*I. J. Computer Network and Information Security*, **2022, 4, 60-74**
Published Online August 2022 in MECS (http://www.mecs-press.org/)
DOI:10.5815/ijcnis.2022.04.05

# Secured Wireless Sensor Network Protocol using Rabin-assisted Multifactor Authentication

**Javeria Ambareen**
School of Computing & Information Technology, REVA University, Bengaluru, India
E-mail: javeriaster@gmail.com

**Prabhakar M**
School of Computing & Information Technology, REVA University, Bengaluru, India
E-mail: prabhakar.m@reva.edu.in

Received: 08 June 2021; Revised: 11 August 2021; Accepted: 30 September 2021; Published: 08 August 2022

**Abstract:** Wireless sensor networks (WSNs) when combined with Internet-of-things (IoT) enable a wide range of applications across multiple domains. Sensor nodes in these wireless sensor networks like any other Internet-connected device are resource constrained and vulnerable to a variety of malicious attacks thereby compromising security. Consequently, a secure and efficient lightweight cryptographic protocol is required that can provide a balance between end-to-end security offering all features but yet lightweight. For secure data transmission and access, newer multi-factor authentication and key management features must be developed as majority of existing techniques have high computational overheads and are vulnerable to a wide range of attacks. In this paper, we propose a Rabin-assisted three-factor authentication protocol that uses the computational asymmetry of Rabin cryptosystem in addition to user password, smartcard and biometric for increased security. NS2 based simulation proves that the proposed protocol outperforms the baseline ad-hoc on-demand distance vector (AODV) protocol in terms of throughput, computation cost, and delay performance. Also, it has the ability to tolerate most common attacks and offers additional functionality features thereby offering a lightweight and highly secure protocol that can be extended to other critical domains like Smart Transportation Systems (STS), Smart grids, Smart buildings etc.

**Index Terms:** Wireless Sensor Network, internet-of-thing, multifactor authentication, rabin cryptosystem, smartcard, biometrics.

## 1. Introduction

IoT-integrated WSN systems need data that is secure when transmitted over insecure channels thereby bringing security to the forefront for researchers to address [1, 2]. As depicted in Fig 1, a *WSN* can be either distributed or hierarchical in nature. For distributed architecture, the nodes are deployed randomly and transmission of data takes place between them and a special node called as Manager Node or base station *(BS)*. In hierarchical architecture, a finite set of sensor nodes *(SN)* are installed as a cluster (group) with data transmission happening from them to a cluster head *(CH)*. More complex operations and broader transmission range exist in the cluster head and it interacts with the *BS*. The gateway node is an entry point to the WSN and acts as a gateway to other WSN as well. It acts as the overall network administrator.

*SN's* are different from typical computing devices with fewer resources to live on, wireless medium for communication and unattended mode of operation in unfriendly environments. As a result, the general security mechanisms cannot be applied to sensor networks directly. Given the hostile conditions of the target field of a sensor network, sensor nodes can be completely taken over by an adversary. Usually, the *SN's* are not made of rugged hardware [3] and for extracting stored data from the nodes simple power analysis techniques [4, 5] have been used. An attacker can further take advantage of the extracted information to clone new nodes or compromise other genuine nodes to introduce different attacks that consume the resources of the sensor network. To prevent WSNs from such attacks, various security mechanisms like key distribution, user authentication and user access control mechanisms are essential. The most critical design parameter catering to security is authentication. As shown in Fig. 2, for real-world applications access to data from sensor nodes becomes necessary due to the reason that cluster head or a gateway node collects data from sensor nodes only at some periodic intervals, thereby making the data stale. This way any external user can be provided real-time data on-demand provided they are authorized. Therefore, secure and efficient authentication schemes around sensor nodes are required. There could also arise a scenario whereby new nodes may need to be deployed on account of failures due to various reasons e.g. battery drained or a hardware failure or an adversary implanting a dummy

 

malicious node. Thus, new nodes have to prove their identity which requires strict authentication and fool-proof control mechanisms. Generally, the existing models proposed by researchers for user authentications in WSNs are prone to various well-known attacks and/or they are computing and communication heavy [6]. Security and efficiency are the essential attributes of a user authentication scheme in WSNs to make them feasible for using in real-time environments.

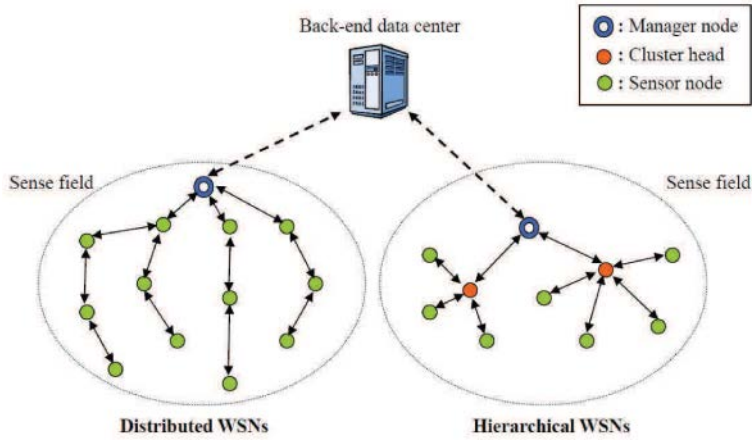


Fig.1. Architecture of a Distributed and Hierarchical wireless sensor network

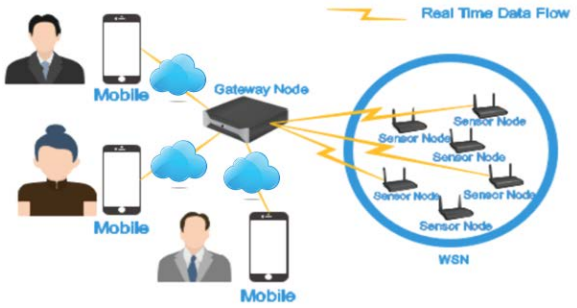


Fig.2. General architecture of device-IoT integrated wireless sensor network

Considering this inevitable need for a secure, robust and efficient IoT-integrated WSN, the paper focuses mainly on user authentication problem and proposes a novel multi-factor authentication protocol. The computational asymmetric characteristic demonstrated by Rabin cryptosystem is used which could make the model lightweight in comparison to other encryption algorithms like RSA and Diffe-Hellman etc. This characteristic suite well for the kind of architecture being targeted in this paper as depicted in Fig 2. As seen, devices which are mobile have fewer resources compared to the devices like the gateway node which have no such restriction.

Therefore, the key contributions of this paper are:

- For internet-connected WSN architecture, a Rabin-assisted lightweight three-factor user authentication and key agreement protocol.
- Simulate various attacks in NS2 and check if the proposed protocol has the ability to tolerate most of the common attacks
- A feature-by-feature comparison of the proposed protocol with other models for determining its efficacy
- Comparing the proposed protocol with baseline AODV protocol to check its performance in terms of communication and computational overheads

The paper is outlined as below. Section 2 details related work carried out by other researchers in this area. This is followed by Preliminaries and Notations used in section 3. Section 4 explains the proposed protocol in detail. Section 5 details the research methodology adopted. Section 6 enumerates the results and discusses them in context to existing work. Section 7 is a conclusion followed by reference papers used in this study.

