, retina and face detection, have been applied as additional security layers. The use of biometric traits as a supplementary security feature in conjunction with classic passwords or smart cards, has exhibited satisfactory results, making it one of the most sought-after techniques. Biometric features are deeply integrated and closely coupled with each individual. On the other hand, the irreplaceable nature of biometric features makes them most resistant to security breaches or to unauthenticated attempts to access specific systems or resources. Over the past few years, numerous research projects have been completed, relying on the efficiency of biometric data as security features [38], [39]. Biometric templates may be considered one of the pre-dominant privacy-preserving biometric approaches [39]. In practice, biometric template \( B \) and certain random constructs called secret key \( K \) are employed to generate a bio hash value (BV) (BV) Bio2H(\( K,B \)) during registration. Noticeably, here Bio2H(\( K,B \)) defines a bilinear hashing of the biometric feature, such as a fingerprint, retina information, etc. Bio2H(\( K,B \)) (two-factor bio hashing) pre-processes the biometric feature template \( B \) to preserve its integrity and accuracy. In other words, the pre-processing of \( B \) leaves the biometric feature concerned intact, even if certain minor variations in the input biometric signal are experienced. Once the BV has been generated, Bio2H(\( K,B \)) compares the inner product of the random vector generated from the user specific secret key \( K \) and the feature vector extracted, against a predefined threshold value. During the verification phase, the same process as applied at the registration stage is followed. In this phase, the received biometric template or signal \( B' \) and the secret key specified by the user as a BV value Bio2H(\( K,B' \)) are estimated. Once the initial BV Bio2H(\( K,B \)) and the re-estimated value of BV Bio2H(\( K,B' \)) have been obtained at the user end, they are compared to authenticate the user [39]. The key assumption for use of any bio-hashing technique is that the user has to be registered/on-boarded with at least one biometric feature captured (biometric template \( B \)), which is then converted into and stored in a suitable binary format (bits).

In the proposed research, we have considered fingerprint bits BioU\(_i\) to be biometric template \( B \) which, when combined with secret \( K \) at the registration end, is compared with received template \( B' \) combined with \( K \) as a signature to validate the user.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Key exchange</th>
<th>Encryption/decryption</th>
<th>Digital signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffie Hellmen</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSA</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RSA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ECC</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bio2H</td>
<td>Two factor bio hashing</td>
</tr>
<tr>
<td>( K )</td>
<td>Secret key</td>
</tr>
<tr>
<td>( B )</td>
<td>Biometric feature template</td>
</tr>
<tr>
<td>( B' )</td>
<td>Received biometric feature template</td>
</tr>
<tr>
<td>( P, Q )</td>
<td>Points of elliptic curve</td>
</tr>
<tr>
<td>( x, y )</td>
<td>Coordinates on elliptic curve</td>
</tr>
<tr>
<td>( M )</td>
<td>Plain message</td>
</tr>
<tr>
<td>( C )</td>
<td>Cipher text</td>
</tr>
<tr>
<td>( P )</td>
<td>Prime number</td>
</tr>
<tr>
<td>( GF )</td>
<td>Galois field</td>
</tr>
<tr>
<td>( a, b )</td>
<td>Coefficients</td>
</tr>
<tr>
<td>( G )</td>
<td>Generator point</td>
</tr>
<tr>
<td>( n, h )</td>
<td>Cryptographic prime factor and cofactor</td>
</tr>
<tr>
<td>( r )</td>
<td>Random integer</td>
</tr>
<tr>
<td>( Pu )</td>
<td>Public key</td>
</tr>
<tr>
<td>( UID_i )</td>
<td>( i )-th user identification</td>
</tr>
<tr>
<td>( PWU_i )</td>
<td>( i )-th user password</td>
</tr>
<tr>
<td>( l )</td>
<td>Integer</td>
</tr>
<tr>
<td>( U_i )</td>
<td>( i )-th user</td>
</tr>
<tr>
<td>( AD_j )</td>
<td>Admin/sensor node</td>
</tr>
<tr>
<td>( ADID_j )</td>
<td>Admin’s ID</td>
</tr>
<tr>
<td>( ADxi )</td>
<td>Admin’s ( x ) value</td>
</tr>
<tr>
<td>( h() )</td>
<td>Hash function</td>
</tr>
<tr>
<td>( R_{shrd} )</td>
<td>Shared random number</td>
</tr>
<tr>
<td>( SCN_j )</td>
<td>Smart card number</td>
</tr>
<tr>
<td>( SCS )</td>
<td>Smart card storing</td>
</tr>
<tr>
<td>( BioU )</td>
<td>Bio information of user ( U_i )</td>
</tr>
<tr>
<td>( SC )</td>
<td>Smart card</td>
</tr>
<tr>
<td>( Z_i )</td>
<td>Arbitrary number</td>
</tr>
<tr>
<td>( EID )</td>
<td>Estimated ID</td>
</tr>
<tr>
<td>( GN )</td>
<td>Gateway node</td>
</tr>
<tr>
<td>( ID_{GN} )</td>
<td>ID of gateway node</td>
</tr>
<tr>
<td>( M_1, M_2, M_3, M_4 )</td>
<td>Intermediate values of messages</td>
</tr>
<tr>
<td>( T )</td>
<td>Delay threshold</td>
</tr>
<tr>
<td>( MSG )</td>
<td>Message</td>
</tr>
<tr>
<td>( SK_{GN} )</td>
<td>Session key of gateway node</td>
</tr>
<tr>
<td>( DD_1, ZZ_4 )</td>
<td>Derived parameter</td>
</tr>
<tr>
<td>( q^{pw}, f^{pw}, g^{pw} )</td>
<td>New estimated hash values</td>
</tr>
</tbody>
</table>
3.2 Rabin-Assisted Curve Cryptography

