High-Confidence Computing 4 (2024) 100178

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Research article

# Anonymity preserving lightweight authentication protocol for resource-limited wireless sensor networks

Vincent Omollo Nyangaresi [a,*], Ganesh Keshaorao Yenurkar [b]

[a] Jaramogi Oginga Odinga University of Science & Technology, Bondo, Kenya
[b] Yeshwantrao Chavan College of Engineering, Wanadongri 441110, India

ARTICLE INFO

*Article history:*
Received 14 May 2023
Revised 14 September 2023
Accepted 30 October 2023
Available online 24 November 2023

*Keywords:*
Anonymity
Authentication
Formal security
Lightweight
Privacy
Sensor
WSN

ABSTRACT

Wireless sensor networks have been deployed in areas such as healthcare, military, transportation and home automation to collect data and forward it to remote users for further processing. Since open wireless communication channels are utilized for data transmissions, the exchanged messages are vulnerable to various threats such as eavesdropping and message falsifications. Therefore, many security solutions have been introduced to address these challenges. However, the resource-constrained nature of the sensor nodes makes it inefficient to deploy the conventional security schemes which require long keys for improved security. Therefore, lightweight authentication protocols have been presented. Unfortunately, majority of these schemes are still insecure while others incur relatively higher energy, computation, communication and storage complexities. In this paper, a protocol that deploys only lightweight one-way hashing and exclusive OR operations is presented. Its formal security analysis using Real-or Random (ROR) model demonstrates its capability to uphold the security of the derived session keys. In addition, its semantic security evaluation shows that it offers user privacy, anonymity, untraceability, authentication, session key agreement and key secrecy. Moreover, it is shown to resist attacks such as side-channeling, physical capture, eavesdropping, offline guessing, spoofing, password loss, session key disclosure, forgery and impersonations. In terms of performance, it has relatively lower communication overheads and improves the computation costs and supported security characteristics by 31.56% and 33.33% respectively.



## 1. Introduction

Wireless Sensor Networks (WSNs) consist of sensor nodes that are characterized by limited energy, memory, transmission range, communication and processing power [1]. These sensors are utilized to measure environmental physical conditions in fields such as the industry, health, military, commerce, transportation and home automation. As explained in [2,3], WSNs are a sub-set of the Internet of Things (IoT) and exhibit dynamic topology with self-organizing characteristics. The WSN deployment for data collection in hostile and unattended environment such as military surveillance renders information processing more intelligent and efficient [4]. Normally, the collected data from the sensors is sent to the gateway nodes over wireless channels, which then forward it to the remote users for further processing. In spite of the numerous benefits of deploying WSNs, they face a myriad of privacy and security threats. This is attributed to their operation in hash and unattended environments, and message exchanges over the open wireless channels. Therefore, WSNs are exposed to threats such as privacy disclosure, eavesdropping, as well as message interception and tampering [1,5,6]. In addition, authors in [7] have identified message replays, eavesdropping and active intrusion as being serious issues in WSNs. Moreover, deployment in hash locations has been identified in [8,9] as challenging issue that exposes the sensor nodes to compromise attacks.

Strong mutual authentication can help validate all the received messages and hence help address some of the above security challenges [10]. For instance, it can help prevent illegal message modifications, uphold confidentiality as well as verify the authenticity of the message sources [11]. Effective key management is another approach that can be deployed to secure the communication process in WSNs [12]. As explained in [13], ideal key management must satisfy the goals of flexibility, security and efficiency. User authentication must be executed between the sensor nodes and all users before these session keys can be negotiated. Afterwards, these session keys are utilized to encrypt all sensitive data so as to prevent unauthorized access [5]. Unfortunately, the resource-constrained nature of the sensor nodes impedes the deployment of long keys for enhanced security [4,14]. This is because long keys result in high computation, communication and storage costs. In addition, some deficiencies have been noted

* Corresponding author.
*E-mail address:* vnyangaresi@jooust.ac.ke (V.O. Nyangaresi).

https://doi.org/10.1016/j.hcc.2023.100178

in majority of the conventional authentication schemes, exposing the sensor nodes to attacks such as spoofing [1]. It is evident that authentication in WSNs can allow the node to verify whether data have been sent from authorized sources and protect the original data from changes. However, there are some security deficiencies in most of the existing authentication protocols, such as ID spoofing attacks [1]. There is therefore need to develop a truly lightweight security protocol that can protect the sensitive data exchanged in WSNs. The proposed protocol uses lightweight cryptographic operations such as one-way hashing and XOR that offers the required efficiency. In addition, it deploys pseudo-identities instead of the real identities of the communicating entities so as to preserve their privacy. Moreover, random nonces are incorporated in the derivation of the session keys so as to preserve key secrecy.

### 1.1. Motivation

The wireless sensor networks face numerous security and privacy challenges due to the message exchanges over the open public channels. Therefore, these networks are vulnerable to attacks such as node capture, message falsification, tampering and eavesdropping. These can lead to user privacy disclosure, forgery, session hijackings among other threats. In addition, the sensor nodes are resource-limited, implying that conventional authentication schemes requiring large key sizes for improved security can incur heavy computation, energy, communication and storage overheads. Therefore, all these techniques are unsuitable for this sensor environment. Although many protocols have been developed, the focus is normally on either security or performance but not both. Consequently, most of the presented schemes are either vulnerable to attacks or have high computation, storage, communication and energy complexities. Since these sensors are deployed in sensitive environments such as military surveillance, any successful compromise can have serious repercussions. Therefore, a suitable authentication protocol needs to be provably secure and lightweight.

### 1.2. Research contributions

In the face of the security and performance issues in Section 1 above, this paper makes the following contributions:

- A protocol that incorporates pseudo-identities is developed to provide privacy, anonymity and untraceability of the users and sensor nodes.
- Random nonces are included in all the derived session keys to make them one-time. This helps preserve backward and forward key secrecy.
- Lightweight cryptographic operations such as exclusive Or (XOR) and one-way hashing are deployed during mutual authentications. This renders our protocol efficient for the resource-limited sensor nodes.
- Formal security analysis is executed using the Real-or Random (ROR) model. This demonstrates that the proposed protocol upholds the security of the session keys derived by the user, gateway node and sensor node.
- Extensive semantic security analysis is carried out, which shows that the proposed protocol is resilient against common wireless sensor network attacks such as packet replays, KSSTI, session key disclosure, forgery, impersonation, MitM, side-channeling, spoofing, physical capture, password loss, eavesdropping, DoS and offline guessing.

The rest of this article is structured as follows: Section 2 describes related work while Section 3 presents the proposed protocol. On the other hand, Section 4 details the security evaluation of our scheme while Section 5 presents its performance evaluation. Finally, Section 6 concludes the paper and gives some future research directions.

## 2. Related work

Many security solutions have been developed over the recent past to secure the sensor nodes from attacks. For ease of understanding, these existing works have been categorized based on the most common cryptographic operations as well as the number of factors involved during the authentication process. In this regard, password, elliptic curve, blockchain, smartcards, Public Key Cryptography (PKC), Media Access Control (MAC), bilinear pairing, one-way hashing, Chebyshev chaotic, two-factor, and three-factor authentication based schemes were found to be popular and common. However, most of these schemes have some challenges which limit their applicability in sensor networks as discussed below.

Based on passwords, authentication protocols have been developed in [15,16]. However, these schemes are susceptible to offline password guessing attacks [11]. To provide forward key secrecy and address stolen verifier attacks, an elliptic curve based scheme is presented in [17]. Unfortunately, this protocol cannot withstand session-specific random number leakage and privileged insider attacks. In addition, it cannot offer forward key secrecy [18]. Although the protocol in [19] can address some of these challenges, the deployed blockchain technology results in high storage and computation overheads [20]. To curb this, two-factor authentication schemes are developed in [21,22], based on smartcards and passwords. However, this makes them vulnerable to smartcard loss and offline password guessing attacks. To offer untraceability and user anonymity, an authentication scheme is presented in [23]. Unfortunately, this protocol is susceptible to Denial of Service (DoS) attacks and cannot preserve forward secrecy [24]. Similarly, the scheme in [25] cannot attain forward secrecy, unlinkability and user anonymity [26]. In addition, it is vulnerable to impersonation and offline password guessing attacks. Public Key Cryptography (PKC) plays critical roles in securing the WSN communication process. As such, PKC based protocols have been developed in [27,28]. Here, pairs of public and private keys are utilized. Unfortunately, PKC requires long key sizes to offer high levels of security protection, which renders it unsuitable for sensor nodes [29]. To preserve data integrity and availability, a Media Access Control (MAC) based authentication model is presented in [30]. However, it is difficult to preserve non-repudiation in MAC addresses [1]. On the other hand, the scheme in [31] fails to verify the password correctness and can potentially result in high computation overheads at the gateway node [5]. Similarly, the Elliptic Curve Cryptography (ECC) based protocols in [32,33] have high communication overheads [4]. Another ECC based scheme is developed in [34] for security enhancement in wireless sensor networks. Unfortunately, this technique is prone to offline password guessing, replay and sensor node capture attacks. In addition, it does not uphold unlinkability, session key secrecy, anonymity and perfect forward secrecy [11].

To provide user authentication in healthcare-based wireless sensor networks, a security scheme is presented in [35]. However, the deployed smartcard renders it vulnerable to smartcard loss attacks. In addition, its bilinear pairing operations during data verification lead to high computation overheads [36]. Therefore, an enhanced scheme based on ECC is presented in [37]. However, the authors in [38] analyzed this protocol and found it vulnerable to information disclosure attacks and cannot achieve strong key security. As such, they proposed an improved scheme that solved these challenges. Unfortunately, this scheme is susceptible to session key exposure and masquerade attacks [39]. In addition, it fails to attain untraceability and anonymity. Therefore, an enhanced three-factor authentication technique is introduced in [40]. However, this approach is still prone to

de-synchronization and stolen-verifier attacks. It also fails to provide perfect forward key secrecy [41]. Therefore, an ECC-based anonymous security solution is developed in [41]. However, user identities can be recovered upon adversarial capture of the sensor nodes in both [40,41]. In addition, the scheme in [41] has unrealistic network model in which the sensor node communicates directly with the remote users devoid of the gateway node. As such, the sensor node battery can be quickly drained. To secure the communication between the sensors and users, hash-based authenticated protocols are developed in [42,43]. However, the scheme in [42] is vulnerable to information leakage attack and cannot achieve anonymity [5]. Similarly, the protocol in [43] is resilient against offline password guessing attack but fails to protect against information leakage attack. This is because of the dependency among the random numbers deployed for session key derivation. On the other hand, a trust-based authentication model is introduced in [44]. Unfortunately, the inclusion of timestamps during trust score computation renders it vulnerable to de-synchronization attacks [45]. Similarly, the protocol in [46] is susceptible to de-synchronization, privileged insider and user impersonation attacks. It also fails to attain untraceability and anonymity [5]. Based on these weaknesses, a three-factor user authentication scheme is presented in [5]. Although this scheme is resilient against user identity and password offline guessing attacks upon smart card loss, it has not been evaluated against attacks such as side-channeling, forgery and spoofing. Based on ECC, authentication protocols have been presented in [47,48]. However, these two protocols cannot provide three-factor security [38]. In addition, the scheme in [47] is vulnerable to replay and counterfeit attacks, while the protocol in [48] is susceptible to information leakage attack [38]. On the other hand, the Rabin PKC based scheme in [49] incurs heavy computation overheads that can drain sensor battery [50]. Two schemes based on Chebyshev chaotic mapping have been presented in [51,52] to enhance efficiency and security in WSNs. However, these two methods cannot withstand gateway node impersonation attacks [5]. In addition, the protocol in [52] fails to protect against privileged insider and impersonation attacks [5].

In spite of the high number of security solutions developed over the recent past, it is evident that the attainment of perfect security in the face of limited resources at the sensor node still remains challenging. In this environment, various options such as passwords, cryptographic protocols, IDs, MAC addresses and certificates have been deployed during authentication [53]. However, each of these methods has some challenges. For example, ID-based schemes are susceptible to spoofing attacks while the storage complexity for certificate-based techniques is high for sensor nodes. On the other hand, non-repudiation cannot be assured for password, ID and MAC address based schemes [1]. Regarding cryptographic protocols, their security is hinged on the strength of the underlying encryption algorithms. On the other hand, data increase in blockchain networks result in increased number of blocks and the surging memory requirements [54]. In addition, any increase in data leads to surging number of transactions and energy requirements for validating the blocks as well as addition of new blocks to the chain [55]. The proposed protocol is shown to incur the least computation costs while at the same time supporting the highest number of security and privacy features. It therefore solves some of the performance, privacy and security challenges inherent in most of the existing works.