## 2. Related work

Authentication process can be carried out by considering a single factor or multiple factors. Single factor authentication process is limited to systems which require less security. Authors in [7] have demonstrated an authentication model which considered two different factors, users and IoT devices that are involved in the

authentication process. This model lacks resistance to most of the authentication related attacks like Smart Card Loss Attack (SCLA), Offline Password guessing attack etc. These limitations have been addressed in [8] by designing a unique multifactor authentication algorithm specially designed for hierarchical IoT network (HIoTN). This scheme was quite expensive with respect to computational complexities, cost and communication overhead. These complexities can be reduced by using biometric features which was demonstrated by [9]. They designed an algorithm for heterogeneous WSN. However, smart card loss can be expected because the adversary can access the personal data stored on smart card. The network layer routing attacks on WSNs were detailed by authors in [9]. Their findings reveal that the majority of existing WSN secure routing systems are inefficient for a variety of reasons, including high energy consumption and significant communication overhead. Signature based authentication algorithms are also found in the same domain which is observed in [10]. They have developed an algorithm for most of the advanced IoT applications. Instead of single, a group key was established in one of the algorithm proposed by [11]. It focuses more on multicast communication specially in IoT where resource constrained devices are found. However it was found that information sharing was highly ambiguous and because of this ambiguity the efficacy of the network is limited. An Elliptic curve cryptography based authentication algorithm was devised by [12] which are meant for RFID devices in the health care sector. Authors in [13] offered a security solution for hierarchical wireless sensor networks based on trust. Using a combination of trust values in a cluster and fuzzy logic, the proposed solution was able to considerably increase security and prevent hostile or untrustworthy nodes from becoming the cluster head according to simulated results. Authors in [14] presented a key management scheme for flat networks and a hierarchical network in which the base station serves as the secure third party in charge of key distribution and network security management.

## 3. Preliminaries and Notations

### *3.1. Bio-hashing*

*Bio-hashing*: Biometric is one of the most promising techniques used to verify a user's identity [15]. It is quite advantageous over the conventional methods like password and smart card to authenticate the user. Data on biometric features are closely linked to each person and cannot be substituted. As a consequence, biometric data exposure results in extreme threats to privacy. Several systems are proposed to safeguard the biometric privacy. A bio-hash value $bh(k, b)$ is generated during the enrolment process using a biometric template and a random secret key $k$. In the biometric input signal, we have to make this function invariant to any little variations also, so pre-processing is performed on *b*. Then, to generate the bio-hash value $bh(k, b)$the inner product of the vector randomly generated using the unique secret key $k$of the user is compared with the function vector got vis-à-vis a predefined threshold. At the time of verification the received biometric signal $b'$is used to generate$bh(k, b')$, where $k$ is the secret key given by the user. Once it is generated, it is compared to $bh(k, b)$ that is stored.

### *3.2. Rabin cryptosystem*

A public key cryptosystem, Rabin is entirely based on integer factorization. It has three different phases- Key generation phase, Encryption phase and Decryption phase [16].

*Key generation phase*:$p, q \equiv 3(mod\ 4)$, such that the value of $p$ and $q$ are some random large prime numbers. The product of these two prime numbers is$N$. i.e. $N = pq$. The value of this $N$is a public key, and $(p, q)$pair is treated as the private key.

*Encryption*: For a given plain text *m* the cipher text *c* can be found as equation (1)

$$c = m^2 mod\ N \tag{1}$$

*Decryption*: The plain text can be recovered by decrypting the cipher text as given in equation (2)

$$m = \sqrt{c}\, mod\ N \tag{2}$$

In particular, the recipient knowing the private keys $(p,\ q)$ applies the Chinese remainder theorem to extract four possible plain texts$\{m_1, m_2, m_3, m_4\}$. There are many ways to find out a correct plain text among the four. One of the way or technique is padding technique, where a pre-defined padding is done in the plain text. For e.g. say $x$ is the square root of a number $y$, then it can be said that a solution exists such that $y = x^2 mod\ N$. Here $y$ is called the quadratic residue of $mod\ N$ [17].

### *3.3. Notations*

Notations made to use in the paper are described below in Table 1.

Table 1. Notations used in the research

| Notation | Description | Notation | Description | Notation | Description |
|---|---|---|---|---|---|
| $bio2h$ | Bio Hashing with 2 different factors | $p$ | Number (prime) | $pwu_i$ | User Pw |
| $k$ | Session Key | $GF$ | Galois Field | $ui$ | User |
| $b$ | Biometric value | $a, b$ | Coefficients of expression | $ad_j$ | Administrator node |
| $b'$ | Received Biometric value | $g$ | Point to generate | $ad_{IDj}$ | Administrator ID |
| $P, Q$ | Elliptic curve points | $r$ | Integer (random) | $ad_{xi}$ | Integer for administrator |
| $x, y$ | Two Co-ordinates | $pr$ | Private Key | $h()$ | Hashing function |
| $m$ | Normal plain Text | $pu$ | Public Key | $r_{shrd}$ | Random number (Shared) |
| $c$ | Cipher text | $u_{IDi}$ | User ID | $ee_i, ff_i, ll_i$ | Hashed value |
| $SCNumber_i$ | Smart card number | $Z_i$ | Arbitrary number | $msg_1, msg_2, msg_3, msg_4$ | Estimated intermediate Messages |
| $SCS$ | Smart Card Storing | $eID$ | Estimated ID | $t$ | Delay Threshold |
| $bio_{ui}$ | Bio information of user $U_i$ | $gn$ | Gateway Node | $msg$ | Message |
| $sc$ | Smart Card | $ID_{gn}$ | ID f gateway node | $sk_{gn}$ | Session Key of the GWN |
| $dd_i, zz_i$ | Parameter | $eei^{new}, ffi^{new}, ggi^{new}$ | Estimated hash values | | |

## 4. Proposed Protocol

The proposed three-factor authentication (3FA) protocol with key management is primarily discussed in this section. Rabin-assisted lightweight cryptosystem is used which does not compromise with the security of the scheme. Compared to other encryption methods like RSA and ECC, Rabin cryptosystem has a special feature of asymmetric computations[18,19]. In the proposed system the encryption process is quite light weight whereas the decryption has heavy computation. This proposed method is quite suitable in the wireless sensor network integration with IoT that we are looking at as the IoT devices are not rich in resources whereas the gateways devices have enough resources for this computation. Fig 3 depicts the complete flow-chart of the proposed system.

### *4.1. System Model*

Overall, there are seven phases in the proposed protocol which are explained below:

*A. System setup Phase (Rabin-assisted cryptosystem establishment)*

***Step 1:*** In this inception step, two unique prime numbers *p* and *q* are generated by the admin node (a*dm)* and the product is determined as $N = pq$. These two values are considered as the private key, and then the admin node chooses a key called master key (say $x_{GN}$) and an additional integer value $(2^4 \leq l \leq 2^8)$which is primarily meant for the receiver side to verify the password locally. It also makes use of another module called Fuzzy Verifier. The public key is calculated as given in equation (3)

$$c = [(r.g), (m + r.pu)] \tag{3}$$

This method of finding public key involves an initial point on the curve. In this method, the initial (say, first) point of the Cipher text pair $(r.g)$ is multiplied with the private key$(pr)$. The resultant is then subtracted from the second point in the Cipher text pair.

***Step-2:*** In this step, $adm$ chooses an identity $ad_{IDj}$ and determines the fresh secret key as in equation (4)

$$admn_{X_j} = h\left(admn_{IDj} \parallel x_{GN}\right) \tag{4}$$

*for* $admn_j (1 \leq j \leq m)$.

***Step-3:*** In this step $adm$ generates a number $r_{shrd}$ randomly, and the same is communicated to gateway node and the node$admn_{j,}$ the $admn_j$ node stores this information in memory$\left\langle admn_{IDj} admn_{X_j}, r_{shrd}\right\rangle$.

*B. User Registration Phase*

During this phase, user registration takes place, wherein the $i^{th}$ user performs the following three steps where the *adm* registers the node.

***Step-1:*** The *ith* user $u_i$ shares his identity $U_{IDi}$. Additionally, it shares its personal credentials through some secure communication channel.

***Step-2:*** Once the admin receives this information from the new user, the admin cross verifies for the uniqueness of the user and his identity by comparing the data existed in the database. Once verified and confirmed the uniqueness, the $adm$ acknowledges the same to the user about his new identity. The $adm$then selects $x_I$ as an arbitrary number and then finds value of the variable $d_i = h(x_{GN} \parallel u_{IDi} \parallel x_i)$ and $ll_i = h(x_{GN} \parallel sc_{N_i})$, where $sc_{N_i}$ is a user or a node's smart card number. To estimate $\mathrm{d}^I$ and $\mathrm{ll}^I$ values, $adm$ enables smart card storing $SCS$ in the following format $\langle d_i\, ll_{i,}, SCnumber_i l, n, bio2h(.,.), h()\rangle$, which is assigned to the user $u_I$ securely. Thus, $adm$ maintains a database for each $u_i's$personal credentials$\langle u_{IDi}, sc_{N_i}, x_i, Personal\ credential\rangle$.