There are numerous public key cryptosystems, such as RSA, DSA or Diffie-Hellmen. The majority of these classic approaches suffer from huge computational overheads and time complexities (Table 1). Table 2 summarizes the notation used. Robustness of the ECC cryptosystem may be visualized based on the intricacy of the elliptic curve discrete log-arithmetic problem (ECDLP). Let the expression be \( Q = kP \), where \( P \) and \( Q \) pertain to \( \text{Fp}(a,b) \). With \( k \) and \( P \) values provided, it becomes easy to calculate \( Q \). On the contrary, with \( P \) and \( Q \) pre-specified, it becomes more complicated to estimate \( k \), especially when \( k \) is large. Noticeably, \( k \) states the discrete logarithm of \( Q \) to the base \( P \), signifying the discrete logarithm problem for ECC. This computational complexity enhances attack resilience of ECC.

The predominant process involved in ECC is known as point multiplication. Functionally, elliptic curve \( E \) is defined as:

\[
y^2 = x^3 + ax + b
\]  

(1)

In Eq. (1), the highest degree is 3. In the proposed method, ECC encrypts data \( M \) and generates ciphertext \( C \) and vice versa, using a certain finite set of points in the elliptic curve over \( GF(p) \). Equation (1) (Weierstrass equation), \( y^2 = x^3 + ax + b \) is used in conjunction with modulo \( p \) to generate points on the elliptic curve. Typically, the elliptic curve-specific variables are \( a, b, G, n, h, r \), where \( p \) states a prime number, \( a \) and \( b \) are the coefficients, and parameter \( G \) presents a generator point. The other parameters, such as \( n \) and \( h \), define the cryptographic prime factor and the co-factor, respectively. Here, \( r \) is a random integer which is lower than \( n \). The proposed model employs the finite field elliptic curve with modulo 263. A snippet of the applied method is given as Algorithm 1.