## 3. The proposed protocol

The key concepts deployed in the proposed protocol are first presented in this section, followed by the actual protocol. This includes the mathematical formulations, threat model, security requirements, key design principles, network model and the various phases of the proposed scheme.

### 3.1. Mathematical preliminaries

In this sub-section, some mathematical formulations for the one-way hashing operations are presented. This include its collision-resistant property as well as its output format.

**Definition 1.** Suppose that $s$ is a variable length input string and $h(.)$ is a one-way hashing function such that $s \in \{0,1\}^*$. Considering h: $\{0,1\}^* \rightarrow \{0,1\}^L$ as a deterministic function, then when $s$ is supplied as input to $h(.)$, an $L$ bits fixed length output string $h(s)$ is produced.

**Definition 2.** Let $\bar{A}$ be an adversary interested in finding a collision for $h(.)$ in time $T$. Then, $Adv_{\bar{A}}^{h(.)}(T)$ denotes the advantage that $\bar{A}$ has for this collision. In addition, $Adv_{\bar{A}}^{h(.)}(T) = P_r\left[(\sigma_1, \sigma_2)\right] \in_R^{\bar{A}}: \sigma_1 \neq \sigma_2, h(\sigma_1) = h(\sigma_2)$.

**Definition 3.** Consider an $(\omega, T)$ − adversary attempting to compromise the one-way hashing function's collision resistance. Then at most run time $T$, $Adv_{\bar{A}}^{h(.)}(T) \leq \omega$.

### 3.2. Threat model

In this protocol, an adversary is assumed to have all the capabilities advocated in the Dolev–Yao (D–Y) and Canetti and Krawczyk (C–K) threat model. Basically, the attacker is thought to have capabilities of intercepting, eavesdropping, altering, deleting and replaying messages exchanged over the public channels. In addition, the adversary can have access to secret security tokens deployed in the authentication procedures. These may comprise of tokens stored in sensor devices, long-term private keys as well as transient parameters used to derive the session keys.

### 3.3. Security requirements

In the face of the attacker with the capabilities under the D–Y and C–K threat models, an ideal authentication protocol should have the following requirements to ensure secure communication process among the user, sensor and gateway node.

*User privacy:* After successful adversarial capture of the exchanged messages between the remote user and the sensor nodes, it should be impossible to discern personally identifiable information of the user from these messages.

*Anonymity:* It should be cumbersome for the attackers to establish the actual identity of the users and sensors based on any captured messages.

*Untraceability:* Upon successful capture of the transmitted messages, adversaries should be incapable of tracing these messages to particular users and sensor nodes.

*Authentication:* All the sources of the transmitted messages must be verified at the receiver end.

*Session key agreement:* After the completion of the communication entities verification procedures, session keys need to be negotiated, which will be utilized to encipher all the exchanged data.

*Confidentiality and integrity:* During the message exchanges over the wireless sensor networks, it should be cumbersome for the attackers to discern the nature of transmitted data. In addition, it should be difficult for the adversaries to modify the exchanged messages over the public channels.

*Key secrecy:* Any exposure of the long term secret keys deployed to compute the session key should not enable the attacker to

discern the session key for the previous communication process. In addition, these secret keys should not facilitate adversarial derivation of the session key for the subsequent communication process.

*Robustness against attacks:* The WSN authentication protocol should be capable of resisting common attack vectors such as side-channeling, physical capture, eavesdropping, offline guessing, spoofing, password loss, session key disclosure, forgery, impersonation, Man-in-the-Middle (MitM), privileged insider, Known session-specific temporary information (KSSTI), Denial of Sleep (DoS) and replays.

*Password revocation:* To prevent authorized access, user passwords should be revoked upon adversarial compromise.

### 3.4. Key design principles

To provide superior security, privacy and performance features, the proposed protocol adheres to the following design principles.

*Untraceability and anonymity of the sensor nodes:* During mutual authentication, we use the pseudo-identities of the sensor nodes instead of their real identities. After every successful session, these pseudo-identities are refreshed. This renders then session-specific and hence adversaries are unable to discern and track the communicating sensors.

*Perfect backward and forward key secrecy:* The derived session keys incorporate random nonces which are encapsulated in collision-resistant one-way hashing function. Therefore, it is computationally infeasible to reverse this function and obtain the nonces for subsequent and past session key derivation. The frequent refreshing of these random nonces implies that the derived session keys are stochastic hence cannot be easily guessed.

*Resilience against spoofing and denial of sleep attacks:* During the mutual authentication procedures, the user identities, sensor pseudo-identity, random nonces and other intermediary keying parameters are verified at the receiver end. This helps in the detection of the frequent re-transmissions of large quantities of old but valid messages using the spoofed user and sensors.

*Impersonation attack prevention:* In the proposed protocol, only short term and transient parameters are utilized during session keys derivation. This effectively thwarts the capture of long term keying parameters that can be utilized by the adversaries to masquerade as legitimate networks entities.

*Resilience against password loss attacks*: In our scheme, the four-step password update phase described in Section 3.10 is invoked upon password compromise or loss. As such, an adversary with the old password is unable to authenticate successfully. Any adversarial attempt to refresh the password will fail since numerous keying materials are required in this process, all of which are unavailable to the attacker.

*Robustness against KSSTI and replay attacks*: In the proposed protocol, the session keys are independently derived at each of the communicating entities. During these derivations, user passwords, and short term secrets such as nonces and other ephemerals are encapsulated before being hashed. Therefore, even if all the nonces are captured, the attacker still needs to reverse the one-way hashing function to obtain other keying materials. Since this is computationally infeasible, these two attacks are prevented.

*Resilience against eavesdropping and session key disclosure attacks:* The session keys are independently derived by each of the communicating entities, using random nonces among other ephemerals. During mutual authentication, these keying materials are never exchanged in plaintext over public channels. This makes it impossible for the adversary to eavesdrop these materials. The hashing of the generated session key makes it difficult for the attackers to decipher the keying parameters.

*Robustness against privileged insider attacks*: The aim of this attack is to obtain secret parameters such as user real identity, password and shared common keys. In our scheme, all these parameters are encapsulated in other keying materials before being hashed. Therefore, it is difficult for the adversaries to easily obtain these parameters.

*User privacy preservation:* During the registration process, pseudo-identities are generated for each user. Later on, these pseudo-identities are incorporated in the exchanged messages when performing mutual authentication and key negotiation. The frequent refreshing of user pseudo-identity after each session makes it difficult for the attacker to identify and track users.

*Resilience against MitM and offline guessing attacks:* In our protocol, the attacker cannot correctly guess or compute the ephemerals used to derive the session keys. This is because the keying materials are either stored in memory, while others are independently derived at the communicating entities and never shared in plaintext over public channels. Therefore, they are unavailable for the attackers and hence MitM and offline guessing attacks are prevented.

*Forgery attacks prevention:* Our protocol incorporates random nonces as well as some private keys and identities of the communicating entities in the exchanged messages. Since all these parameters are never exchanged in plaintext over the public channels, the attacker is unable to obtain them. As such, forgery of exchanged messages fails.

*Resilience against physical capture and side-channeling attacks:* Although the adversary may succeed in physically capturing and extracting memory-resident security parameters, the derivation of other keying ephemerals still fails. This is because the attacker does not have access to security values such as user password, identities and random nonces. In addition, the exchanged messages cannot be forged since the adversary requires additional parameters apart from the ones obtained from the devices' memory.

*Confidentiality and integrity are upheld:* To preserve confidentiality, all exchanged messages are enciphered before being transmitted over the public channels. On the other hand, the integrity of all the session keys are preserved by making it difficult to derive past and future keys as explained under '*Perfect backward and forward key secrecy*' section above.

*Support for scalability:* In our scheme, the communicating entities do not require any verifier tables during the mutual authentication and key agreement procedures. As such, additional users and sensor nodes can join and leave the network flawlessly devoid of adverse effects on its performance.

### 3.5. Network model

The main components in our protocol include the user $U_i$, mobile device $MD_i$, gateway node ($GWN_i$) and sensor node ($SN_i$) as shown in Fig. 1. As shown, the user deploys $MD_i$ to interact with $SN_i$ through the $GWN_i$. After successful registration over secured channels, all other messages are exchanged over the open public channels.

Table 1 presents the symbols utilized throughout this paper. The main procedures involved in this scheme include system setup, registration phase, authentication, key agreement, and password change phases.

The sub-sections below describes of these phases in some greater details.

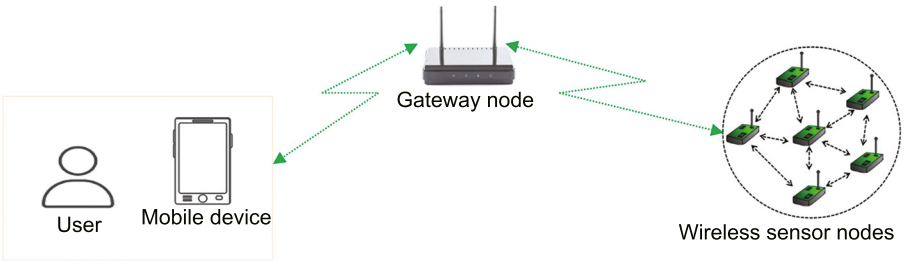


**Fig. 1.** Network model.

**Table 1**
Symbol descriptions.

| Symbol | Description |
|---|---|
| $U_i$, $GWN_i$, $SN_i$ | *ith* user, gateway node and sensor node |
| $GID_i$ | Gateway node *i* unique identity |
| $SK_G$ | Gateway node private key |
| $SID_i$ | Sensor node *i* unique identity |
| $NID_j$ | Network *j*'s unique identity |
| $GS_C$ | Shared common key between the $GWN_i$ and $SN_i$ |
| $R_i$ | Random nonce *i* |
| $PID_K$ | Sensor node *i* pseudo-identity |
| $\Psi_s$, $\Psi_G$, $\psi_U$ | Common session key derived at $SN_i$, $GWN_i$&$U_i$ |
| $UID_i$, $PW_U$ | User's unique identity and password |
| $PID_i$ | User's pseudo-identity |
| $MD_i$ | User's mobile device |
| $\|$ | Concatenation operation |
| $\oplus$ | XOR operation |

### 3.6. System setup phase

In the proposed protocol, all the message exchanges between the user ($U_i$) and the wireless sensors ($SN_i$) are via the gateway node ($GWN_i$). The following two steps are executed during the initialization phase.

**Step 1** The $GWN_i$ generates its unique identity $GID_i$. Next, it chooses some one way hashing function $h$ (.) before generating its private key $SK_G$ as shown in Fig. 2. The one-way hashing so selected is assumed to be collision-resistant.

**Step 2:** The $GWN_i$ generates some network identities $NID_j$ to uniquely recognize sensor nodes within the same network.

### 3.7. Sensor node registration

All the sensor nodes must be registered at the gateway node before being deployed for data collection in their particular application domains. The following two steps are followed to accomplish this process.

**Step 1:** Every wireless sensor node $SN_i$ generates its unique identity $SID_i$. Next, it derives $GS_C = h(SID_i \| SK_G \| NID_j)$, which serves as the common shared key between itself and the gateway node $GWN_i$. It then sends parameter set $\{SID_i, GS_C\}$ to the $GWN_i$ over some secured channels as shown in Fig. 2.

**Step 2:** On receiving these values, the $GWN_i$ generates random nonces $R_1$ and $R_2$. Next, it generates pseudo-identity $PID_K$ for each $SN_i$. It then sends parameter set $\{SID_i, GS_C, GID_i, R_1, R_2, PID_K\}$ to the $SN_i$ over secure channels. Next, the $GWN_i$ stores parameters $\{SID_i, PID_K, NID_j, R_1, h(R_2)\}$ in its database. Finally, the wireless sensor nodes are installed in their application domain.