***Step-3:*** In this step the card information is read using a card reader and in addition to this information the user also provides user id and password $\langle u_{IDi}, pwu_i\rangle$ and as an additional factor it considers and adds bio-metric information of the user$bio_{U_i}$. Smart card now selects a number $r_i$ which is normally a random number and then it determines the attribute$bioz_i = bio2h(r_i, bio_{U_i})$.

*C. Login Phase*

Once registration is done the user can login to the system, wherein the user has to enter its credentials. Any sensor data can be fetched by the user using these steps.

***Step-1:*** In this step,$u_i$ connects his card $SC$ and enters its credentials$u_{IDi}{}^*$, password $pwu_i{}^*$ and its fingerprint information $bio_{U_i}$. Now, the deployed smart card $SC$ estimates $bioz_i{}^* = Bio2h(r_i^*, bio_{U_i})$ and a variable $ee_i^* = h(h(\ pwu_i{}^* \parallel u_{IDi}{}^* \parallel bioZ_i{}^*)modl)$. If $ee_i^* \neq ee_i$, the card rejects the user and the data from the sensor will not be able to be accessed by the user.

***Step-2:*** In this phase an arbitrary number $z_i$is generated by the smart card, it also adds one more important factor i.e. timestamp $t_{stamp1}$, and it estimates the following values as given in equations (5)

$$\left.\begin{aligned} d_i^* &= f\,f_i \oplus h(pwu_i^* \parallel u_{IDi}^* \parallel bioz_i^*) \\ ll_i^* &= g_i \oplus h(pwu_i^* \oplus u_{IDi}^* \oplus bioz_i^*) \\ ll_i^* &= g_i \oplus h(pwu_i^* \oplus u_{IDi}^* \oplus bioz_i^*) \\ msg_1 &= (u_{I_{Di}} \parallel sc \parallel_{(Ni)} z_i)^2 mod\ n \\ msg_2 &= h(d_i^* \parallel ll_i^* \parallel z_i \parallel t_{s_{tamp1}}) \end{aligned}\right] \quad (5)$$

***Step-3:*** In the final step of this phase the user selects the sensor's $ad_{IDJ}$from which the user wishes to access the data, and the $sc$estimates value $eid_j = admn_{IDj} \oplus h(u_{IDi} \parallel z_i \parallel t_{stamp1})$, this information is transmitted to the *gateway node* as $Msg_1 = \langle msg_1,\ msg_2, t_1, eid_j\rangle$.

*D. Authentication Phase*

To perform mutual authentication that is both secure and valid and session key agreement amongst the three stakeholders (i.e.$u_i$,$gn$ and$admn_j$), our proposed protocol performs the following processes:

***Step-1:*** Once receiving $Msg_1$ from $u_i$, $gn$ firstly deciphers $msg_1$ sing $p$and $q$to get $u_{IDi}{}'$,$sc_{Ni}{}'$ and $z_i{}'$. Here, the value of $x_i$ is got based on $u_{IDi}{}'$ which then is checked and compared with entries in $sc_{Ni}{}'$. If the values are different, $gn$ doesn't accept the request and stops the session else finds the following values as given in equation (6):

$$\left.\begin{aligned} ll_i^{'} &= h\left( sc_{N_i}^{'} \oplus x_{GN} \right) \\ d_i' &= h(u_{1Di}' \oplus x_{GN} \oplus x_i) \\ z_i' &= msg_2 \oplus h(d_i' \oplus t_{stamp1}) \\ msg_2^{'} &= h\left( d_i^{'} \oplus ll_i^{'} \oplus z_i^{'} \oplus t_{stamp1} \right) \end{aligned}\right] \quad (6)$$

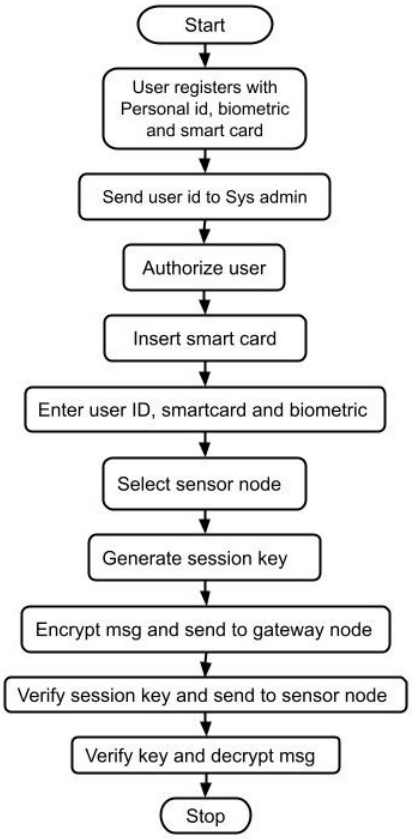


Fig.3. Flow-chart of the proposed multi-factor authentication protocol

In case of $msg_2^{'} \neq msg_2$, $gn$comes out of the on-going session. Alternatively, if $msg_2^{'} = msg_2$ it calculates the following as given in equations (7):

$$\left.\begin{aligned} admn_{IDj}^{'} &= uid_j \oplus h(z_1 \oplus u_{IDi} \oplus t_{stamp1}) \\ \mathrm{admn}_{\mathrm{X_j}}^{'} &= \mathrm{h}\left(\mathrm{admn}_{\mathrm{IDj}}^{'} \oplus \mathrm{x}_{\mathrm{GN}}\right) \\ msg_3 &= h\left(u_{IDi}^{'} \oplus admn_{IDj}^{'} \oplus id_{GN} \oplus admn_{X_j}^{'} \oplus z_i^{'} \oplus t_{stamp2}\right) \\ msg_4 &= u_{IDi}^{'} \oplus h\,(id_{GN} \oplus admn_{X_j}^{'} \oplus t_{stamp2}) \\ msg_5 &= z_i \oplus h(u_{IDi}^{'} \oplus admn_{IDj}^{'} \oplus admn_{X_j}^{'} \oplus t_{stamp2}) \end{aligned}\right] \quad (7)$$

*Once estimating the above parameters, $gn$ transmits message, $Msg_2 =< id_{GN}, msg_3, msg_4,\ t_{stamp1} >$ to $admn_j$*

***Step-2:*** On receiving the message $msg_2$, in case of affirmation $\mathrm{ad}j$ stops the session, otherwise estimates the following as given in equations (8)

$$\left.\begin{aligned} u_{IDi}^{**} &= msg_4 \oplus h(id_{GN} \| ad_{X_j} \| t_{stamp2}) \\ z_i^{**} &= msg_5 \oplus h(u_{IDi}^{**} \| ad_{IDj} \| ad_{X_j} \| t_{stamp2}) \\ msg_3^{'} &= h(u_{IDi}^{**} \| ad_{IDj} \| id_{GN} \| ad_{X_j} \| z_i^{**} \| t_{stamp2}) \end{aligned}\right] \quad (8)$$

*Now, $ad_j$ comes out of the session in case of $msg_3^{'} \neq msg_3$ otherwise it considers genuine the user $u_i$ and $gn$. Authenticating the user $u_i$ and $gwn$, $ad_j$ determines the values as given in equations (9)*

$$\left.\begin{aligned} sk_j &= h(u_{IDi}^{**} \| ad_{IDj} \| z_i^{**} \| z_j) \\ msg_6 &= h(sk_j \| ad_{X_j} \| z_j \| t_{stamp3}) \\ msg_7 &= z_i^{**} \oplus z_j \end{aligned}\right] \quad (9)$$

where $z_j$ is randomly generated number by $ad_j$
$ad_j$ then forwards $Msg_3 =< msg_6, msg_7, t_3 >$ to the gateway node $gn$ in the ascending phase.