Algorithm 1: The proposed model concept

```{}
Chose an elliptic curve having modulo \( p(y^2 = x^3 + ax + b \mod p) \)
Assign the values of \( a \) and \( b \), and of the coefficients.
Estimate the value of \( y^2 = x^3 + ax + b \mod p \)

For \( x = 0 \) to \( (p - 1) \)
\[ S = x^3 + ax + b \mod p \]
For \( d = 0 \) to \( (p - 1)/2 \)
\[ T = d^2 \mod p \]
If \( T = S \)
\[ Y1 = d \] and \( Y2 = p - d \)
Else \( d = d + 1 \)
\[ x = x + 1 \]
\( (x,Y1), (x,Y1) \)
```

4. System Model

Unlike in the classic RSA, in the ECC-based cryptosystem presented in this paper we have designed a multifactor authentication and key management strategy. The proposed system incorporates the ECC cryptosystem, one-directional bio-hashing, time stamping and fuzzy verification. Such an approach results in a robust security solution enabling to avoid attacks, like SCLA and location tracking. The use of the ECC model is a lightweight crypto-solution. The use of the timestamp method enables mitigating session-specific temporary information attacks, eventually avoiding any tracking attacks. On the other hand, the use of the fuzzy verifier enables the model we have proposed to perform local verification of the users’ passwords, thus offering a robust scheme for malicious node isolation. A snippet of the proposed multi-factor authentication and key management model is given below.

4.1 Model Description

As discussed above, the proposed system is initiated by relying on the Rabin-based ECC cryptosystem.

In this preliminary phase, a sensor administration node ADM generates two prime numbers \( p \) and \( q \), and estimates variable \( N = pq \). Preserving \( (p, q) \) as the private key, it selects a master reference key \( x_{GN} \) in addition to an integer \( l (2^4 \leq l \leq 2^5) \) which is used at the receiver for local password verification, performed with the use of the fuzzy verifier. Estimating the receiver’s master reference
key, hereinafter referred to as the public key, the cipher-text point is obtained as \( C = \left(\{r.G\}, \{M + r.Phr\}\right) \). The first point of the cipher-text pair \( r.G \) is multiplied by private key \( Prh \), and the result obtained is subtracted from the second point of the cipher-text pair.

Next, ADM selects an identity \( AD_{ID_j} \) and estimates the secret key:

\[
AD_{X_j} = h(AD_{ID_j}||x_{GN}) \text{ for } AD_j (1 \leq j \leq M).
\]

Then, ADM generates a random number \( R_{shrd} \), which is shared between the GN and the node \( AD_j \). Node \( AD_j \) stores \( AD_{ID_j} \) and \( R_{shrd} \) in its memory.

### 4.1.1. User Registration and Login

During the user registration phase, user \( U_i \) performs the following steps to get registered with ADM.

User \( U_i \) transmits the selected identity \( U_{ID_i} \) along with their personal credentials to ADM, by means of a specific secure channel.

Upon receiving information from \( U_i \), ADM verifies whether \( U_{ID_i} \) exists in the table or in the database. If the verification is positive, ADM acknowledges \( U_i \) to select an updated or a new identity: else it generates an arbitrary number \( x_i \), and then estimates \( d_i = h(U_{ID_i} || x_{GN}) \) and \( L_i = h(SC_{Ni} || x_{GN}) \). Here, \( SC_{Ni} \) defines the SC number belonging to a user or a node. Estimating the values of \( d_i \) and \( L_i \), ADM enables SCS.

\( U_i \) connects the smart card into a card reader and then enters \( U_{ID_i}, PW_{U_i} \); and imports \( Bio_{U_i} \); SC selects a random number \( r_i \) and estimates the attributes:

\[
\begin{align*}
BioZ_i &= Bio2H(r_i, Bio_{U_i}), \\
e_i &= h(U_{ID_i} || PW_{U_i} || BioZ_i) \mod l, \\
f_i &= d_i \oplus h(U_{ID_i} || PW_{U_i} || BioZ_i), \text{ and} \\
g_i &= L_i \oplus h(U_{ID_i} || PW_{U_i} || BioZ_i).
\end{align*}
\]

Upon performing user registration, the user requires a login to the system. This step occurs when user \( U_i \) intends to access sensor data.