### 3.8. User registration

In this protocol, user $U_i$ deploys a smart mobile device $MD_i$ to communicate with the gateway node $GWN_i$. As such, the $MD_i$ generates and stores security tokens on behalf of the user. The registration process is accomplished through the following four steps.

**Step 1:** The user $U_i$ chooses unique identity $UID_i$ and strong password $PW_U$. Next, $MD_i$ selects random nonce $R_3$ that is deployed to derive parameter $A_1 = h(PW_U \| R_3)$. This is followed by the transmission of parameter set $\{UID_i, PW_U\}$ to the $GWN_i$ through some secure channels as shown in Fig. 2.

**Step 2:** On receiving $\{UID_i, PW_U\}$, the $GWN_i$ checks if the received $UID_i$ is already in its database. Basically, the registration of this new user is rejected if this identity is already in $GWN_i$'s repository. Otherwise, the $GWN_i$ generates pseudo-identity $PID_i$ for this particular user.

**Step 3:** The $GWN_i$ generates random nonce $R_4$ for the $U_i$. This is followed by the derivation of tokens $A_2 = h(PID_i \| R_4 \| GID_i \| SK_G) \oplus A_1$ and $A_3 = h(UID_i \| SK_G) \oplus h(UID_i \| A_1)$. Next, it stores parameter set $\{PID_i, R_4, UID_i, A_1\}$ in its database. It finally sends parameter set $\{A_2, A_3, PID_i, GID_i\}$ to user $MD_i$ over some secured channels.

**Step 4:** Upon receiving the above parameter set, $MD_i$ computes $A_4 = h(UID_i \| PW_U) \oplus R_3$. Finally, parameter set $\{A_2, A_3, A_4, PID_i, GID_i\}$ is stored in $U_i$'s mobile device $MD_i$.

### 3.9. Mutual authentication and key agreement phase

This phase is triggered whenever the user wants to have some access to the wireless sensor data. Here, the communication among the $U_i$, $GWN_i$ and $SN_i$ is executed over insecure public channels. For enhanced security and privacy, the following 10 procedures are carried out.

**Step 1:** The user $U_i$ inputs unique identity $UID_i$ and password $PW_U$ to the $MD_i$. Next, the $MD_i$ derives $R_3 = A_4 \oplus h(UID_i \| PW_U)$ and $A_1 = h(PW_U \| R_3)$. This is followed by the generation of random nonce $R_5$ before selecting this particular $SN_i$'s pseudo-identity $SID_i$.

**Step 2:** The $MD_i$ derives $A_5 = A_2 \oplus A_1$, $B_1 = A_5 \oplus A_1 \oplus R_5$, $B_2 = SID_i \oplus h(UID_i \| R_5)$, $B_3 = h(PID_i \| GID_i \| SID_i \| A_5 \| UID_i \| R_5)$. At the end, it constructs message $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$ that it forwards to the $GWN_i$ over public channels as shown in Fig. 3.

**Step 3:** Upon receiving message $MAK_1$ from the $MD_i$, the $GWN_i$ extracts parameters $GID_i$ and $PID_i$. It then retrieves the corresponding $UID_i$, $R_4$ and $A_1$ from its repository. If these values do not exist in the database, the request is flagged as malicious and the authentication session is stopped. Otherwise, the $GWN_i$ derives $A_5 = h(PID_i \| R_4 \| GID_i \| SK_G)$ and $R_5 = B_1 \oplus A_5 \oplus A_1$.

**Step 4:** The $GWN_i$ generates random nonces $R_6$ and $R_2^*$. This is followed by the computation of $SID_i = B_2 \oplus h(UID_i \| R_5)$ and retrieval of nonce $R_1$ from the database. Next, it generates new sensor node pseudo-identity $PID_K^*$ for the sensor node.

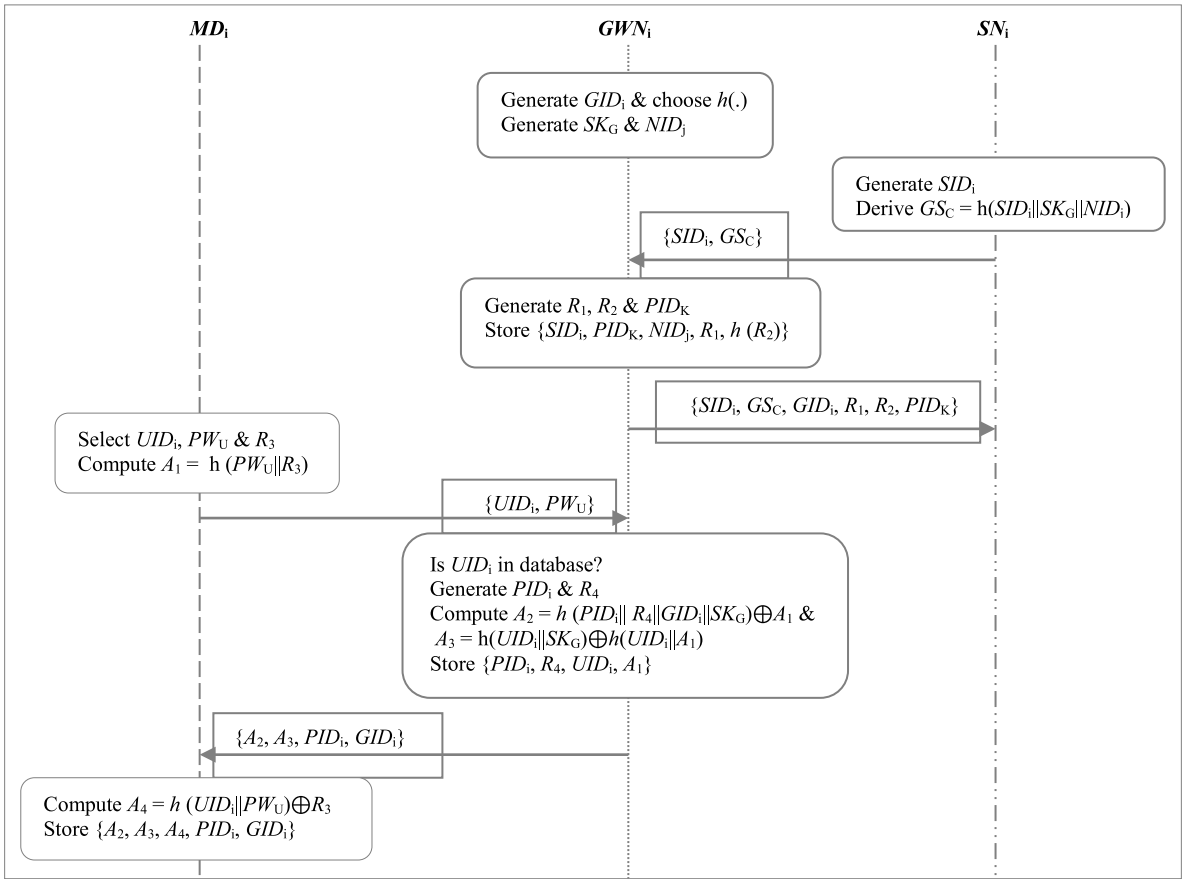


**Fig. 2.** System setup and registration phases.

**Step 5:** The gateway node $GWN_i$ calculates parameters $GS_C = h\ (SID_i \parallel SK_G \parallel NID_j)$, $B_4 = h\ (GS_C \parallel GID_i)$, $B_5 = (R_5 \oplus A_1 \oplus B_4 \oplus R_1$, $C_1 = R_6 \oplus B_4 \oplus SID_i \oplus R_1$, $C_2 = PID_K^* \oplus R_6 \oplus R_1$, $C_3 = h\ (R_6 \parallel R_1 \parallel B_4) \oplus R_2^*$ and $C_4 = h\ (PID_K \parallel C_2 \parallel C_3 \parallel GS_C \parallel R_5 \oplus A_1 \parallel R_6)$. Finally it composes message $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$ that it forwards to wireless sensor node $SN_i$.

**Step 6:** Upon getting message $MAK_2$ from the $GWN_i$, the $SN_i$ validates $PID_K$ in this message against its equivalence in its memory. If this verification fails, the authentication session is stopped. Otherwise, the $SN_i$ derives $B_4 = h\ (GS_C \parallel GID_i)$, $(R_5 \oplus A_1) = B_5 \oplus B_4 \oplus R_1$ and $R_6 = C_1 \oplus B_4 \oplus SID_i \oplus R_1$. Next, it computes $C_4^* = h\ (PID_K \parallel C_2 \parallel C_3 \parallel GS_C \parallel R_5 \oplus A_1 \parallel R_6)$ and validates it against its equivalence $C_4$ in message $MAK_2$. On condition that this verification flops, the session is aborted. Otherwise, it generates nonce $R_7$ that it uses to compute $R_2^* = h\ (R_6 \parallel R_1 \parallel B_4) \oplus C_3$, $PID_K^* = C_2 \oplus R_6 \oplus R_1$ and $C_5 = R_6 \oplus B_4 \oplus R_2$.

**Step 7:** The $SN_i$ stores $PID_K^*$, $R_2^*$ and $R_1^* = h\ (R_1)$ in its memory. This is followed by the computation of the session key $\psi_S = h\ (R_5 \oplus A_1 \parallel R_6 \parallel R_7)$, $D_1 = h\ (GS_C \parallel R_6) \oplus h\ (R_1) \oplus R_7$ and $D_2 = h\ (C_5 \parallel D_1 \parallel \psi_S \parallel SID_i \parallel GID_i \parallel R_7)$. At the end, it constructs message $MAK_3 = \{C_5, D_1, D_2\}$ which is sent to the $GWN_i$ over insecure communication channels.

**Step 8:** On receiving message $MAK_3$ from $SN_i$, the $GWN_i$ retrieves $R_1$ from its database and computes $R_1^{**} = h\ (R_1)$ as well as $R_2^{**} = R_6 \oplus B_4 \oplus C_5$. Thereafter, it confirms whether $h\ (R_2) \stackrel{?}{=} h\ (R_2^{**})$ such that the session is terminated when this validation flops. Otherwise, the $GWN_i$ computes $R_7 = D_1 \oplus h\ (GS_C \parallel R_6) \oplus h\ (R_1^{**})$ and session key $\psi_G = h\ (R_5 \oplus A_1 \parallel R_6 \parallel R_7)$. Next, it computes $D_2^* = h\ (C_5 \parallel D_1 \parallel \psi_G \parallel SID_i \parallel GID_i \parallel R_7)$ and checks if $D_2^* \stackrel{?}{=} D_2$. On condition that these parameters are not identical, the session is terminated. Otherwise, it generates new pseudo-identity $PID_i^*$ for the user $U_i$ before storing $PID_K^*$ and $R_2^*$ in its database.

**Step 9:** The $GWN_i$ substitutes parameter set $\{R_1^{**}, h\ (R_4)\}$ with $\{R_1, R_4\}$. Thereafter, it derives $D_3 = h\ (PID_i^* \parallel h\ (R_4) \parallel GID_i \parallel SK_G \parallel\ ) \oplus h\ (R_5 \parallel A_1)$, $D_4 = h\ (R_5 \parallel UID_i) \oplus R_6$, $D_5 = h\ (R_5 \parallel R_6 \parallel A_1) \oplus R_7$, $E_1 = h\ (h\ (UID_i \parallel SK_G) \parallel R_7) \oplus PID_i^*$ and $E_2 = h\ (\psi_G \parallel UID_i \parallel D_3 \parallel PID_i^*)$. Lastly, it constructs message $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$ that it transmits over to the user $MD_i$.

**Step 10:** Upon receiving message $MAK_4$ from the $GWN_i$, $MD_i$ retrieves nonces $R_6$ and $R_7$ as $R_6 = D_4 \oplus h\ (R_5 \parallel UID_i)$ and $R_7 = D_5 \oplus h\ (R_5 \parallel R_6 \parallel A_1)$. This is followed by the derivation of the $U_i$'s new pseudo-identity $PID_i^* = E_1 \oplus h\ (h(UID_i \parallel SK_G) \parallel R_7)$. Next, the $MD_i$ computes the session key as $\psi_U = h\ (R_5 \oplus A_1 \parallel R_6 \parallel R_7)$. Thereafter, it computes $E_2^* = h\ (\psi_U \parallel UID_i \parallel D_3 \parallel PID_i^*)$ and confirms whether $E_2^* \stackrel{?}{=} E_2$. The session is aborted when these two values are dissimilar. Otherwise, new $A_2$ is re-computed as $A_2^* = D_3 \oplus h\ (R_5 \parallel A_1)$. Finally, parameter set $\{PID_i^*, A_2^*\}$ is stored in $MD_i$'s memory.