***Step-3:*** In this phase $gwn$ verifies whether the difference value of $|t_{stamp4} - t_{stamp3}| \leq \Delta t$ (delay threshold) exists, where $t_{stamp4}$ refers the current timestamp. In case the condition doesn't holds the criteria $|t_{stamp4} - t_{stamp3}| \leq \Delta t$), $gn$ terminates the session. In case of $|t_{stamp4} - t_{stamp3}| \leq \Delta t$ it determines $z_j^{'} = msg_7 \oplus z_i^{'}$, $\mathrm{sk_{GN}} = h(\mathrm{z^{'}} \| \mathrm{ad_{IDj}} \| u_{\mathrm{IDi}} \| \mathrm{z^{'}})$ and $msg_6^{'} = h(sk_{GN} \| \mathrm{ad}_{X_j}^{'} \| z_i^{'} \| t_{stamp3})$

In case $msg_6^{'} \neq msg_6$, $gn$ stops the session and calculates $msg_8 = h(sk_{GN} \| u_{IDi}{}' \| d_i^{'} \| z_i^{'})$.
Therefore, $Msg_4\langle msg_7, msg_8 \rangle$ to $u_i$ is transmitted by $gn$

***Step-4:*** *Th* $u_i$ estimates $z_j^* = m_7 \oplus z_i$, $sk_i = h(u_{IDi} \| ad_{IDj} \| z_i \| z_j^*)$ when it receives $msg_4$ and $msg_8 = h(sk_i \| u_{\mathrm{IDi}} \| d_i \| z_j^*)$.

Here, if $msg_8^{'} \neq msg_8$, $u_i$ dismisses the session, or else it considers that the gateway-node $gwn$ and $ad_j$ are genuine. A combined session key $sk_i = sk_j = sk_{GN}$ is then generated immediately amongst the contributing $u_i$, $gwn$ and $ad_j$

*E. Identity Update Phase*

IoT-integrated wireless sensor network being a dynamic network may require updating user details and hence in this phase, the identity is securely updated by a registered user as shown below:

***Step-1:*** The user $u_i$ inserts the card and punches-in its identity $id_i^*$ in conjunction with other authenticating parameters like password $pwu_i^*$ and fingerprints $bio_{U_i}$. After this the smart card computes two-factor hashing $b_i^* = bio2h(r_i, bio_{U_i})$ [40] along with $ee_i^* = h(u_{IDi} \| pwu_i \| bioz_i) modl$. Thus, in case of $ee_i^* \neq ee_i$, it rejects $u_i$ login request and thus avoids any unauthenticated access to the sensed data across the network. It is then followed by estimation of a new *id* for $u_i$ where it inputs updated identity $u_{IDi}^{new}$ and simultaneously generates a timestamp $t_{\mathrm{stmpid}}$, in addition to the following as given in equations (10)

$$\left.\begin{aligned} di^* &= f_i \oplus h(pwu_i^* \| u_{IDi}^* \| bioz_i^*) \\ ll_i^* &= g_i \oplus h(pwu_i^* \| u_{IDi}^* \| bioz_i^*) \\ dd_i &= (u_{IDi} \| sc_{N_i} \| u_{IDi}^{new})^2 modn \\ z_i &= h\left(d_i^* \| ll_i^* \| u_{IDi}^{new} \| t_{stampid}\right) \end{aligned}\right] \quad (10)$$

Once estimating the above derived parameters, smart card *sc* transmits a new updated signal $\langle z_i, dd_i, t_{stamp\_id} \rangle$ to $gn$

***Step-2:*** In this step, $dd_i$ is decrypted by $gn$ using $p$ and $q$ to obtain $u_{IDi}^{'}$, $sc_{N\_i}^{'}$, $u_{IDi}^{new}$. Furthermore, it retrieves

$x_i$ according to $u_{IDi}^{'}$, and verifies whether $sc_{N_i}^{'}$ matches the value in the entry. In case these two parameters do not match, $gwn$ terminates; otherwise, it estimates the following as given in equations (11):

$$\left.\begin{aligned} ll_i^{'} &= h(sc_{N\_i}^{'} \| x_{GN}) \\ d_i^{'} &= h\left(u_{IDi}^{'} \| x_{GN} \| x_i\right) \\ z_i^{'} &= h\left(d_i^{*} \| ll_i^{*} \| u_{IDi}^{new} \| t_{stampid}\right) \end{aligned}\right] \tag{11}$$

Noticeably, in case of $z_i^{'} \neq z_i$ , $gn$ aborts the current session; else it estimates $\mathrm{d_i^{**}} = \mathrm{h}\left(\mathrm{u_{IDi}^{new}} \| \mathrm{x_{GN}} \| \mathrm{x_i}\right)$, $y_i = h(d_i^{'} \| t_{stamp\_id}) \oplus d_i^{**}$ and $zz_i = h\left(L_i^{'} \| u_{IDi}^{new} \| d_i^{**} \| d_i^{'} \| t_{stamp\_id}\right)$

Once estimating the above stated variables, $gwn$ transmits, $< zz_i, y_i >$to the card and updates $u_{IDi}^{new}$ to the database.

***Step-3****:* In this phase, the card estimates the following and verifies to check if $zz_i^{*} = zz_i$ exists and calculates as per equations (12)

$$\left.\begin{aligned} d_i^{**} &= y_i \oplus h\left(d_i^{*} \| t_{stamp\_id}\right) \\ zz_i^{*} &= h\left(d_i^{**} \| d_i^{*} \| ll_i^{*} \| u_{IDi}^{new} \| t_{stamp\_id}\right) \\ zz_i^{*} &= h\left(d_i^{**} \| d_i^{*} \| ll_i^{*} \| u_{IDi}^{new} \| t_{stamp\_id}\right) \\ zz_i^{*} &= h\left(d_i^{**} \| d_i^{*} \| ll_i^{*} \| u_{IDi}^{new} \| t_{stamp\_id}\right) \end{aligned}\right] \tag{12}$$

In case of, $zz_i^{*} = zz_i$ the SC card estimates as per equations (13):

$$\left.\begin{aligned} ee_i^{new} &= h(h\left(pwu_i \| bioz_i \| u_{IDi}^{new}\right) \\ gg_i^{new} &= l_i \oplus h\left(pwu_i \oplus bioz_i \oplus u_{IDi}^{new}\right) \end{aligned}\right] \tag{13}$$

Once, estimating these variables *sc* card replaces the old information with $< ee_i^{new}, ff_i^{new}, gg_i^{new} >$

*F. Password Change Phase*

Password change and update to retain security of the user is a commonly desired feature in IoT-integrated WSN. Practically, the password $pw_i$ is updated locally by an authorized user $u_i$. We have incorporated the following approach in the protocol to perform user's password change.

***Step-1:*** In this phase, *sc* is inserted into a card reader and performs login as discussed in previous phase that enables verification of the genuineness of the biometric variable or fingerprint, *pw*, and his identity.