First, \( U_i \) connects SC and feeds its unique identity \( U_{ID_i} \) along with password-related information \( PW_{U_i}^* \). In addition, it requires that fingerprint information \( Bio_{U_i} \), is embedding or attached. Now, the deployed SC estimates \( BioZ_i^* = Bio2H(r_i^*, Bio_{U_i}) \) in addition to \( e_i^* = h(U_{ID_i} || PW_{U_i}^* || BioZ_i) \mod l \). This process ensures that in the case of \( e_i^* \neq e_i \), the card will reject \( U_i \), without granting access to sensor data.

Next, the SC generates an arbitrary number \( z_i \) along with a timestamp \( T_{stamp1} \), and calculates:

\[
\begin{align*}
d_i^* &= f_i \oplus h(U_{ID_i} || PW_{U_i} || BioZ_i), \\
L_i^* &= g_i \oplus h(U_{ID_i} || PW_{U_i} || BioZ_i^*), \\
M_1 &= (U_{ID_i} || SC_{Ni} || z_i)^2 \mod n, \\
M_2 &= h(d_i^* || L_i^* || z_i || T_{stamp1}).
\end{align*}
\]

Finally, \( U_i \) selects the ID of the \( j \)-th sensor \( AD_{ID_j} \) which it intends to access, and the SC estimates value \( EID_j = AD_{ID_j} \odot h(U_{ID_i} || z_i || T_{stamp1}) \), which is then appended and transmitted to the GN as \( MSG_1 = <M_1, M_2, T_1, EID_j> \).

### 4.1.2. Authentication

To perform a secure or valid mutual authentication, and to agree on the session key, the proposed model performs the following processes.

Upon receiving \( MSG_1 \) from \( U_i \), GN at first decrypts \( M_1 \) using \( p \) and \( q \) to obtain \( U_{ID_j}, SC_{Ni}^* \), and \( z_i \). \( x_i \) is retrieved as per \( U_{ID_j} \), which is then validated in the entries with \( SC_{Ni}^* \). If these values fail to match, GN rejects the request and terminates the process. Otherwise, it estimates:

\[
\begin{align*}
L_i' &= h(SC_{Ni}^* || x_{GN}), \\
d_i' &= h(U_{ID_i} || x_{GN} || x_i), \\
z_i' &= M_2 \oplus h(d_i' || T_{stamp1}), \\
M_3 &= h(d_i' || z_i' || T_{stamp1}).
\end{align*}
\]

In the case of \( M_3' \neq M_3 \), GN aborts the current session. On the contrary, for \( M_3' = M_3 \) it estimates:

\[
\begin{align*}
AD_{ID_j} &= EID_j \odot h(U_{ID_i} || z_i || T_{stamp1}), \\
AD_{X_j}' &= h(AD_{ID_j} || x_{GN}), \\
M_3 &= h(U_{ID_i} || AD_{ID_j} || AD_{X_j} || z_i' || T_{stamp2}), \\
M_4 &= U_{ID_i} \oplus h(DGN || AD_{X_j} || T_{stamp2}).
\end{align*}
\]

Next, GN transmits message \( MSG_2 = <ID_{GN}, M_3, M_4, T_{stamp1} > \) to \( AD_j \).