### 3.10. Password change phase

This phase is triggered when the user $U_i$ password $PW_U$ is compromised, or when organizational security policy advocates for frequent password refreshing. This is a four-step process as described below.

**Step 1** The user $U_i$ inputs the current unique identity $UID_i$ and password $PW_U$ to the $MD_i$. Next, the $MD_i$ derives $R_3 = A_4 \oplus h\ (UID_i \parallel PW_U)$, $A_1 = h\ (PW_U \parallel R_3)$. This is followed by the retrieval

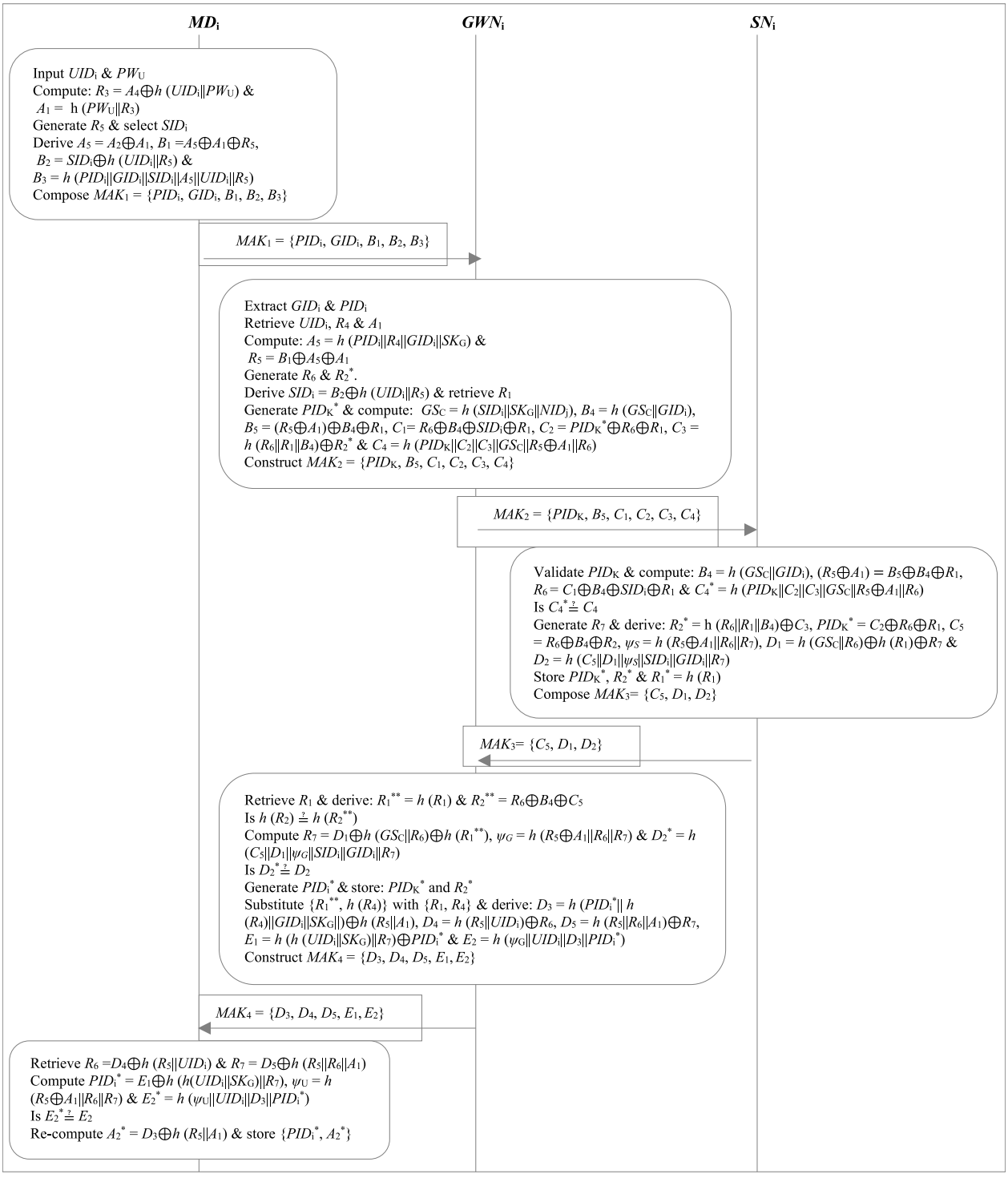


**Fig. 3.** Authentication and key agreement phase.

of random nonce $R_5$ that is utilized to derive $A_5 = A_2 \oplus A_1$, $E_3 = A_5 \oplus R_5$ and $E_4 = UID_i \oplus h(R_5 \| A_5)$.

**Step 2:** The $U_i$ selects new password $PW_U^*$ and pseudo-identity $PID_i^{**}$. Next the $MD_i$ selects new nonce $R_3^*$ that it deploys to compute $A_1^* = h(PW_U^* \| R_3^*)$, $E_5 = A_1^* \oplus E_4$, $F_1 = PID_i^{**} \oplus h(UID_i \| R_5)$ and $F_2 = h(PID_i^{**} \| PID_i \| GID_i \| A_5 \| E_5 \| UID_i \| R_5)$. At last, it constructs message $PC_1 = \{PID_i, GID_i, E_3, E_5, F_1, F_2\}$ that is forwarded to the gateway node $GWN_i$.

**Step 3:** On getting password change request message $PC_1$ from the user $U_i$, the $GWN_i$ confirms whether parameter set $\{GID_i, PID_i\}$ is in its database. If this is not the case, the password change request is denied and an error message is sent to $U_i$. Otherwise, it retrieves $UID_i$ and $R_4$ from its database and derives $A_5 = h(PID_i \| R_4 \| GID_i \| SK_G)$, $R_5 = A_5 \oplus E_3$, $E_4 = UID_i \oplus h(R_5 \| A_5)$, $A_1^* = E_4 \oplus E_5$ and $PID_i^{**} = F_1 \oplus h(UID_i \| R_5)$.

**Step 4:** The $GWN_i$ re-computes $F_2^* = h(PID_i^{**} \| PID_i \| GID_i \| A_5 \| E_5 \| UID_i \| R_5)$ and checks if $F_2^* \stackrel{?}{=} F_2$. The password session is aborted if this verification fails. Otherwise, it stores $PID_i^{**}$ and $A_1^*$ in its database. Next, it derives $F_3 = h(PID_i^{**} \| h(R_4) \| GID_i \| SK_G) \oplus h(R_5 \| UID_i)$ and $F_4 = h(UID_i \| PID_i \| PID_i^{**} \| A_5 \| F_3)$.

**Table 2**
Executed random oracles.

| Query | Description |
|---|---|
| *Send* ($\bar{I^1}$, *MAK*) | Adversary $\bar{A}$ sends message *MAK* to $\bar{I^1}$, and receives response from $\bar{I^1}$. This is a classical active attack. |
| *Execute* ($SN_i$, $U_i$, $GWN_i$) | $\bar{A}$ can eavesdrop messages exchanged among the $SN_i$, $U_i$ and $GWN_i$. It is a typical passive attack. |
| *Corrupt* ($SN_i$) | Adversary $\bar{A}$ can extract all secret tokens stored in $SN_i$'s memory |
| *Corrupt* ($MD_i$) | $\bar{A}$ can extract all secret tokens stored in $MD_i$'s memory |
| *Reveal* ($\bar{I^1}$) | Adversary $\bar{A}$ can discern session key established between $\bar{I^1}$ and its corresponding communicating entity |
| *Test*($\bar{I^1}$) | Attacker $\bar{A}$ can request session key from $\bar{I^1}$, which then probabilistically outputs the outcome of a flipped unbiased coin λ. |

Finally, it constructs message $PC_2 = \{F_3, F_4\}$ and forwards it to the user's $MD_i$.

**Step 5:** After obtaining message $PC_2$ from the $GWN_i$, the $MD_i$ derives $F_4^* = h\ (UID_i \parallel PID_i \parallel PID_i^{**} \parallel A_5 \parallel F_3)$ and checks if $F_4^* \stackrel{?}{=} F_4$. Basically, the password change session is terminated if this verification is unsuccessful. Otherwise, it re-computes $A_2^* = F_3 \oplus h\ (R_5 \parallel UID_i) \oplus A_1^*$, $A_3^* = A_3 \oplus h(UID_i \parallel A_1) \oplus h(UID_i \parallel A_1^*)$ and $A_4^* = h\ (UID_i \parallel PW_U^*) \oplus R_3^*$. At last, it stores parameter set $\{A_2^*, A_3^*, A_4^*, PID_i^{**}\}$ in its memory.

## 4. Security analysis

In this section, the formal and semantic security analyzes of the proposed protocol is provided. The sub-sections below gives some detailed descriptions of these procedures.

### 4.1. Formal security analysis

In this section, the Real-Or-Random (ROR) model is utilized to demonstrate that the proposed protocol offers security to the session keys derived at the $U_i$, $GWN_i$ and $SN_i$. The choice of ROR model is informed by the fact that it has been extensively deployed to formally analyze many authentication protocols. To achieve this, the collision resistant property of $h$ (.) in Section 3.1 above is used. Essentially, the one-way collision-resistant hash function $h$(.) is accessible to all entities including adversary $\bar{A}$. As such, this hash function is modeled as a random oracle, *Hash*. The *Send*, *Execute*, *Reveal*, *Corrupt* and *Text* are other random oracles (queries) that $\bar{A}$ can perform as shown in Table 2 below. Basically, ROR model permits $\bar{A}$ to interconnect with the $i^{th}$ instance $\bar{I^1}$ of an executing party such as $GWN_i$, $U_i$ and $SN_i$.

During this formal analysis, it is assumed that the adversary can guess low entropy passwords in accordance with Zipf's law. Suppose that $\bar{A}$ is interested in deriving the session key negotiated among the $U_i$, $GWN_i$ and $SN_i$. Letting $\dot{Z}$ denote the proposed protocol, the following lemma holds:

**Corollary.** *The advantage that $\bar{A}$ has in executing in polynomial time $_T$ to break $\dot{Z}$'s semantic security is represented as $Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T)$. Taking λ and λ* as the correct and guessed bits respectively, then $Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T) = |\ 2\ Pr[\lambda^* = \lambda] - 1|$ .*

**Theorem 1.** *Let $\bar{A}$ be an attacker executing in $_T$ against $\dot{Z}$, and $Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T)$ be $\bar{A}$'s advantage in computing session key negotiated among the $U_i$, $GWN_i$ and $SN_i$ during the authentication and key agreement (AKA) to break $\dot{Z}$'s semantic security in $_T$. Suppose that $N_H$ is the number of hash queries, $N_S$ is the number of Send queries and $|H|$ is the range space for the one-way hash function h Then:*

$$Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T) \leq \frac{N_H^2}{|H|} + 2z_1 N_S^{z_2}$$

*where $z_1$ and $z_2$ are Zipf's parameters.*

**Proof.** To execute this attestation, three adversarial games $\bar{A}_k^{Game}$, $k = 0, 1, 2$ are defined. Suppose that $S\_\bar{A}_k^{Game}$ denotes the successful guessing of bit $q$ by $\bar{A}$ in game $\bar{A}_k^{Game}$. Then its success probability is represented by $Adv_{\bar{A}}^{Game_k} = P_r[S\_\bar{A}_k^{Game}]$. The three games alluded above are described below.