***Step-2:*** In this phase the user $u_i$ inputs a new *pw*, $pw_i^{new}$ and estimate's the following as per equations (14)

$$\left.\begin{aligned} ee_i^{new} &= h\left(h\left(u_{IDi}^{new} \| pwu_i \| bioz_i\right) modl\right) \\ d_i^{'} &= f_i \oplus h(\left(u_{IDi}^{new} \| pwu_i \| bioz_i\right) \\ ff_i^{new} &= d_i^{'} \oplus h(\left(u_{IDi}^{new} \| pwu_i \| bioz_i\right) \\ ll_i^{'} &= gg_i \oplus h\left(\left(u_{IDi}^{new} \oplus pwu_i \oplus bioz_i\right)\right) \\ gg_i^{new} &= ll_i^{'} \oplus h\left(u_{IDi} \oplus pwu_i^{new}\right) \end{aligned}\right] \tag{14}$$

***Step-3:*** In this phase, the initial values $< ee_i, ff_i, gg_i >$ estimated are updated as. $\langle ee_i^{new}, ff_i^{new}, gg_i^{new} \rangle$

*G. Smart Card Revocation Phase*

In a lot of scenarios, the nodes or users can be in mobile ad-hoc networks with the probability of card getting lost. In such cases if $u_i's$ card is lost or stolen, the user $u_i$ can retain a new smart card $sc$ by executing the following procedures:

***Step-1:*** In this phase, the selected user $u_i$ provides its identity $u_{IDi}$ in addition to the personal credentials to the admin through a secure channel. In this step, initially $adm$ verifies the credentials provided by $u_i'$. In case of positive or valid user information, $adm$ estimates the following attribute as per equation (15):

$$\left.\begin{aligned} d_i^{new} &= h(u_{IDi} \oplus x_{GN} \oplus x_i) \\ ll_i^{new} &= h\left( sc_{N\_i}{}^{new} \oplus x_{GN} \right) \end{aligned}\right] \tag{15}$$

Now $< d_i^{new},\ ll_i^{new}, sc_{N_i}{}^{new}, ll, n,\ \ bio2h\ (.,.)h(), >$is shared with the user securely. Then $admn$ then stores this information into the database with the new smart card details $sc_{N\_i}{}^{new}$

***Step-2:*** In this phase, the user makes use of this smart card and enters, $<\ u_{IDi}, pwu_i >$ and $fng_i$. The smart card selects $r_i$ a random number. It finds $b_i = \text{bh}(\text{finger}^i, r_i)$ , $ee_i^{new} = h(h(pwu_i \ \|\ u_{IDi} \ \|\ bioz_i)modl)$ , and $ff_i^{new} = d_i^{new} \oplus h((pwu_i \ \|\ u_{IDi} \ \|\ bioz_i)$.

Finally, the smart card stores $< ee_i^{new},\ ff_i^{new}, gg_i^{new},\ sc_{N\_i}{}^{new},\ ll, n, r_i\ , bio2h(.,.)\ h(),\ >$ into its database and removes the earlier information $\langle d_i^{new},\ ll_i^{new} \rangle$

## 5. Research Methodology

A qualitative and quantitative evaluation is carried out. Under qualitative evaluation, we check the efficacy of the model to avoid various attack conditions, both theoretically and confirming by simulating the various attacks. Theoretically, we compare with other published reference works. Authors in [21] have compared the features and attack resiliency with authors in [22-24]. We use the tabulation data to compare the proposed model on all those features and more. In quantitative evaluation, we simulate the proposed model and the native AODV routing protocol. Below are details of the simulation environment and how an attack is simulated.

The proposed protocol is simulated in Network simulator 2 (NS2) environments. TCP, UDP, and a variety of additional routing protocols can all be simulated using it. It uses the Tool Command language, which is an object-oriented programming language (*OTcl*). The *OTcl* programming language may be used to create various network topologies and routing protocols. The simulation interface scripts are provided by the C++ environment, and the main NS programme simulates that topology with specified parameters. In NS2, the network animator (NAM) gave a graphical representation of the network. The control options in the NAM interface allowed us to fast forward, pause, stop, and play the simulation of various attacks. The system used is a Pentium dual core with 1GB RAM, 120GB hard disk, 15" LED monitor, a keyboard and a mouse. The Operating system used was Windows XP, Implementation- NS2, Front end- *OTcl* and Cygwin tool to simulate in Windows. A brief simulation methodology is shared below:

*Creating and configuring the connection*

Any number of dynamic nodes can be constructed. The source, destination, and malicious node can all be entered. It is possible to generate node movement and divide nodes into zones. Protocols like TCP and UDP can be used to establish a connection between the nodes. TCP stands for Transmission Control Protocol.

Node Creation:

```
for{set a 0} {$a<$val(m)} {incr a}
{set node_($a) [$s node]
$node_($a) random-motion 1}
```

TCP connection establishment:

```
Set gptrace [open gptrace.tr w] Set tcp [new Agent/TCP]
$tcp trace rtt_
$tcp trace cwnd_
$tep attach $gptrace
```

Area defining:

*$node_(1) setX_25.000000000000*
*$node_(1) setY_0.000000000001*
*$node_(1) setX_0.000000000001*

Initiating malicious node:

*$n at 40.0 "[$node_ (10) set ragent_] malicious*

Example: Simulation of replay attack

A Node 22 is created as the user node, it tries to log into the gateway node by sending the required credentials and gets logged-in successfully. Next, we make another node say Node 21 (created as malicious node in simulation) to login to the same gateway by resending the same credentials which Node 22 has sent. The gateway normally stores the value *F* and respective timestamp at the time of successful login. When gateway finds that node 21 also tried to login with same credentials as node 22, the gateway verifies the value of *F* and timestamp. If it matches it denies the login request confirming it is a replay attack. Fig. 4 is a snapshot of the simulation carried out in NS2 environment.

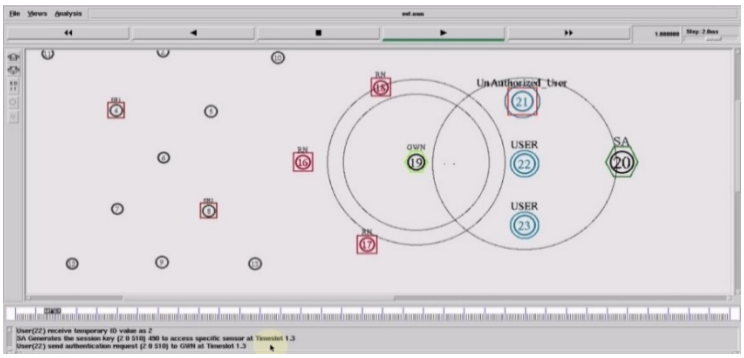


Fig.4. Snapshot of replay attack in NS2 environment

## 6. Results and Discussion

Results are tabulated based on qualitative and quantitative evaluation as mentioned above.

### *6.1. Qualitative evaluation*

Under this study, the proposed protocol is evaluated by checking resiliency against various attack conditions and comparing with other published reference works [25-27]. We also simulate the various attack conditions in the proposed protocol to confirm its resiliency.

#### *A. SCLA Resilience*

Consider that an attacker say, opponent gets the smart card SC detailed information containing $< ee_i,\ ff_i,\ g_i,\ sc_{N_i},\ ll,\ n,\ r_i,\ bio2h\ (.,.),\ hash1(\ ) >$of the valid user $u_i$ where, $e_i = h(h(pwu_i \parallel u_{IDi} \parallel bioz_i)modl)$ and $gg_i = ll_i \oplus h(u_{IDi} \oplus pwu_i \oplus bioz_i)$ Though with above retrieved values *A* can guess user credential $u_{IDi}{}^*$ and $pwu_i{}^*$ thereby we can find $ee_i^* = h(h(pwu_i{}^* \parallel u_{IDi}{}^* \parallel bioz_i)modl)$, it is observed that the exact value of $u_{IDi}{}^*$ and $pwu_i{}^*$cannot be retrieved as $e_i$ used is a fuzzy verifier. Thus, SCLA attacks are avoided. When it concerns SCLA type-2 attack, the following is adopted:

Assuming the adversary node *A* has deciphered message $Msg_1 = \langle msg_1, msg_2, t_1, eid_j \rangle$transmitted by $u_i$ when performing login, then as per equations (16)

$$\left.\begin{aligned} d_i^* &= ff_i \oplus h\left( pwu_i{}^* \parallel u_{IDi}{}^* \parallel bioz_i{}^* \right) \\ ll_i^* &= gg_i \oplus h\left( pwu_i{}^* \oplus u_{IDi}{}^* \oplus bioz_i{}^* \right) \end{aligned}\right] \tag{16}$$

It is worthwhile to note that $g_i$ is got from the smart card $sc$ of the user$u_i$. Since the quadratic residue problem is quite complex[28], it becomes impractical to find the value of $r_1$ by the adversary with the help of $msg_1 = (u_{IDi} \parallel SCNumber_i \parallel z_i)^2 modn$ and hence it stops A to estimate the value $msg_2^* = h(d_i^* \parallel ll_i^* \parallel z_i \parallel t_1)$, that is mandatorily required for verification of user $u_{IDi}{}^*$ and$pwu_i{}^*$. Thus, any SCLA type-2 attack is resisted by the proposed protocol.