If the next step, \( AD_j \) verifies whether \( |T_{stamp2} - T_{stamp1}| \leq \Delta T \) exists, where \( T_{stamp2} \) signifies the current timestamp. If the verification result is positive, \( AD_j \) stops the session, otherwise it estimates:

\[
\begin{align*}
U_{ID_i}^* &= M_4 \oplus h(ID_{GN} || AD_{X_j} || T_{stamp2}), \\
z_i^* &= M_5 \oplus h(U_{ID_i}^* || AD_{ID_j} || AD_{X_j} || T_{stamp2}), \\
M_7 &= h(U_{ID_i}^* || AD_{ID_j} || AD_{X_j} || z_i^* || T_{stamp2}).
\end{align*}
\]

Now, \( AD_j \) aborts the connection in the case of \( M_7' \neq M_7 \). Otherwise, it confirms the authenticity of \( U_i \) and GN. Next, \( AD_j \) estimates the following:

\[
\begin{align*}
S_{Kj} &= h(U_{ID_i}^* || AD_{ID_j} || z_i^* || z_j), \\
M_6 &= h(SK_{GN} || AD_{X_j} || T_{stamp3}), \\
M_7 &= z_i^* \oplus z_j, \text{ where } z_j \text{ is a random number generated by } AD_j.
\end{align*}
\]

In the next phase, \( AD_j \) forwards \( MSG_3 = <M_6, M_7, T_3 > \) to GN.

In the following phase, GN verifies whether \( |T_{stamp4} - T_{stamp3}| \leq \Delta T \). If not, GN terminates the session. Otherwise, it estimates:

\[
\begin{align*}
z_j' &= M_7 \oplus z_j', \\
S_{KGN} &= h(U_{ID_i}^* || AD_{ID_j} || z_i^* || z_j'), \\
M_8 &= h(SK_{GN} || AD_{X_j} || z_i^* || T_{stamp3}).
\end{align*}
\]

In the case of \( M_8' \neq M_8 \), GN terminates the connection and estimates \( M_8 = h(SK_{GN} || U_{ID_i}^* || d_i' || z_j') \). Thus, the gateway node GN transmits \( MSG_4 = <M_7, M_8 > \) to \( U_i \).
Once receiving $MSG_A$, $U_i$ estimates:

$$z_j^* = M_1 \oplus z_i,$$

$$SK_i = h(U_{ID_i}||AD_{ID_j}||z_i||z_j^*)$$, and

$$M_k^* = h(SK_j||U_{ID_j}||d_j||z_j^*).$$

If $M_k^* \neq M_k$, $U_i$ terminates the session. Otherwise, it assumes that gateway node GN and $AD_j$ are authentic. At this instant, a common session key $SK_i = SK_j = SK_{GN}$ is established between the participating $U_i$, GN and $AD_j$.

5. Results and Discussion

To examine the effectiveness of the proposed model, reference works [38], [39], where three factor-based authentication was proposed, have been taken into consideration. The overall results have been assessed in terms of qualitative and quantitative, or empirical outcomes. As far as the qualitative assessment is concerned, the proposed method has been examined for its ability to avoid different attack scenarios. On the other hand, in the quantitative assessment, the model has been integrated with a WSN simulator-2, version (NS-2). The outcomes have been examined in terms of latency, packet delivery and packet loss rates.

5.1. Resistance to SCLA Attacks

Consider that attacker $A$ obtains SC information containing $<e_i, f_i, g_i, SC_N, l, n, r_i, Bio2H(\ldots), Hash1()>$ of valid user $U_i$.

Attacker $A$ may be able to guess the user credentials of $U_{ID_i}$ as well as $PWU_i^*$, which may help in estimating $e_i^* = h(U_{ID_i}^*||PWU_i^*||BioZ_1)$. However, they cannot retrieve the correct value of $U_{ID_i}$ and $PWU_i^*$ due to $e_i$ being a "fuzzy verifier". This novelty enables the proposed system to avoid an SCLA attack. In addition, the proposed model is capable of withstanding an SCLA Type II attack as well.