**$\bar{A}_0^{Game}$:** This is the game that mimics adversarial actual attack against $\dot{Z}$ under the ROR model. Initially, $\bar{A}$ picks bit $c$, and based on the above *Corollary*:

$$Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T) = |2Adv_{\bar{A}}^{Game_0}| \quad (1)$$

**$\bar{A}_1^{Game}$:** In this game, it is assumed that $\bar{A}$ can eavesdrop all the messages exchanged during the AKA procedures. These messages include $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$, $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$, $MAK_3 = \{C_5, D_1, D_2\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$. To accomplish this, $\bar{A}$ executes the *Execute* ($SN_i$, $U_i$, $GWN_i$) query. These three communicating entities negotiate session keys given by $\psi_U = h\ (R_5 \oplus A_1 \parallel R_6 \parallel R_7) = \psi_S = \psi_G$. With the help of *Reveal* () and *Test* () queries, $\bar{A}$ verifies whether the computed session key is the valid one or just a random one. Based on Lemma 7, eavesdropping of all the above exchanged messages cannot help $\bar{A}$ to derive the session keys. This is because parameters $A_1$, $R_5$, $R_6$ and $R_7$ are still required. Therefore, $\bar{A}_0^{Game}$ and $\bar{A}_1^{Game}$ are indistinguishable and hence:

$$\bar{A}_0^{Game} = \bar{A}_1^{Game} \quad (2)$$

**$\bar{A}_2^{Game}$:** This game is achieved by simulating *Corrupt* ($SN_i$), *Hash* and *Corrupt* ($MD_i$). In accordance with *Definition 1* to *Definition* 3 in Section 3.1 above, messages $MAK_1$, $MAK_2$, $MAK_3$ and $MAK_4$ will not experience any hash collision due to the incorporation of random nonces in the hash values of their constituent parameters. In accordance with Lemma 13, the execution of *Corrupt* ($SN_i$) and *Corrupt* ($MD_i$) queries would not yield values required to derive the session keys. This is because $\bar{A}$ still needs $A_1$, $PW_U$ and the $U_i$'s $R_3$. To derive any valid $R_3 = A_4 \oplus h\ (UID_i \parallel PW_U)$ requires user password $PW_U$ and real identity $UID_i$. As such, both $\bar{A}_1^{Game}$ and $\bar{A}_2^{Game}$ are indistinguishable, devoid of these three queries. Suppose that the system only permits limited wrong passwords inputs. Then, based on Zipf's law of passwords and the birthday paradox, the following holds:

$$|Adv_{\bar{A}}^{Game_1} - Adv_{\bar{A}}^{Game_2}| \leq \frac{N_H^2}{2|H|} + z_1 N_S^{z_2} \quad (3)$$

All the queries described above are executed by $\bar{A}$. Therefore, it is only the random guessing of bit $c$ after the successful execution of the *Test*($\bar{I^1}$) query that might result in a game win. Therefore:

$$Adv_{\bar{A}}^{Game_2} = \frac{1}{2} \quad (4)$$

Solving Eq. (1), Eq. (2) and Eq. (4) yields the following:

$$\frac{1}{2} Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T) = |Adv_{\bar{A}}^{Game_0} - \frac{1}{2}| = |Adv_{\bar{A}}^{Game_1} - Adv_{\bar{A}}^{Game_2}| \quad (5)$$

On the other hand, solving Eqs. (3) and (5) results in the following:

$$\frac{1}{2} Adv_{\bar{A}}^{\dot{Z}}(\mathcal{P}_T) \leq \frac{N_H^2}{2|H|} + z_1 N_S^{z_2} \quad (6)$$

The multiplication of both side of this last equation by 2 yields the final equation as follows:

$$Adv_{\hat{A}}^{\hat{Z}}(\mathcal{P}_T) \leq \frac{N_H^2}{|H|} + 2z_1 N_S^{z_2} \quad (7)$$

Since Eq. (7) is similar to the formulated theorem, the security of the derived session keys has been successfully demonstrated.

### 4.2. Semantic security analysis

In this sub-section, various lemmas are formulated and proofed. The goal is to demonstrate the security features offered by this protocol. In addition, these lemmas show the resilience of the developed protocol against typical wireless sensor network attacks.

**Lemma 1.** *Untraceability and anonymity of the sensor nodes are upheld*

**Proof.** In this protocol, the $GWN_i$ generates pseudo-identity $PID_K$ for each sensor node $SN_i$. Next, it sends parameter set $\{SID_i, GS_C, GID_i, R_1, R_2, PID_K\}$ to the $SN_i$ over secure channels. During the mutual authentication and key negotiation phase, the $GWN_i$ transmits message $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$ to wireless sensor $SN_i$ over public channels. Evidently, $MAK_2$ contains sensor pseudo-identity $PID_K$ instead of the sensor real identity $SID_i$. After every successful mutual authentication process, the $SN_i$ derives new pseudo-identity as $PID_K^* = C_2 \oplus R_6 \oplus R_1$. Therefore, this pseudo-identity is session-specific and hence an attacker is unable to discern the communicating sensors. In addition, the adversary cannot determine whether any two different messages emanate from the same sensor node or not.

**Lemma 2.** *This protocol offers perfect backward and forward key secrecy*

**Proof.** The three communicating entities in this scheme include the user, gateway node and the sensor node. Here, the user derives the session key as $\psi_U = h(R_5 \oplus A_1 \| R_6 \| R_7)$ while the $SN_i$ computes the session key as $\psi_S = h(R_5 \oplus A_1 \| R_6 \| R_7)$. On the other hand, the $GWN_i$ calculates this session key as $\psi_G = h(R_5 \oplus A_1 \| R_6 \| R_7)$. Clearly, all these three session keys incorporate random nonces $R_5$, $R_6$ and $R_7$ that are then encapsulated in one-way hash function. Therefore, it is computationally impossible for the attacker to reverse this hash function and accurately derive these nonces. In addition, the adversary must accurately derive $A_1 = h(PW_U \| R_3)$. However, this requires knowledge of user password $PW_U$ and correct guessing of nonce $R_3$. During the mutual authentication process, the $GWN_i$ re-computes $R_5$ and $R_7$ as $R_5 = B_1 \oplus A_5 \oplus A_1$ and $R_7 = D_1 \oplus h(GS_C \| R_6) \oplus h(R_1^{**})$, while the $SN_i$ re-computes $R_6$ as $R_6 = C_1 \oplus B_4 \oplus SID_i \oplus R_1$. Similarly, $MD_i$ retrieves nonces $R_6$ and $R_7$ as $R_6 = D_4 \oplus h(R_5 \| UID_i)$ and $R_7 = D_5 \oplus h(R_5 \| R_6 \| A_1)$. Evidently, these nonces are frequently refreshed and hence the derived session keys are also stochastic. As such, an attacker is unable to obtain past and future sessions key upon compromise of the present session keys.

**Lemma 3.** *This protocol is resilient against spoofing and denial of sleep attacks*

**Proof.** The goal of these attacks is to send large quantities of old but valid messages using the spoofed user and sensors. During the mutual authentication process, messages $MAK_1$, $MAK_2$, $MAK_3$ and $MAK_4$ are exchanged. Here, $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$, $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$, $MAK_3 = \{C_5, D_1, D_2\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$. Upon receiving message $MAK_1$, the $GWN_i$ extracts $GID_i$ and $PID_i$ before retrieving the corresponding $UID_i$, $R_4$ and $A_1$ from its repository. If these values do not exist in the database, the request is flagged as malicious and the authentication session is stopped. On the other hand, upon receiving message $MAK_2$ from the $GWN_i$, the $SN_i$ validates $PID_K$ in this message against its equivalence in its memory. If this verification fails, the authentication session is stopped. Similarly, on receiving message $MAK_3$ from $SN_i$, the $GWN_i$ retrieves $R_1$ from its database and computes $R_1^{**} = h(R_1)$ as well as $R_2^{**} = R_6 \oplus B_4 \oplus C_5$. This is followed by the confirmation of whether $h(R_2) \stackrel{?}{=} h(R_2^{**})$ such that the session is terminated when this validation flops. Finally, on getting message $MAK_4$ from the $GWN_i$, the $U_i$ derives the session key as $\psi_U = h(R_5 \oplus A_1 \| R_6 \| R_7)$ followed by the computation of $E_2^* = h(\psi_U \| UID_i \| D_3 \| PID_i^*)$. Next, it confirms whether $E_2^* \stackrel{?}{=} E_2$ such that the session is terminated when these two values are dissimilar. As such, both spoofing and denial of sleep attacks are effectively detected and thwarted.

**Lemma 4.** *Impersonation attack is effectively thwarted*

**Proof.** The aim of this attack is to capture long term keying parameters and use them to masquerade as legitimate entities. To prevent this attack, only short term and transient parameters are deployed to derive the session keys. Suppose that an adversary has captured long term secret keys such as the gateway node private key, $SK_G$ and shared common key between the $GWN_i$ and $SN_i$, $GS_C$. Next, an attempt is made to impersonate $U_i$, $GWN_i$ and $SN_i$. This is achieved through bogus messages $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$, $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$, $MAK_3 = \{C_5, D_1, D_2\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$. Here, $B_1 = A_5 \oplus A_1 \oplus R_5$, $B_2 = SID_i \oplus h(UID_i \| R_5)$, $B_3 = h(PID_i \| GID_i \| SID_i \| A_5 \| UID_i \| R_5)$, $B_5 = (R_5 \oplus A_1 \oplus B_4 \oplus R_1$, $C_1 = R_6 \oplus B_4 \oplus SID_i \oplus R_1$, $C_2 = PID_K^* \oplus R_6 \oplus R_1$, $C_3 = h(R_6 \| R_1 \| B_4) \oplus R_2^*$, $C_4 = h(PID_K \| C_2 \| C_3 \| GS_C \| R_5 \oplus A_1 \| R_6)$, $C_5 = R_6 \oplus B_4 \oplus R_2$, $D_1 = h(GS_C \| R_6) \oplus h(R_1) \oplus R_7$, $D_2 = h(C_5 \| D_1 \| \psi_S \| SID_i \| GID_i \| R_7)$, $A_1 = h(PW_U \| R_3)$, $A_2 = h(PID_i \| R_4 \| GID_i \| SK_G) \oplus A_1$ and $A_5 = A_2 \oplus A_1$. It is clear that $SK_G$ is incorporated in $A_2$ while $GS_C$ is included in parameters $C_4$ and $D_1$. Therefore, impersonation attacks will fail due to unavailability of other keying parameters for the exchanged messages.

**Lemma 5.** *Password loss attacks are prevented*

**Proof.** The assumption made in this attack is that adversaries have captured user password $PW_U$. Next, attempts are made to compromise the security of the authentication process. In this protocol, $PW_U$ is incorporated in parameters $A_1 = h(PW_U \| R_3)$ and $A_4 = h(UID_i \| PW_U) \oplus R_3$. However, upon password compromise, the user invokes the four-step process to derive the new password $PW_U^*$ and new parameters $A_1^* = h(PW_U^* \| R_3^*)$, $A_2^* = F_3 \oplus h(R_5 \| UID_i) \oplus A_1^*$, $A_3^* = A_3 \oplus h(UID_i \| A_1) \oplus h(UID_i \| A_1^*)$, $F_2^* = h(PID_i^{**} \| PID_i \| GID_i \| A_5 \| E_5 \| UID_i \| R_5)$ and $F_4^* = h(UID_i \| PID_i \| PID_i^{**} \| A_5 \| F_3)$. As such, an adversary with the old password is unable to authenticate successfully. Any adversarial update of password $PW_U$ will fail since user identity $UID_i$ is required. In addition, the attacker lacks random nonce $R_5$ stored in the user's $MD_i$. This nonce is required to compute parameters $A_5 = A_2 \oplus A_1$, $E_3 = A_5 \oplus R_5$ and $E_4 = UID_i \oplus h(R_5 \| A_5)$.

**Lemma 6.** *This protocol resists KSSTI and replay attacks*

**Proof.** The assumption made in this attack is that the adversary has captured short term secrets such as nonces. In this protocol, three session keys are independently derived at the $U_i$, $GWN_i$ and

$SN_i$. These session keys include $\psi_U = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$, $\psi_S = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$ and $\psi_G = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$. Here, $R_5$, $R_6$ and $R_7$ are the random nonces while $A_1 = \mathrm{h}\,(PW_U \parallel R_3)$. As such, even if all the four nonces are captured by the adversary, the session keys cannot be derived. This is because the attacker still needs user password $PW_U$. The four messages exchanged during the mutual authentication procedures include $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$, $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$, $MAK_3 = \{C_5, D_1, D_2\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$. Clearly, none of these messages contain the plaintext $PW_U$. The only parameters that incorporate $PW_U$ are $A_1 = \mathrm{h}\,(PW_U \parallel R_3)$ and $A_4 = h\,(UID_i \parallel PW_U) \oplus R_3$. Therefore, parameters $B_1 = A_5 \oplus A_1 \oplus R_5$, $B_5 = (R_5 \oplus A_1 \oplus B_4 \oplus R_1$ and $C_4 = h\,(PID_K \parallel C_2 \parallel C_3 \parallel GS_C \parallel R_5 \oplus A_1 \parallel R_6)$ contain $PW_U$. However, the $PW_U$ in $A_1$ and $A_4$ is protected by the one-way hashing function. Since it is computationally difficulty to reverse the one-way hashing function, an attacker is unable to obtain user password $PW_U$ and hence KSSTI attacks flops.