*B. Known Session Information Attack (KSIA) resilience*

In existing methods have used the value which is static in nature, $h\left(u_{\mathrm{IDi}}{}' \parallel \mathrm{admn}^{X_j}{}'\right)$ to get the ephemeral arbitrary (EA) numbers, where $admn_{X_j}{}'$ signifies the (nodes) key shared between $ad_j$ and $gw$. Therefore, revealing the "ephemeral random" (ER) number $z_i$ causes compromise of the static value $h\left(u_{IDi}{}' \parallel admn_{X_j}{}'\right)$ that eventually leads to a compromise of EA numbers. To mitigate this risk, our proposed protocol introduces the concept of timestamp and one-way hashing. It estimates $msg_5 = z_i \oplus h\left(u_{IDi}{}' \parallel admn_{IDj}{}' \parallel admn_{X_j}{}' \parallel t_2\right)$ which is in fact varying or changing in each authentication phase and thus avoids endangering the ephemeral arbitrary number in other authentication sessions, thus avoiding these types of attacks.

*C. User Impersonation Attack resilience*

Under proposed methodology, the adversary A is unable to perform any user impersonation attack. Assuming adversary A somehow gets hold of $u_i's$ smart card *SC* and lays hand on information like $< ee_i, ff_i, gg_i, sc_{N_i}, ll, n, r_i, bio2h(.,.), hash1() >$. The proposed protocol makes sure that the adversary A to have all the features which includes $pwu_i$, smart card SC, and biometric information which is used to produce the message $Msg_1 = \langle msg_1, msg_2, t_1, eid_j \rangle$ from the practical point of view, the value m2 signifies the authenticity of the user $u_i$, where $msg_2 = h(d_i^* \parallel ll_i^* \parallel z_i \parallel t_1)$. The message $msg_2$ encompass $d_i^* = ff_i \oplus h(pwu_i{}^* \parallel u_{IDi}{}^* \parallel bioz_i{}^*)$ and $ll_i^* = g_i \oplus h(pwu_i{}^* \oplus u_{IDi}{}^* \oplus bioz_i{}^*)$. But it is noteworthy that without sharing $\mathrm{pwu_i}$, *SC* and biometric information the adversary cannot estimate $d_i^*$ or $ll_i^*$.

*D. Gateway Impersonation Attack Resilience*

An adversary A is capable of impersonating by pretending like $gw$ to either user $u_i$ or $ad_j$ in the protocol[29]. To impersonate as $gw$ and $ad_j$, the intruder A requires estimating $msg_3 = h(u_{IDi}{}' \parallel ad_{IDj}{}' \parallel id_{GN} \parallel ad_{X_j}{}' \parallel z_i{}' \parallel t_2)$. The adversary cannot estimate the value of m3 without the knowledge of $h(u_{IDi}{}' \parallel x_{GWN})$ it is not possible for the adversary to estimate the value of $msg_3$ . also, it ensures that the adversary cannot retrieve any information from the previous session, since it uses the hash algorithm and the time stamping technique. On the other hand, impersonating as $gw$ to the user $u_i$, the adversary needs to estimate a valid factor $msg_8 = h(z_{Session_{GN}} \parallel u_{IDi}{}' \parallel d_i' \parallel z_j'))$. To achieve it, adversary needs to have information about $z_i$ which also helps in the estimating the values $z_{Session_{GN}} = h(u_{IDi}, \parallel ad_{IDj} \parallel z_i)$. In order to retrieve the value of $z_i$ adversary *A* needs to know the secret key of $gn$. This is impractical as the secret key is preserved or protected by the administrator. From further exploring it is found that the adversary A can impersonate by decrypting $m_1 = (u_{IDi} \parallel sc_{N\_i} \parallel z_i)^2$ mod; however it is highly cumbersome. Hence the proposed model avoids any possibility of the gateway node impersonation attack.

*E. Modification Attack Resilience*

The adversary A is unable to make any modification in the messages $Msg_1 = \langle msg_1, msg_2, t_1, eid_J \rangle$, $Msg_2 = \langle id_{GN}, msg_3, msg_4, msg_5, t_2 \rangle$ $Msg_3 =< msg_6, msg_7, t_3 >$ *or* $Msg_3 =< msg_7, msg_8, t_3 >$ in the proposed model. Assume that an adversary A can intercept any one of the message parts, modify it and forward it further. But since each message in our model is protected by a hash value and normally this is estimated using a secret value, hence it is difficult to retrieve this value by the adversary. For e.g., in $\mathrm{msg_1}$, the intruder A cannot estimate the value of $msg_2 = h(d_i^* \parallel ll_i^* \parallel z_i \parallel t_1)$ , because $d_i^* = ff_i \oplus h(pwu_i{}^* \parallel u_{IDi}{}^* \parallel bioz_i{}^*)$ and $ll_i^* = g_i \oplus h(pwu_i{}^* \oplus u_{IDi}{}^* \oplus bioz_i{}^*)$ the receiver can detect if any message is made by estimating the value of each message, hence the model is modification resilient.

*F. Replay Attack Resilience*

In IoT, there can be the possibility of mobile nodes where the attacker can try to forward any old replay message sent by any particular user or stakeholder. However, in proposed protocol with timestamp implemented enables resisting replay attacks. Here, the messages are protected by a hash value which is calculated by means of a shared secret key in-between the transmitter and receiver. For illustration, the messages are protected like, $Msg_1 =< msg_1, msg_2, t_1, eid_J >$ , $Msg_2 =< id_{GN}, msg_3, msg_4, msg_5, t_2 >$ , and $Msg_3 =< msg_6, msg_7, t_3 >$ . In this manner, the attacker module A is unable to bypass the time stamp (here, $t_1$, $t_2$ and $t_3$). In case an intruder tries to replay the previous message, receiver would be able to identify it instantly by verifying its hash value and related timestamp.

*G. Verification Credential attack Resilience*

There can be the scenario when an adversary can attack authentication server and can retrieve or steal the

verification information including hashed password. Noticeably, our proposed protocol enables server to retain the attributes like $< u_{IDi}, sc_{N\_i}, x_i, Personal\ info >$ with no password related information stored. If the authentication server is compromised still the adversary cannot access the information regarding the password.

*H. Mutual Verification*

In IoT-integrated WSN, nodes require authenticating each other. Practically, an intruder can't easily retrieve$m = h(\ d_i^* \parallel ll_i^* \parallel z_i \parallel t_1)$ without the knowledge of private key, $d_i^*$ and $ll_i^*$. In this scenario the gateway node $gn$ can't authenticate $u_i$ by verifying precision of $msg_2$. Similarly, user $u_i$ can verify gateway node $gn$ by checking the correctness of $msg_8 = h(z_{Session_{GN}} \parallel u_{IDi}{}' \parallel d_i' \parallel z_j{}')$. In this manner, the user $u_i$ and the gateway node can authenticate each other. Furthermore, $ad_j$ authorizes $gn$ by checking correctness of $msg_3 = h(u_{IDi}{}' \parallel ad_{IDj}{}' \parallel id_{GN} \parallel ad_{X_j}{}' \parallel z_i' \parallel t_2)$. Similarly, $gn$ can authorize $ad_j$ by verifying if $msg_6 = h(z_{Session_j} \parallel ad_{X_j} \parallel z_j \parallel t)$ is correct. This ensures that the protocol proposes mutual authentication between gateway node $gn$ and the$ad_j$.