Consider that the attacker node $A$ has identified the message $MSG_1 = <M_1, M_2, T_1, EID_j>$ transmitted by $U_i$ when logging in, where:

$$d_j^* = f_i \oplus h(U_{ID_i}^*||PWU_i^*||BioZ_1^*),$$

$$L_j^* = g_i \oplus h(U_{ID_i}^*||PWU_i^*||BioZ_1^*).$$

In the above expression, $g_i$ is obtained from SC of $U_i$. SC. The complexity of the quadratic residue problem makes it infeasible for the attacker to estimate $R_1$ from the value $M_1 = (U_{ID_i}||SC_N||z_i) \mod n$, hence it prevents $A$ from estimating value $M_2^* = h(d_j^*||L_j^*||z_i||T_1)$, that is required to verify the user with its $U_{ID_i}$ and $PWU_i^*$. This approach makes the proposed model resilient to SCLA Type II attacks.

5.1.1. Resistance KSSTIA and User Impersonation Attacks

In the existing methods, their authors applied a static value $h(U_{ID_i}^*||AD_{X_j}^*)$ to secure the ephemeral arbitrary numbers, where variable $AD_{X_j}^*$ signifies the node’s key shared between $AD_j$ and GW. Consequently, the revealing of ephemeral random number $z_i$ compromises the static value $h(U_{ID_i}^*||AD_{X_j}^*)$, which eventually leads to revealing or compromising the ephemeral arbitrary numbers in other key management or authentication sessions. To alleviate the risk, a multi-factor authentication and key management policy with timestamp and one-way hashing concepts is incorporated into the proposed model. Such an approach allows the proposed model to avoid KSSTIA type attacks.

In user impersonation attack scenarios, attacker $A$ is prevented from performing any user impersonation attacks. Consider that attacker $A$ manages to retrieve $U_i$’s SC card and extracts information like $<e_i, f_i, g_i, SC_N, l, n, r_i, Bio2H(\ldots), Hash1()>$. Furthermore, let’s assume that $A$ has already identified or obtained the messages communicated in the previous authentication sessions. In such cases, our proposed security model forces $A$ to have all authorizing factors including $PWU_i$, SC, and biometric information for generating a certain valid message $MSG_1 = <M_1, M_2, T_1, EID_j >$. Practically, the key required to facilitate the authenticity of the user $U_i$ refers to value $M_2 = h(d_j^*||L_j^*||z_i||T_1)$. The vital constructs of $M_2$ encompass $d_j^* = f_i \oplus h(U_{ID_i}^*||PWU_i^*||BioZ_1^*)$ and $L_j^* = g_i \oplus h(U_{ID_i}^*||PWU_i^*||BioZ_1^*)$. However, without providing the password for that user $PWU_i$, as well as SC and biometric information, the intruder or attacker $A$ cannot estimate $d_j^*$ or $L_j^*$.

5.1.2. Resistance to Gateway Impersonation Attacks

In the proposed system, attacker $A$ is capable of impersonating the user by pretending GW to be either user $U_i$ or $AD_j$. To impersonate GW as $AD_j$, the intruder $A$ must estimate $M_3 = h(U_{ID_i}||AD_{ID_j}||ID_{GW}||AD_{X_j}||z_i^*||T_2)$. However, without having precise information about $h(U_{ID_i}||x_{GW})$, it is impossible for the intruder to estimate the value of $M_3$. On the other hand, the use of the hash algorithm and the timestamping technique means that intruder $A$ will not be able to retrieve any significant information from messages obtained from the previous authentication sessions. On the contrary, impersonating as GN to user $U_i$, the intruder requires estimating a valid factor $M_8 = h(Z_{Session}(\_GN)||U_{ID_j}^*||d_j^*||z_j^*)$. To achieve it, the intruder requires having information about $z_i$ that further helps in the estimation of $Z_{Session}(\_GN) = h(U_{ID_j}||AD_{ID_j}||z_i||z_j)$. To retrieve the value of $z_i$, $A$ requires knowing the secret key of GN. This condition cannot be fulfilled, as the secret key is preserved or protected by the administrator. An in-depth analysis shows that $A$ can impersonate by decrypting $M_1 = (U_{ID_i}||SC_N||z_i) \mod n$. However, this is a highly tedious and challenging task due to the computational complexity of Rabin-assisted ECC and its allied quadratic residue problem (QRP). Therefore, the proposed security model or the allied protocol eliminates any possibility of a gateway node impersonation attack.
5.1.3. Resistance to Modification and Replay Attacks