**Lemma 7.** *Eavesdropping and session key disclosure attacks are vetoed.*

**Proof.** In the proposed protocol, the user, gateway node and sensor node independently derive the session keys as $\psi_U = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$, $\psi_S = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$ and $\psi_G = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$. Since $A_1 = \mathrm{h}\,(PW_U \parallel R_3)$, then four random nonces $R_3$, $R_5$, $R_6$ and $R_7$ are all required. Here, user password $PW_U$ is generated at the $MD_i$ and is transmitted together with $UID_i$ to the $GWN_i$ over some secure channels. Similarly, random nonces $R_3$ and $R_5$ are generated at the user's $MD_i$. On the other hand, $GWN_i$ generates random nonces $R_6$ while the $SN_i$ generates nonce $R_7$. During the mutual authentication procedures, none of the messages $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$, $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$, $MAK_3 = \{C_5, D_1, D_2\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$ convey these nonces in plaintext. As such, they cannot be eavesdropped over the public channel. By Lemma 6, an adversary cannot easily obtain password $PW_U$. In addition, the generated session keys are hashed values which cannot be reversed. Therefore, their keying parameters cannot be easily deciphered and hence session disclosure attack fails.

**Lemma 8.** *This scheme is robust against privileged insider attacks.*

**Proof.** The aim of the malicious users such as network administrators is to decipher credentials such as user real identity $UID_i$, password $PW_U$ and common key $GS_C$ shared between the $GWN_i$ and $SN_i$. Here, user password $PW_U$ is encapsulated in $A_1 = \mathrm{h}\,(PW_U \parallel R_3)$ and $A_4 = h\,(UID_i \parallel PW_U) \oplus R_3$. Due to the difficulty of reversing the one-way hashing function, an adversary cannot obtain $PW_U$ from these two values. To derive any valid $GS_C = h\,(SID_i \parallel SK_G \parallel NID_j)$, the attacker requires gateway node private key $SK_G$, sensor node $i$ unique identity $SID_i$ as well as network $j$'s unique identity $NID_j$. By Lemma 3, Lemma 4, Lemma 6 andLemma 7, none of the exchanged messages contain the plaintext $SK_G$, $SID_i$ $NID_j$. Therefore, the derivation of $GS_C$ and hence privileged insider attacks are effectively prevented.

**Lemma 9.** *User privacy is upheld.*

**Proof.** During the registration phase, the $GWN_i$ generates pseudo-identity $PID_i$ for each user $U_i$. During the authentication and key agreement phase, message $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$ carries this pseudo-identity instead of the user's real identity $UID_i$. Similarly, message $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$ incorporates $PID_i$ in parameters $D_3 = h\,(PID_i^* \parallel h\,(R_4) \parallel GID_i \parallel SK_G \parallel) \oplus h\,(R_5 \parallel A_1)$ and $E_2 = h\,(\psi_G \parallel UID_i \parallel D_3 \parallel PID_i^*)$. After successful validation of $D_2^*$ against $D_2$, the $GWN_i$ generates new pseudo-identity $PID_i^*$ for the user $U_i$ before storing $PID_K^*$ and $R_2^*$ in its database. Due to the usage of different user pseudo-identities for each session, it becomes difficult for the attacker to track users in this scheme.

**Lemma 10.** *MitM and offline guessing attacks are thwarted.*

**Proof.** Suppose that an adversary has captured authentication messages $MAK_1$, $MAK_2$, $MAK_3$ and $MAK_4$ exchanged among the $U_i$, $GWN_i$ and $SN_i$ over public channels. It is also assumed that the adversary has successfully extracted parameter set $\{A_2^*, A_3, A_4, PID_i^*\}$ stored in the $U_i$'s $MD_i$. The goal is to use these parameters to guess authentication parameters $D_2^* = h\,(C_5 \parallel D_1 \parallel \psi_G \parallel SID_i \parallel GID_i \parallel R_7)$ and $E_2^* = h\,(\psi_U \parallel UID_i \parallel D_3 \parallel PID_i^*)$. Clearly, values $C_5$, $D_1$, $GID_i$ and $D_3$ can be obtained from the exchanged messages, while $PID_i^*$ can be recovered from $MD_i$'s memory. However, the adversary still needs values $SID_i$, $\psi_G$ and $R_7$ to correctly guess $D_2^*$. Similarly, the adversary still requires $\psi_U$ and $UID_i$ to correctly guess value $E_2^*$. As such, offline guessing attack against this protocol fails.

**Lemma 11.** *This protocol is robust against forgery attacks.*

**Proof.** The aim of this attack is to forge gateway node authentication messages $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$. Here, $B_5 = (R_5 \oplus A_1 \oplus B_4 \oplus R_1$, $C_1 = R_6 \oplus B_4 \oplus SID_i \oplus R_1$, $C_2 = PID_K^* \oplus R_6 \oplus R_1$, $C_3 = h\,(R_6 \parallel R_1 \parallel B_4) \oplus R_2^*$, $C_4 = h\,(PID_K \parallel C_2 \parallel C_3 \parallel GS_C \parallel R_5 \oplus A_1 \parallel R_6)$, $D_3 = h\,(PID_i^* \parallel h\,(R_4) \parallel GID_i \parallel SK_G \parallel) \oplus h\,(R_5 \parallel A_1)$, $D_4 = h\,(R_5 \parallel UID_i) \oplus R_6$, $D_5 = h\,(R_5 \parallel R_6 \parallel A_1) \oplus R_7$, $E_1 = h\,(h\,(UID_i \parallel SK_G) \parallel R_7) \oplus PID_i^*$ and $E_2 = h\,(\psi_G \parallel UID_i \parallel D_3 \parallel PID_i^*)$. Evidently, any forgery of message $MAK_2$ requires gateway node private key $SK_G$, sensor node $i$ unique identity $SID_i$, $GS_C$ as well as nonces $R_1$, $R_5$ and $R_6$ among other values. Similarly, any forgery of message $MAK_4$ requires knowledge of $SK_G$, $UID_i$, $GID_i$ as well as nonces $R_4$, $R_5$, $R_6$ and $R_7$ among other values. Since $SK_G$, $SID_i$, $GS_C$ and $UID_i$ as well as nonces are never exchanged in plaintext over the public channels, they cannot be intercepted by the adversary. Therefore gateway forgery flops. Suppose that the adversary is interested in forging user authentication message $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$. Here, $B_1 = A_5 \oplus A_1 \oplus R_5$, $B_2 = SID_i \oplus h\,(UID_i \parallel R_5)$, $B_3 = h\,(PID_i \parallel GID_i \parallel SID_i \parallel A_5 \parallel UID_i \parallel R_5)$, $A_5 = A_2 \oplus A_1$, $A_1 = \mathrm{h}\,(PW_U \parallel R_3)$ and $A_2 = h\,(PID_i \parallel R_4 \parallel GID_i \parallel SK_G) \oplus A_1$. Evidently, the attacker needs $SK_G$, $SID_i$, $UID_i$, $PW_U$ as well as nonces $R_3$, $R_4$ and $R_5$. By *Lemma* 11, none of these parameters are transmitted in plaintext over the public channels. As such, they cannot be intercepted by the adversary and hence this forgery fails.

**Lemma 12.** *This scheme is resilient against physical capture and side-channeling attacks.*

**Proof.** The assumption in this attack is that an adversary can physically capture the sensor node $SN_i$ as well as mobile device $MD_i$. Afterwards, side-channeling is deployed to extract all the secret tokens stored in their memories. Thereafter, an attempt is made to compromise any other sensor node $SN_p$.

*Case 1:* During the registration phase, parameter set $\{SID_i, GS_C, GID_i, R_1, R_2, PID_K\}$ is stored in sensor memory. Using the retrieved values, the attacker attempts to compromise $SN_p$'s message $MAK_3 = \{C_5, D_1, D_2\}$. Here, $C_5 = R_6 \oplus B_4 \oplus R_2$, $B_4 = h\,(GS_C \parallel GID_i) \oplus h\,(R_1) \oplus R_7$, $D_2 = h\,(C_5 \parallel D_1 \parallel \psi_S \parallel SID_i \parallel GID_i \parallel R_7)$, $\psi_S = h\,(R_5 \oplus A_1 \parallel R_6 \parallel R_7)$ and $A_1 = \mathrm{h}\,(PW_U \parallel R_3)$. Evidently, the attacker needs knowledge of user password $PW_U$ as well as random nonces $R_6$ and $R_7$. Since these parameters cannot be recovered from sensor memory and over public channels, the compromise of message $MAK_3$ fails. In

addition, the recovered $SID_i$ and $PID_K$ belong to $SN_i$ and not $SN_p$. Similarly, the common key is shared between the $GWN_i$ and $SN_i$, and not with $SN_p$. As such, the derivation of parameters $C_5$, $D_1$, $D_2$, $\psi_S$ and $A_1$ flops. Therefore, sensor $SN_p$'s authentication message $MAK_3$ is secure even when other sensor nodes are under active physical attacks.

*Case 2:* After successful registration at the $GWN_i$, parameter set $\{A_2, A_3, A_4, PID_i, GID_i\}$ is stored in $U_i$'s mobile device $MD_i$. During the AKA procedures, the $MD_i$ transmits message $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$ to the $GWN_i$ over public channels. Here, $B_1 = A_5 \oplus A_1 \oplus R_5$, $B_2 = SID_i \oplus h(UID_i \| R_5)$ and $B_3 = h(PID_i \| GID_i \| SID_i \| A_5 \| UID_i \| R_5)$. It is clear that the derivation of $U_i$'s message $MAK_1$ requires $SID_i$, $UID_i$, $PW_U$ as well as nonces $R_3$ and $R_5$. Evidently, none of these parameters can be recovered from $MD_i$'s memory. Therefore, both physical capture and side-channeling attacks fail.

**Lemma 13.** *Confidentiality and integrity are upheld in this protocol.*

**Proof.** Suppose that attacker is interested in eavesdropping the exchanged messages and access confidential information such as the real identity of the communicating entities. In addition, the attacker may try to capture and modify the exchanged messages and hence compromising their integrity. However, in accordance with Lemma 7, eavesdropping and session key disclosure attacks are vetoed. In addition, Lemma 1 demonstrates that untraceability and anonymity of the sensor nodes are upheld and hence cannot be discerned from the exchanged messages. Regarding integrity, Lemma 2 describes the difficulty of deriving past and future sessions key upon compromise of the present session keys and hence their integrity of these session keys is upheld. Similarly, Lemma 6 demonstrates the difficulty of using captured short term secrets such as nonces to launch both replay and KSSTI attacks. Since key secrecy is preserved, and both repay and KSSTI attacks are thwarted, the proposed protocol prevents adversarial modification of the exchanged messages and hence integrity is upheld.

**Lemma 14.** *The proposed protocol is scalable.*

**Proof.** During the mutual authentication and key agreement phase, no verifier tables need to be maintained by the $U_i$, $GWN_i$ or $SN_i$. As such, more users and sensor nodes can join and leave the network seamlessly without adversely affecting its performance. This is unlike schemes that are based on verifier tables where an increase in the number of users or sensors can lead to long search times in these verifier tables, which degrade network performance.

## 5. Performance evaluation

In this section, the proposed protocol is evaluated in terms of computation overheads, communication costs and supported security characteristics. These three metrics are selected because of their frequent deployment during performance evaluation of authentication protocols. The detailed descriptions of these costs are illustrated in the sub-sections below.