*I. Node or User's Anonymity*

Anonymity of the nodes or the users is vital in IoT-integrated WSN environment [30,31]. There can be a situation when the intruder A can retrieve the messages inter-communicated by the user and tries to identify it, which from the proposed protocol, it can be found in $msg_1 = \left(m_{IDi} \parallel sc_{N_i} \parallel z_i,\right)^2 \ mod\ n$. To get user identity $u_{IDi}$, while executing the Rabin assisted test of the gateway, the intruder needs to have knowledge of secret key. But in practice, it is infeasible to get it, since it is preserved with administrator. Also, the computational-hardness of Quadratic Residue Problem (QRP) does not allow the intruder at any cost to obtain $u_{IDi}$ by decrypting the value $msg_1 = \left(u_{IDi} \parallel sc_{N_i} \parallel z_i,\right)^2 \ mod\ n$. In this way, the proposed protocol preserves the user anonymity across the network. Table 2 compares the proposed protocol with other reference work on key security features.

Table 2. Compassion of proposed protocol security features with published reference work

| Security features | Authors in [8] | Authors in [9] | Authors in [10] | Authors in [11] | Authors in [5] | Proposed |
|---|---|---|---|---|---|---|
| SCLA resiliency | ☒ | ☒ | ☒ | ☑ | ☒ | ☑ |
| KSIA resiliency | ☒ | ☒ | ☒ | ☑ | ☑ | ☑ |
| User impersonation attack resiliency | ☒ | ☒ | ☒ | ☒ | ☑ | ☑ |
| Gateway impersonation attack resiliency | ☑ | ☒ | ☒ | ☒ | ☑ | ☑ |
| Modification attack resiliency | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| Replay attack resiliency | ☒ | ☒ | ☒ | ☒ | ☑ | ☑ |
| Verification credential attack resiliency | ☑ | ☑ | ☑ | ☑ | ☑ | ☑ |
| Mutual verification | ☑ | ☒ | ☒ | ☒ | ☑ | ☑ |
| Node or user's anonymity | ☑ | ☒ | ☒ | ☑ | ☑ | ☑ |
| Three-factor authentication | ☑ | ☒ | ☒ | ☒ | ☒ | ☑ |

It is observed that the protocol meets all the security parameters compared to existing models.

*6.2. Quantitative evaluation*

In this section, we simulate the model and assess its efficacy on three counts as discussed below.

*Simulation:*

The simulation consists of 24 mobile nodes working cooperative across the region. After implementing the 3FA, we compared it to the native Ad-hoc On-Demand Distance Vector (AODV) routing protocol and plotted performance metrics such as throughput, packet delivery ratio, and end-to-end delay. In all three cases, TCL script is used to create a wireless network with a communication model. When the *tcl* script is run, it creates a trace file that contains all of the simulation events. The needed parameter is determined using an *awk* script that processes the trace file and outputs *a .xg file* as a result. The results *(.xg file)* generated from the execution of the *awk* script are used to plot an *Xgraph* for the parameters. The *'parameter_EXI.tr'* and *'parameter_PRO.tr'* for existing (*EXI*) and proposed (*PRO*) protocols is plotted as shown in figures below.

*Throughput*

The number of correctly received packets in a unit of time is measured in bits per second (bps), as seen in Fig 5.

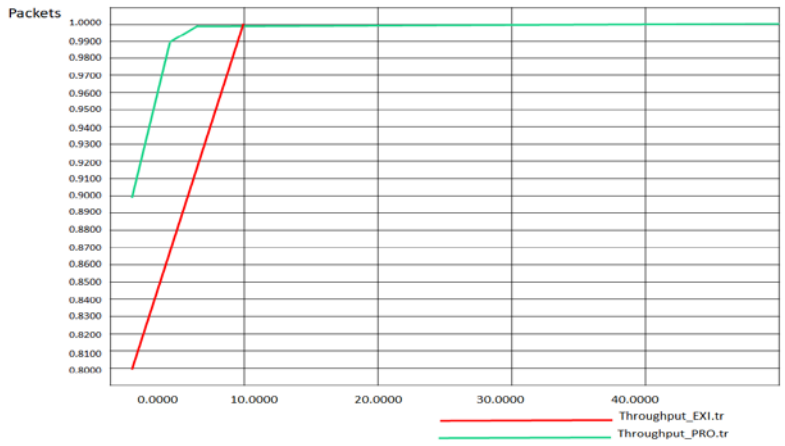


Fig.5. Throughput of the proposed protocol vs. existing AODV protocol

*Computation cost*

The Packet Delivery Ratio (PDR), which is the number of received and created packets as recorded in the trace file [32], is used to compute the computation cost. As demonstrated in Fig 6, computation cost is defined as the ratio between the destination's received packets and the source's created packets.

*End-to-end delay*

As shown in Fig 7, it is the difference between the time when the sender generated the packet and the time when the receiver received the packet.

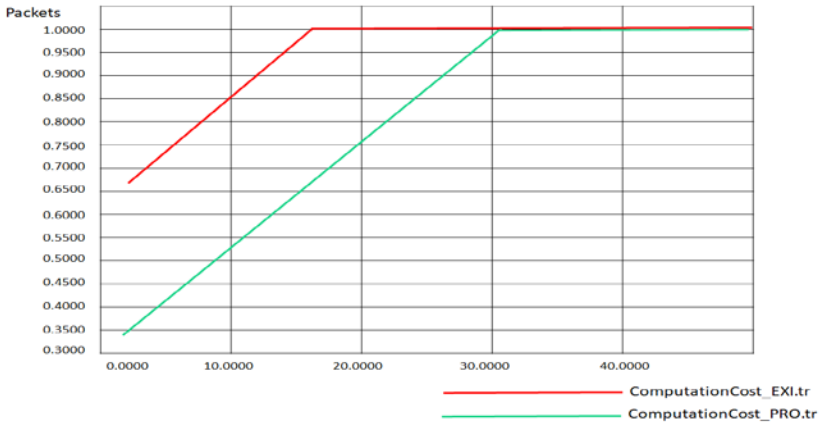


Fig.6. Computation cost of the proposed protocol vs. existing AODV protocol

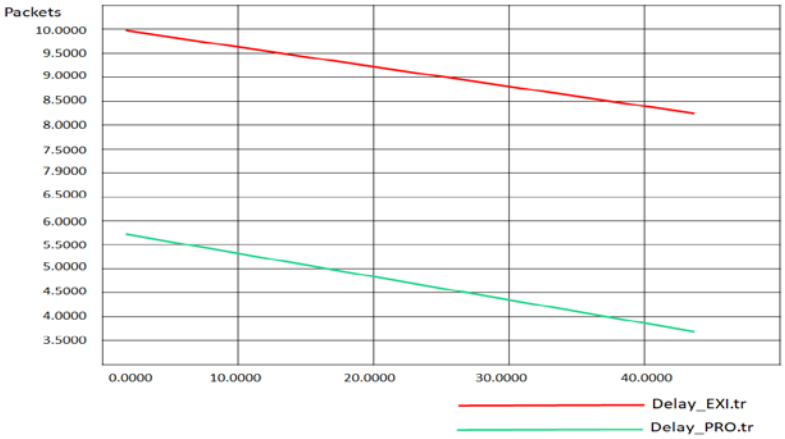


Fig.7. Delay performance of the proposed protocol vs. existing AODV protocol

It is observed that the proposed protocol demonstrates similar throughput and computation cost and has lower delay performance compared to the native AODV routing protocol [32]. It proves that the proposed protocol doesn't suffer from any delay, computational overheads or throughput issues in spite of added features. From the qualitative and quantitative results achieved, the robustness of the protocol is proven for IoT-integrated WSN communication.

## 7. Conclusion

A lightweight Rabin-assisted three-factor mutual authentication protocol is proposed. It is observed that the proposed protocol is resistant to most of the commonly known attacks in wireless sensor networks compared to current research. In addition, it offers additional features in comparison with existing protocols. Under quantitative assessments through simulation and comparing it with native AODV routing protocol, the proposed protocol outperforms the baseline protocol in terms of higher throughput, lower computation cost and delay performance. In spite of the additional features implemented, the proposed protocol is lightweight as can be seen from the quantitative assessments.