In the proposed method, intruder A is unable to make any modification to messages $MSG_1 = < M_1, M_2, T_1, EID_1 >$, $MSG_2 = < ID_{GW}, M_1, M_2, T_2 >$, $MSG_3 = < ID_{GW}, M_7, T_3 >$, or $MSG_4 = < M_7, M_8 >$. Consider that attacker A is capable of intercepting any one of the message chunks. Then, it may be able to modify the same and transmit them further. However, in the proposed method, each message is protected by means of a hash value that is estimated using a secret value. Therefore, the attacker cannot retrieve the message. For illustration, in $MSG_1$, attacker A cannot estimate the value of $M_2 = h(d^*_i || L^*_i || z_i || T_1)$, because $d^*_i = f_i \otimes h(U_{ID_i} || PWU^*_i || BioZ^*_i)$ and $L^*_i = g_i \otimes h(U_{ID_i} \otimes PWU^*_i \otimes BioZ^*_i)$ are secret values which cannot be estimated without knowing $PWU_i$, SC or the biometric feature. In the case of any modification, the receiver can detect it when checking the correctness of the hash value in each message. Thus, our proposed system can be described as being modification resilient.

In IoT, mobile nodes may be used. In such a scenario, the attacker may try to replay a stale message transmitted by a certain user. In the proposed method, the timestamp feature allows to resist any replay attacks.

5.1.4. Resistance to Insider Attacks and Verification of Credentials

Typically, nodes or users may sign up to various applications or information systems using similar passwords. Should an insider somehow get access to the password, they may use it for impersonating the user and getting access to their data. User $U_i$ submits $U_{ID_i}$ when signing up or registering. Therefore, the insider will not be able to achieve the user’s password.

While attacking the authentication server, verification information, e.g. the password, may be retrieved or stolen. The proposed method enables the server to retain such attributes as $< U_{ID_i}; SC_i; N_i; x_i; $ personal credential $>$ and does not store any password-related information. Therefore, even after gaining access to the authentication server, the attacker cannot obtain the user’s password-related information.

5.1.5. Mutual Verification, Session Key Attack, and Anonymity

In WSN-based IoT, mutual authentication is required between the nodes. In practice, the intruder is not capable of easily retrieving $M_2 = h(d^*_i || L^*_i || z_i || T_1)$ without having the genuine private key of the user, $d^*_i$ and $L^*_i$. In such a case, the gateway node GN is not able to authenticate $U_i$ by verifying the precision of $M_2$. In the same way, user $U_i$ can verify gateway node GN by verifying the correctness of $M_6 = h(Z_{Session_i} || U_{ID_i} || d'_i || z'_j)$. In this scenario, user $U_i$ and the GN can authenticate each other. Furthermore, $AD_j$ authorizes GN by checking the correctness of $M_3 = h(U_{ID_i} || AD_{ID_j} || ID_{GW} || AD_{X_j} || z'_j || T_2)$. Similarly, gateway node can authorize $AD_j$ by verifying the correctness of $M_6 = h(Z_{Session_i} || AD_{X_j} || z'_j || T_3)$. The proposed security model offers seamless mutual authentication between GN and $AD_j$.

During mutual authentication, session key $Z_{Session} = h(U_{ID_i} || AD_{ID_j} || z_i || z_j)$ is formed between user $U_{ID_i}$ and $AD_j$ to secure future communication. Noticeably, the security of $Z_{Session}$ relies on the secrecy of the random numbers involved. In fact, these values remain protected by the