### 5.1. Computation costs

The various cryptographic operations executed during the mutual authentication and key negotiation phase are taken into consideration here. During this phase, the $U_i$ executes 10 one-way hashing operations while the gateway node carries out 17 one-way hashing operations. On the other hand, the sensor node executes 7 one-way hashing operations. Therefore, a total of 34 hashing operations are executed. The experimentations were executed in a HP ProBook 430 G2 laptop machine equipped with Intel® Core (TM) i5-4210U CPU @ 2.4 GHz, RAM size of 4 GB and running on Windows 10 Pro 64-bit. In addition, the Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) was deployed for the various cryptographic primitives. Table 3 presents the execution durations for the various cryptographic operations.

**Table 3**
Cryptographic durations.

| Operation | Duration (ms) |
|---|---|
| One-way hashing ($T_H$) | 0.25 |
| Symmetric encryption/decryption ($T_{ED}$) | 0.67 |
| Elliptic curve point multiplication ($T_M$) | 5.74 |
| Chebyshev polynomial ($T_{CP}$) | 2.25 |
| Fuzzy extraction ($T_{FE}$) | 5.42 |
| Elliptic curve point addition ($T_A$) | 0.72 |
| Modular square ($T_{MS}$) | 0.93 |
| Quadratic residual ($T_{QR}$) | 1.87 |
| Biometric hash function ($T_{BH}$) | 2.31 |

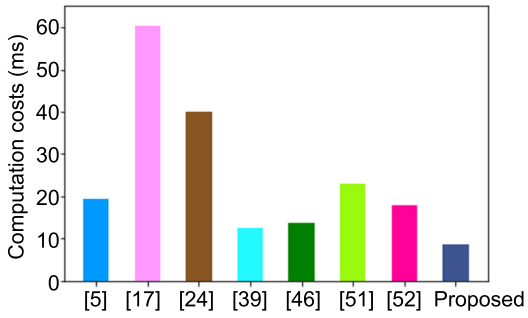


**Fig. 4.** Computation costs comparisons.

Using the values in Table 3, the overall execution time for the proposed protocol is 8.50 ms. This value is then compared with other related schemes in [5,17,24,39,46,51,52] as shown in Table 4.

As shown in Table 4, the protocols in [5,17,24,39,46,51,52] have computation costs of 19.50 ms, 12.42 ms, 60.34 ms, 17.86 ms, 13.67 ms, 39.94 ms and 22.99 ms respectively. On the other hand, the proposed protocol incurs a computation cost of only 8.50 ms. Based on Fig. 4, the scheme in [17] incurs the highest computation overheads while the proposed protocol incurs the least computation costs.

Among the related schemes, the one developed in [39] has the least computation overhead of 12.42 ms. Using this value as the benchmark, the proposed protocol achieves a 31.56% reduction in computation costs and is therefore the most applicable for resource-limited sensor node devices.

### 5.2. Communication costs

During the mutual authentication and key agreement phase, four messages are exchanged. These messages include $MAK_1 = \{PID_i, GID_i, B_1, B_2, B_3\}$, $MAK_2 = \{PID_K, B_5, C_1, C_2, C_3, C_4\}$, $MAK_3 = \{C_5, D_1, D_2\}$ and $MAK_4 = \{D_3, D_4, D_5, E_1, E_2\}$. Here, $B_1 = A_5 \oplus A_1 \oplus R_5$, $B_2 = SID_i \oplus h(UID_i \| R_5)$, $B_3 = h(PID_i \| GID_i \| SID_i \| A_5 \| UID_i \| R_5)$, $B_5 = (R_5 \oplus A_1 \oplus B_4 \oplus R_1$, $C_1 = R_6 \oplus B_4 \oplus SID_i \oplus R_1$, $C_2 = PID_K^* \oplus R_6 \oplus R_1$, $C_3 = h(R_6 \| R_1 \| B_4) \oplus R_2^*$, $C_4 = h(PID_K \| C_2 \| C_3 \| GS_C \| R_5 \oplus A_1 \| R_6)$, $C_5 = R_6 \oplus B_4 \oplus R_2$, $D_1 = h(GS_C \| R_6) \oplus h(R_1) \oplus R_7$, $D_2 = h(C_5 \| D_1 \| \psi_S \| SID_i \| GID_i \| R_7)$, $D_3 = h(PID_i^* \| h(R_4) \| GID_i \| SK_G \|) \oplus h(R_5 \| A_1)$, $D_4 = h(R_5 \| UID_i) \oplus R_6$, $D_5 = h(R_5 \| R_6 \| A_1) \oplus R_7$, $E_1 = h(h(UID_i \| SK_G) \| R_7) \oplus PID_i^*$ and $E_2 = h(\psi_G \| UID_i \| D_3 \| PID_i^*)$. For fair comparisons, the values in [5] are deployed. Table 5 presents the sizes of the various cryptographic operation outputs.

**Table 4**
Computation costs comparisons.

| Scheme | $U_i$ | $GWN_i$ | $SN_i$ | Total cost | Cost (ms) |
|---|---|---|---|---|---|
| Mo et al. [5] | $11T_H + 3T_{CP}$ | $9T_H + T_{CP}$ | $4T_H + 2T_{CP}$ | $24T_H + 6T_{CP}$ | 19.50 |
| Moghadam et al. [17] | $16T_H + 4T_M + 2T_{ED}$ | $5T_H + 3T_M + 2T_{ED}$ | $3T_H + 2T_M$ | $24T_H + 9T_M + 4T_{ED}$ | 60.34 |
| Li et al. [24] | $10T_H + 3T_M$ | $8T_H + T_M$ | $4T_H + 2T_M$ | $22T_H + 6T_M$ | 39.94 |
| Yu et al. [39] | $T_{FE} + 11T_H$ | $11T_H$ | $6T_H$ | $T_{FE} + 28T_H$ | 12.42 |
| Shin et al. [46] | $T_{FE} + 12T_H$ | $15T_H$ | $6T_H$ | $T_{FE} + 33T_H$ | 13.67 |
| Wang et al. [51] | $T_{FE} + 6T_H + 3T_{CP}$ | $7T_H + T_{CP}$ | $4T_H + 2T_{CP}$ | $T_{FE} + 17T_H + 6T_{CP}$ | 22.99 |
| Xu et al. [52] | $5T_H + 2T_{CP} + T_{BH} + T_{MS}$ | $6T_H + T_{QR}$ | $4T_H + 2T_{CP}$ | $15T_H + 4T_{CP} + T_{BH} + T_{MS} + T_{QR}$ | 17.86 |
| **Proposed** | $10T_H$ | $17T_H$ | $7T_H$ | $34\ T_H$ | 8.50 |

**Table 5**
Cryptographic output sizes.

| Operation | Size (bits) |
|---|---|
| Hash values | 160 |
| Blocks of symmetric encryption/decryption | 128 |
| Points in elliptic curve | 320 |
| Chebyshev polynomials | 128 |
| Random nonce | 128 |
| Identities | 32 |
| Timestamps | 32 |

**Table 6**
Communication costs comparisons.

| Scheme | Cost (bits) |
|---|---|
| Mo et al. [5] | 1728 |
| Moghadam et al. [17] | 2688 |
| Li et al. [24] | 2720 |
| Yu et al. [39] | 2880 |
| Shin et al. [46] | 1888 |
| Wang et al. [51] | 1888 |
| Xu et al. [52] | 2304 |
| **Proposed** | 2656 |

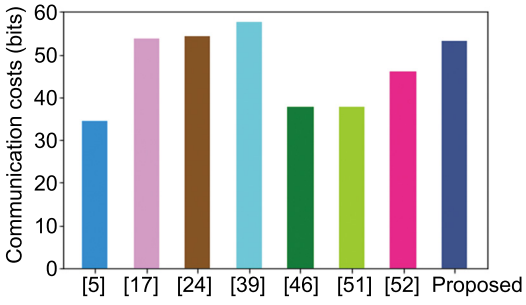


**Fig. 5.** Communication costs comparisons.

Based on the values in Table 5, the size of $MAK_1$ = {32+32+160+160+160} = 544 bits. On the other hand, the size of $MAK_2$ = {32+160+160+160+160+160} = 832 bits, while $MAK_3$= {160+160+160} = 480 bits. Similarly, $MAK_4$ = {160+160+160+160+160} = 800 bits. Consequently, the cumulative communication cost in the proposed protocol is 2656 bits as shown in Table 6.

Based on the values in Table 6, the schemes in [5,17,24,39,46,51,52] have communication costs of 1728 bits, 2880 bits, 2688 bits, 2304 bits, 1888 bits, 2720 bits and 1888 bits respectively. As shown in Fig. 5, the protocol in [5] has the least communication costs of 1728 bits.

This is followed by the schemes in [46,51,52] which incur 1888 bits, 1888 bits and 2304 bits respectively. Although the protocol in [5] incurs the least communication overheads, it is never evaluated against side-channeling, spoofing, password loss, forgery and denial of sleep attacks. Similarly, the scheme in [46] does not offer key secrecy and is never evaluated against side-channeling, offline guessing, spoofing, password loss, forgery and denial of sleep attacks. In addition, it cannot withstand privileged insider, impersonation, session key disclosure, eavesdropping and physical capture attacks [5]. On its part, the protocol in [51] has not been evaluated against side-channeling, offline guessing, spoofing, password loss, forgery and denial of sleep attacks. Finally, the scheme in [52] is susceptible to impersonation, MitM, privileged insider and KSSTI attacks. Therefore, the proposed protocol incurs relatively higher communication overheads but with the strongest security as discussed in Section 5.3.

### 5.3. Supported security features

In this part, the security characteristics of the proposed protocol are compared with those of other related schemes. Table 7 gives a summary of these comparative evaluations. As shown in Table 7, the protocols in [5,17,24,39,46,51,52] and the proposed protocol supports 15, 12, 9, 10, 8, 11, 14 and 20 security characteristics respectively.

Based on these supported features, the scheme in [46] supports only 8 features and hence is the most insecure for WSNs. This is followed by the protocols in [5,17,24,39,51,52] which support 9, 10, 11, 12, 14 and 15 features respectively. On the other hand, the proposed protocol supports all the 20 security characteristics. Therefore, using the scheme in [5] as the benchmark, the proposed protocol posts a 33.33% improvement in the number of supported security characteristics.

### 5.4. Practical considerations

In a typical WSN, sensors collect and forward it the gateway nodes over wireless channels. The gateway nodes in turn forward these data items to the remote users for further processing. In the proposed protocol, the same modus operandi is adopted and hence this protocol is compatible with the existing WSN systems. During the mutual authentication and key negotiation phase, only exclusive Or (XOR) and one-way hashing operations are executed. This renders the proposed protocol lightweight; hence it has high usability in the resource-constrained WSN environment. It has also been demonstrated that this protocol is resilient against numerous WSN attack vectors such as packet replays, KSSTI, session key disclosure, forgery, impersonation, MitM, side-channeling, spoofing, physical capture, password loss, eavesdropping, DoS and offline guessing. This renders it ideal for deployment in WSN to protect against the mentioned attacks.