As a future research direction, the protocol can be extended and adapted to other critical application domains like Smart Transportation Systems (STS), Smart grids, Smart buildings, Smart cities, Smart Drug Delivery System (SDDS), Wireless Medical sensor networks (WMSN) and Nuclear Power Plants (NPP).

## References


[1] S. Hong et al. SNAIL: An IP-based wireless sensor network approach to the Internet of Things, IEEE Wirel. Commun. 2010; vol. 17, no. 6, pp. 34–42, DOI: 10.1109/WMC.2010.5675776.

[2] J. Granjal, E. Monteiro, and J. S. Silva, Security in the integration of low power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Netw. 2015; vol. 24, pp. 264–287, DOI: 10.1016/j.adhoc.2014.08.001.

[3] Z. Sheng et al. A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities, IEEE Wirel. Commun.2013; vol. 20, no. 6, pp. 91–98, DOI: 10.1109/MWC.2013.6704479.

[4] J. Astorga, E. Jacob, N. Toledo, and J. Unzilla, Enhancing secure access to sensor data with user privacy support, Comput. Netw 2014; vol. 64, pp. 159–179, DOI: 10.1016/j.comnet.2014.02.002.

[5] P. Gope and T Hwang, A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks, IEEE Trans. on Indust. Electron. 2016; vol. 63, no. 11, pp. 7124–7132. DOI: 10.1109/TIE.2016.2585081

[6] K. H. M. Wong, Y. Zheng, J. Cao, and S. Wang. A dynamic user authentication scheme for wireless sensor networks. In IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06). 2006; volume 1, pages 244-251.

[7] S. Shin, Two-Factor Authentication LRP-AKE, Revisited 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 990-995

[8] Wang, P., et al., Revisiting Anonymous Two-Factor Authentication Schemes for IoT-Enabled De- vices in Cloud Computing Environments. Security and Communication Networks, 2019. 2019: p. 2516963.

[9] M. Sarvabhatla and C. S. Vorugunti, A Secure Biometric-Based User Authentication Scheme for Heterogeneous WSN, 2014 Fourth International Conference of Emerging Applications of Information Technology, Kolkata, 2014, pp. 367-372

[10] Challa, S., Das, A. K., Kumari, S., Odelu, V., Wu, F., and Li, X. (2016) Provably secure three-factor authentication and key agreement scheme for session initiation protocol. Security Comm. Networks, 9: 5412– 5431.

[11] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila and B. Stiller, Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications in IEEE Access, vol. 3, pp. 1503-1511, 2015

[12] He, Debiao, Jianhua Chen, and Yitao Chen. A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. Security and Communication Networks 5.12 2012 1423-1429.

[13] Mohsen Salehi, Jamal Karimian," A Trust-based Security Approach in Hierarchical Wireless Sensor Networks", International Journal of Wireless and Microwave Technologies, Vol.7, No.6, pp. 58-67, 2017.

[14] Tabassum Ara, M Prabhakar, Multifactor Authentication and Key Management Protocol for WSN-assisted IoT Communication, Journal of Telecommunications and Information Technology 2019, vol-3 17-26

[15] A. K. Das. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. Peer-to-Peer Networking and Applications. 2016; 9(1):223-244.

[16] R. Amin et al., Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks, Comput. Netw. 2016; vol. 101, no. C, pp. 42–62. DOI: 10.1016/j.comnet.2016.01.006.

[17] H. L. Yeh et al., A secured authentication protocol for wireless sensor networks using elliptic curves cryptography, Sensors. 2011; vol. 11, no. 5, pp. 4767–4779, DOI: 10.3390/s110504767.

[18] K. Xue, C. Ma, P. Hong, and R. Ding, A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks, J. of Netw. and Comp. Appl. 2013; vol. 36, no. 1, pp. 316–323. DOI: 10.1016/j.jnca.2012.05.010.

[19] Q. Jiang, J. Ma, X. Lu, and Y. Tian, An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks, Peer-Peer Netw. Appl. 2014; vol. 8, no. 6, pp. 1070-1081.

[20] R. Roman and J. Lopez, Integrating wireless sensor networks and the Internet: A security analysis, Internet Res. 2009; vol. 19, no. 2, pp. 246–259. DOI: 10.1108/10662240910952373.

[21] S. Zahra et al., Fog computing over IoT: A Secure deployment and formal verification, IEEE Access, 2017; vol. 5, pp. 27132–27144, DOI: 10.1109/ACCESS.2017.2766180.

[22] H. Zheng, J. Wu, B. Wang, and J. Chen, Modified cipher text-policy attribute-based encryption scheme with efficient

revocation for PHR system, Mathem. Problems in Engin. 2017; article ID 6808190, pp. 1–10, DOI: 10.1155/2017/6808190.
[23] S. Wang, D. Zhao, and Y. Zhang, Searchable attribute-based encryption scheme with attribute revocation in cloud storage, PLoS ONE. 2017; vol. 12, no. 8. DOI: 10.1371/journal.pone.0183459.
[24] H. Lei et al., Performance analysis of physical layer security over generalized-K fading channels using a mixture gamma distribution, in IEEE Commun. Lett. 2016; vol. 20, no. 2, pp. 408–411. DOI: 10.1109/LCOMM.2015.2504580.
[25] J. Lai, R. Deng, C. Guan, and J. Weng, Attribute-based encryption with verifiable outsourced decryption, IEEE Trans. Inform. Forensics and Secure. 2013; vol. 8, no. 8, pp. 1343–1354, DOI: 10.1109/TIFS.2013.227184
[26] M. Elhoseny et al., Secure medical data transmission model for IoT based healthcare systems, IEEE Access. 2018; vol. 6, pp. 20596–20608. DOI: 10.1109/ACCESS.2018.2817615.
[27] O. Ruan, J. Chen, and M. Zhang, Provably leakage-resilient password-based authenticated key exchange in the standard model, IEEE Access. 2017; vol. 5, pp. 26832–26841. DOI: 10.1109/ACCESS.2017.2776160.
[28] R. Amin and G. P. Biswas, A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks, Ad Hoc Netw. 2016; vol. 36, pp. 58–80. DOI: 10.1016/j.adhoc.2015.05.020.
[29] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, and J. Yearwood, Protection of privacy in biometric data", IEEE Access. 2016; vol. 4, pp. 880–892. DOI; 10.1109/ACCESS.2016.2535120.
[30] T. D. Pramila-Bai, S. A. Rabara, and A. V. Jerald, Elliptic curve cryptography based security framework for Internet of Things and cloud computing, Int. J. of Comp. Sci. & Technol. 2015; vol. 6, no. 3, pp. 223–229.
[31] Yasir Arfat, Riaz Ahmed Shaikh,"A Survey on Secure Routing Protocols in Wireless Sensor Networks", International Journal of Wireless and Microwave Technologies, Vol.6, No.3, pp.9-19, 2016.
[32] Benamar KADRI, Mohammed FEHAM, Abdellah MHAMMED,"Architecture Aware Key Management Scheme for Wireless Sensor Networks", International Journal of Information Technology and Computer Science, vol.4, no.12, pp.50-59, 2012.

## Authors' Profiles

**Javeria Ambareen** is a dedicated security expert. She has almost a decade of experience in academia and industry. She holds an M.Tech degree and is currently pursuing her PhD at REVA University in Bengaluru, India. IoT security, application security, automation, and machine learning are some of her main interests.

**Prabhakar M** is currently an Associate Professor at REVA University's School of Computing and Information Technology in Bengaluru, India, with 21 years of teaching experience. Anna University in Chennai awarded him M.Sc. and Ph.D. degrees in Computer Engineering. ADHOC networks and cybersecurity are two of his research interests.

**How to cite this paper:** Javeria Ambareen, Prabhakar M, "Secured Wireless Sensor Network Protocol using Rabin-assisted Multifactor Authentication", International Journal of Computer Network and Information Security(IJCNIS), Vol.14, No.4, pp.60-74, 2022. DOI:10.5815/ijcnis.2022.04.05