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User untraceability</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Replay attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>User impersonation attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Gateway node impersonation attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Sensor node impersonation attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>De-synchronization attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Support of dynamic node addition</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Insider attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Stolen smartcard attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>User anonymity</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Identity guessing attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Support of three-factor security</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Supports correct password update</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Session key disclosure attack</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Bio hashing</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Quality of service</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Rabin-assisted ECC</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Fuzzy verification</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
</tbody>
</table>
secret values shared between WSN nodes participating in the exchange of each message. Let the session key be $Z_{Session} = h(U_{IDi}||AD_{IDj}||z_i||z_j)$ which is somehow known to intruder A. Even though A knows the session key, they cannot estimate any future or past session keys using $Z_{Session}$ as the session key itself is secured with the help of one-way hash function $Hash()$. In addition, the random number $<z_i,z_j>$ may be different in each session, and, therefore, the proposed model offers resistance to any session key attacks.

A situation may be experienced when intruder A retrieves messages transmitted between the users and tries to identify the user. Now, observing our proposed model, it can be found that it integrates user identity in $M_1 = (U_{IDi}||SC_N_i||z_0)^2 \mod n$. To retrieve user identity $U_{IDi}$, the intruder requires the knowledge of the secret key when executing Rabin-assisted ECC of gateway GW. In practice, it is impossible to retrieve, as it is already stored by ad-ministrator. On the other hand, the computational-complexity of the quadratic residue problem (QRP) makes it near impossible for the intruder to obtain $U_{IDi}$ by decrypting the value of $M_1 = (U_{IDi}||SC_N_i||z_0)^2 \mod n$.

Table 3 compares the proposed solution with other similar methods described in the referenced publications.

### 5.2. Simulation Results

To assess the efficiency of the proposed (secure) WSN routing protocol, we simulated it with a distributed sensor network comprising 50 nodes cooperating across the region. Furthermore, each node was assigned a radio range of 200 m. To examine the effectiveness of the proposed secure routing protocol in avoiding malicious or attack nodes, two compromising or attacker nodes were incorporated in the simulation environment (Table 4).

Figure 1 presents the packet delivery ratio (PDR) performance of our proposed security model and a routing protocol without any security features. As depicted, performance of the proposed model was nearly equivalent to that of the model without any security. This signifies that the proposed approach is lightweight and may be well suited for any WSNs. Similar performance-related results are visualized in Fig. 2, showing that the proposed security model exhibits a much lower packet loss rate than the classic routing protocol.

![Fig. 1. Trade-off analysis for packet delivery ratio.](image1)

![Fig. 2. Trade-off analysis for packet loss.](image2)
of the proposed secure protocol in terms of delay, is a must. Figure 4 presents the end-to-end delay performance of the proposed secure routing protocol.

6. Conclusions

The proposed protocol is characterized by a delay performance that is similar to that of classic or native WSN routing protocols. The proposed secure routing protocol is capable of ensuring the QoS required without suffering any significant delays, computational overheads, packet losses, retransmission rates or multifactor authentication. The key management model allows not only to avoid such security breaches as SCLA and impersonation attacks, but is also capable of ensuring the maximum possible QoS levels by guaranteeing higher packet delivery and minimum latency rates. The proposed model has exhibited satisfactory performance for WSN-based IoT systems.

References


Tabassum Ara is a Research Scholar in Computer Science at Reva University, Bangalore, Karnataka, India, with her research interests focusing on IoT, security and WSN. With 18 years of academic experience, she holds B.Eng., M.Sc. and M.Tech. degrees and is an Assistant Professor at the Department of Computer Science at HKBB College of Engineering in Bangalore.

https://orcid.org/0000-0002-3378-3713
E-mail: tabuara@gmail.com
School of Computing and Information Technology
Reva University
Bangalore, India

M. Prabhakar received his M.Sc. and Ph.D. degrees in Computer Engineering from Anna University, Chennai. He boasts 21 years of teaching experience and is currently working as an Associate Professor at the School of Computing & Information Technology, REVA University, Bangalore, India. His areas of research interest include adhoc networks and cybersecurity.

E-mail: prabhakar.m@reva.edu.in
School of Computing and Information Technology
Reva University
Bangalore, India