## 6. Conclusion

The wireless sensor networks find applications in a number of fields such as military surveillance and intelligent transportation, where public channels are deployed to convey the sensed data to remote users. However, most of these deployment environments are hostile or unattended. Therefore, these channels and environments expose the sensor nodes to numerous threats such as physical node capture and privacy disclosure. As such, many schemes have been presented in literature to curb these

**Table 7**
Security characteristics comparisons.

| Method | Security features | | | | | | Robust against | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | User privacy | Anonymity | Untraceability | Authentication | Session key agreement | Key secrecy | Side-channeling | Physical capture | Eavesdropping | Offline guessing | Spoofing | Password loss | Session key disclosure | Forgery | Impersonation | MitM | Privileged insider | KSSTI | Denial of sleep | Replay |
| [5] | √ | √ | √ | √ | √ | √ | - | √ | √ | √ | - | - | √ | - | √ | √ | √ | √ | - | √ |
| [39] | √ | √ | √ | √ | √ | √ | - | √ | √ | - | - | - | √ | - | × | √ | √ | × | - | √ |
| [17] | √ | √ | √ | √ | √ | × | - | √ | × | - | - | - | × | - | × | √ | √ | √ | - | × |
| [52] | √ | √ | √ | √ | √ | √ | - | √ | √ | - | - | - | √ | - | × | × | × | × | - | √ |
| [46] | √ | √ | √ | √ | √ | × | - | × | × | - | - | - | × | - | × | √ | × | √ | - | √ |
| [24] | √ | √ | √ | √ | √ | √ | - | √ | √ | - | - | - | √ | - | √ | √ | × | × | - | × |
| [51] | √ | √ | √ | √ | √ | √ | - | √ | √ | - | - | - | √ | - | √ | √ | √ | √ | - | √ |
| **Proposed** | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |

√ Supported; × Not supported; - Not considered.

challenges. In these protocols, techniques such as blockchain, passwords and public key cryptography are utilized. However, these technologies have been shown to be either inefficient for the sensor nodes or they still have security holes that can be exploited by attackers. In this regard, this paper developed a lightweight authentication protocol, whose formal security analysis using the Real-or Random (ROR) model has demonstrated its ability to preserve session key security. In addition, informal security analysis has shown its robustness to numerous attack vectors such as spoofing, physical capture, password loss and eavesdropping. Moreover, the comparative performance evaluation has demonstrated that it has relatively lower communication overheads, reduces the computation costs by 31.56% and improves the supported security characteristics by 33.33%. As part of the future work, we will explore ways in which the communication overheads of this protocol can be reduced further.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


[1] M. Dener, A. Orman, BBAP-WSN: A new blockchain-based authentication protocol for wireless sensor networks, Appl. Sci. 13 (3) (2023) 1526.
[2] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, B. Fang, A survey on access control in the age of internet of things, IEEE Internet Things J. 7 (6) (2020) 4682–4696.
[3] A. Abdollahi, K. Rejeb, A. Rejeb, M.M. Mostafa, S. Zailani, Wireless sensor networks in agriculture: Insights from bibliometric analysis, Sustainability 13 (21) (2021) 12011.
[4] P. Kumar, S. Bhushan, M. Kumar, M. Alazab, Secure key management and mutual authentication protocol for wireless sensor network by linking edge devices using hybrid approach, Wirel. Pers. Commun. (2023) 1–23.
[5] J. Mo, Z. Hu, W. Shen, A provably secure three-factor authentication protocol based on chebyshev chaotic mapping for wireless sensor network, IEEE Access 10 (2022) 12137–12152.
[6] V.O. Nyangaresi, Privacy preserving three-factor authentication protocol for secure message forwarding in Wireless Body Area networks, Ad Hoc Netw. 142 (2023) 103117.
[7] M. Azrour, J. Mabrouki, A. Guezzaz, A. Kanwal, Internet of things security: challenges and key issues, Secur. Commun. Netw. 2021 (2021) 1–11.
[8] A. Shahraki, A. Taherkordi, Ø. Haugen, F. Eliassen, A survey and future directions on clustering: From WSNs to IoT and modern networking paradigms, IEEE Trans. Netw. Serv. Manag. 18 (2) (2020) 2242–2274.
[9] E. Yuan, L. Wang, S. Cheng, N. Ao, Q. Guo, A key management scheme based on pairing-free identity based digital signature algorithm for heterogeneous wireless sensor networks, Sensors 20 (6) (2020) 1543.
[10] A.K. Gautam, R. Kumar, A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks, SN Appl. Sci. 3 (1) (2021) 50.
[11] Z. Ding, Q. Xie, Provably secure dynamic anonymous authentication protocol for wireless sensor networks in internet of things, Sustainability 15 (7) (2023) 5734.
[12] Z. Xu, M. Cai, X. Li, T. Hu, Q. Song, Edge-aided reliable data transmission for heterogeneous edge-iot sensor networks, Sensors 19 (9) (2019) 2078.
[13] S. Bhushan, M. Kumar, P. Kumar, T. Stephan, A. Shankar, P. Liu, FAJIT: a fuzzy-based data aggregation technique for energy efficiency in wireless sensor network, Complex Intell. Syst. 7 (2021) 997–1007.
[14] V.O. Nyangaresi, Lightweight anonymous authentication protocol for resource-constrained smart home devices based on elliptic curve cryptography, J. Syst. Archit. 133 (2022) 102763.
[15] S.A. Chaudhry, K. Yahya, S. Garg, G. Kaddoum, M.M. Hassan, Y.B. Zikria, LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments, IEEE Trans. Ind. Inform. 19 (2) (2022) 1504–1511.
[16] A. Irshad, S.A. Chaudhry, M. Sher, B.A. Alzahrani, S. Kumari, X. Li, F. Wu, An anonymous and efficient multiserver authenticated key agreement with offline registration centre, IEEE Syst. J. 13 (1) (2018) 436–446.
[17] M.F. Moghadam, M. Nikooghadam, M.A.B. Al Jabban, M. Alishahi, L. Mortazavi, A. Mohajerzadeh, An efficient authentication and key agreement scheme based on ECDH for wireless sensor network, IEEE Access 8 (2020) 73182–73192.
[18] D.K. Kwon, S.J. Yu, J.Y. Lee, S.H. Son, Y.H. Park, WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks, Sensors 21 (3) (2021) 936.
[19] S. Awan, N. Javaid, S. Ullah, A.U. Khan, A.M. Qamar, J.G. Choi, Blockchain based secure routing and trust management in wireless sensor networks, Sensors 22 (2) (2022) 411.
[20] V.O. Nyangaresi, A formally validated authentication algorithm for secure message forwarding in smart home networks, SN Comput. Sci. 3 (5) (2022) 364.
[21] P. Chandrakar, A secure remote user authentication protocol for healthcare monitoring using wireless medical sensor networks, Int. J. Ambient Comput. Intell. (IJACI) 10 (1) (2019) 96–116.
[22] X. Liu, Z. Guo, J. Ma, Y. Song, A secure authentication scheme for wireless sensor networks based on DAC and Intel SGX, IEEE Internet Things J. 9 (5) (2021) 3533–3547.
[23] R. Amin, S.H. Islam, G.P. Biswas, M.K. Khan, N. Kumar, A robust and anonymous patient monitoring system using wireless medical sensor networks, Future Gener. Comput. Syst. 80 (2018) 483–495.
[24] X. Li, J. Peng, M.S. Obaidat, F. Wu, M.K. Khan, C. Chen, A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems, IEEE Syst. J. 14 (1) (2019) 39–50.
[25] C.T. Chen, C.C. Lee, I.C. Lin, Efficient and secure three-party mutual authentication key agreement protocol for WSNs in IoT environments, Plos one 15 (4) (2020) e0232277.
[26] B. Hu, W. Tang, Q. Xie, A two-factor security authentication scheme for wireless sensor networks in IoT environments, Neurocomputing 500 (2022) 741–749.
[27] U. Gulen, S. Baktir, Elliptic curve cryptography for wireless sensor networks using the number theoretic transform, Sensors 20 (5) (2020) 1507.
[28] A.B. Feroz Khan, G. Anandharaj, A cognitive energy efficient and trusted routing model for the security of wireless sensor networks: CEMT, Wirel. Pers. Commun. 119 (4) (2021) 3149–3159.
[29] V.O. Nyangaresi, M. Ahmad, A. Alkhayyat, W. Feng, Artificial neural network and symmetric key cryptography based verification protocol for 5G enabled Internet of Things, Expert Syst. 39 (10) (2022) e13126.
[30] M.H. Dahshan, Robust data authentication for unattended wireless sensor networks, Telecommun. Syst. 66 (2) (2017) 181–196.
[31] F. Wu, L. Xu, S. Kumari, X. Li, An improved and provably secure three-factor user authentication scheme for wireless sensor networks, Peer-to-Peer Netw. Appl. 11 (2018) 1–20.
[32] M.B. Apsara, P. Dayananda, C.N. Sowmyarani, A review on secure group key management schemes for data gathering in wireless sensor networks, Eng. Technol. Appl. Sci. Res. 10 (1) (2020) 5108–5112.
[33] S. Gupta, S. Gupta, D. Goyal, Comparison of Q-coverage P-connectivity sensor node scheduling heuristic between battery powered WSN & energy harvesting WSN, Int. J. Sens. Wirel. Commun. Control 11 (5) (2021) 553–559.

[34] T.M. Butt, R. Riaz, C. Chakraborty, S.S. Rizvi, A. Paul, Cogent and energy efficient authentication protocol for wsn in iot, Comput. Mater. Contin. 68 (2021) 1877–1898.
[35] C.H. Liu, Y.F. Chung, Secure user authentication scheme for wireless healthcare sensor networks, Comput. Electr. Eng. 59 (2017) 250–261.
[36] V.O. Nyangaresi, M. Abd-Elnaby, M.M. Eid, A. Nabih Zaki Rashed, Trusted authority based session key agreement and authentication algorithm for smart grid networks, Trans. Emerg. Telecommun. Technol. 33 (9) (2022) e4528.
[37] Y. Lu, G. Xu, L. Li, Y. Yang, Anonymous three-factor authenticated key agreement for wireless sensor networks, Wirel. Netw. 25 (2019) 1461–1475.
[38] J. Mo, H. Chen, A lightweight secure user authentication and key agreement protocol for wireless sensor networks, Secur. Commun. Netw. 2019 (2019) 1–17.
[39] S. Yu, Y. Park, SLUA-WSN: Secure and lightweight three-factor-based user authentication protocol for wireless sensor networks, sensors 20 (15) (2020) 4143.
[40] M. Shuai, N. Yu, H. Wang, L. Xiong, Y. Li, A lightweight three-factor Anonymous authentication scheme with privacy protection for personalized healthcare applications, J. Organ. End User Comput. (JOEUC) 33 (3) (2021) 1–18.
[41] Q. Xie, Z. Ding, B. Hu, A secure and privacy-preserving three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things, Secur. Commun. Netw. 2021 (2021) 1–12.
[42] T.Y. Wu, L. Yang, Z. Lee, S.C. Chu, S. Kumari, S. Kumar, A provably secure three-factor authentication protocol for wireless sensor networks, Wirel. Commun. Mob. Comput. 2021 (2021) 1–15.
[43] D. Kumar, A secure and efficient user authentication protocol for wireless sensor network, Multimedia Tools Appl. 80 (18) (2021) 27131–27154.
[44] A. Arivarasi, P. Ramesh, An improved source location privacy protection using adaptive trust sector-based authentication with honey encryption algorithm in WSN, J. Ambient Intell. Humaniz. Comput. (2021) 1–13.
[45] V.O. Nyangaresi, Terminal independent security token derivation scheme for ultra-dense IoT networks, Array 15 (2022) 100210.
[46] S. Shin, T. Kwon, A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes, Sensors 19 (9) (2019) 2012.
[47] D. Rangwani, D. Sadhukhan, S. Ray, M.K. Khan, M. Dasgupta, A robust provable-secure privacy-preserving authentication protocol for industrial internet of things, Peer-to-peer Netw. Appl. 14 (2021) 1548–1571.
[48] A. Jabbari, J.B. Mohasefi, Improvement of a user authentication scheme for wireless sensor networks based on internet of things security, Wirel. Pers. Commun. 116 (3) (2021) 2565–2591.
[49] Q. Jiang, S. Zeadally, J. Ma, D. He, Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks, Ieee Access 5 (2017) 3376–3392.
[50] S.H. Alsamhi, A.V. Shvetsov, S. Kumar, S.V. Shvetsova, M.A. Alhartomi, A. Hawbani, et al., UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation, Drones 6 (7) (2022) 154.
[51] F. Wang, G. Xu, G. Xu, A provably secure anonymous biometrics-based authentication scheme for wireless sensor networks using chaotic map, IEEE Access 7 (2019) 101596-101608.
[52] G. Xu, F. Wang, M. Zhang, J. Peng, Efficient and provably secure anonymous user authentication scheme for patient monitoring using wireless medical sensor networks, IEEE Access 8 (2020) 47282–47294.
[53] M.A. Ferrag, L.A. Maglaras, H. Janicke, J. Jiang, L. Shu, Authentication protocols for internet of things: a comprehensive survey, Secur. Commun. Netw. (2017).
[54] T. Salman, M. Zolanvari, A. Erbad, R. Jain, M. Samaka, Security services using blockchains: A state of the art survey, IEEE Commun. Surv. Tutor. 21 (1) (2018) 858–880.
[55] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with IoT. Challenges and opportunities, Future Gener. Comput. Syst. 88 (2018) 173–190.