Article

Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things

Anup Kumar Maurya¹,²,* and V. N. Sastry¹

¹ Center for Mobile Banking, Institute for Development and Research in Banking Technology, Hyderabad 500057, India; vnsastry@idrbt.ac.in
² Artificial Intelligence Lab, School of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, India

* Correspondence: anupmaurya88@gmail.com or akmaurya@idrbt.ac.in; Tel.: +91-898-566-4374

Received: 21 September 2017; Accepted: 24 October 2017; Published: 30 October 2017

Abstract: To improve the quality of service and reduce the possibility of security attacks, a secure and efficient user authentication mechanism is required for Wireless Sensor Networks (WSNs) and the Internet of Things (IoT). Session key establishment between the sensor node and the user is also required for secure communication. In this paper, we perform the security analysis of A.K.Das’s user authentication scheme (given in 2015), Choi et al.’s scheme (given in 2016), and Park et al.’s scheme (given in 2016). The security analysis shows that their schemes are vulnerable to various attacks like user impersonation attack, sensor node impersonation attack and attacks based on legitimate users. Based on the cryptanalysis of these existing protocols, we propose a secure and efficient authenticated session key establishment protocol which ensures various security features and overcomes the drawbacks of existing protocols. The formal and informal security analysis indicates that the proposed protocol withstands the various security vulnerabilities involved in WSNs. The automated validation using AVISPA and Scyther tool ensures the absence of security attacks in our scheme. The logical verification using the Burrows-Abadi-Needham (BAN) logic confirms the correctness of the proposed protocol. Finally, the comparative analysis based on computational overhead and security features of other existing protocol indicate that the proposed user authentication system is secure and efficient. In future, we intend to implement the proposed protocol in real-world applications of WSNs and IoT.

Keywords: Wireless Sensor Networks (WSNs); Internet of Things (IoT); user authentication; session key; smart card; fuzzy extractor; hash function

1. Introduction

Recent advancements in the micro-electro-mechanical system enable the production of low-cost sensor nodes with small-scale sensing module, a radio frequency transceiver, a small processing module for limited computation, small-scale memory and a short-lived power unit. For instance, a sensor node can have temperature, pressure, humidity and light sensors with 7.7 MHz 8-bit ATmega 128 processor, 4 K byte RAM, 128 K byte ROM, 512 K byte EEPROM, and 2 AA battery. The sensing module may consist of few sensors with analog to digital converters (ADCs). These sensors can measure the change in physical parameters such as temperature, humidity, light, pressure. The analog signals produced by the sensor node based on the measured physical parameters can be transformed into the digital signal using ADC. Then, the digital signals can be fed into the processing element to perform the necessary calculation on raw data, and the transceiver unit communicates with its adjacent sensor nodes. Nowadays, we find sensors are on our smart phones, watches, vehicles, homes, offices, cities, and industries which connect our world more than we ever thought possible.
A WSN [1] or IoT [2] may consist of a large number of scattered sensor nodes capable of collecting data from their surroundings for specific users, communicating with the neighboring sensor nodes using wireless medium and routing the data to the gateway node having trusted high-performance computing resources. Some important aspects of WSNs are as follows:

- The sensor nodes of WSNs sufferer with energy constraints, memory limitations, unreliable communications, higher latency in communication and unattended operation of networks.
- The topology of WSNs can vary very often.
- The sensor node can be deployed densely in WSNs area.

The IoT aims at overcoming the gap between the physical world and its characterization within the digital world. The term things refer to an object that has sensors attached to it, and can transmit data to the internet, where it can be processed, analyzed and used to make decisions, one such example is medical health care system.

An example of medical health care system for monitoring patient’s condition and recovery by authentic medical practitioners and doctors using wireless body area network (WBAN) is shown in Figure 1. The sensor nodes are planted in patient’s body for measuring various parameters like ECG, blood pressure, temperature, visual straight, etc. The measured parameters from different sensor nodes are transmitted to a master sensor node. The master sensor node processes the data locally and sends to the gateway node. Only the authentic medical practitioners and doctors are allowed to access the confidential and real-time data of high-profile patients from the master sensor node and the gateway respectively.

![Figure 1. Wireless body area network (WBAN).](image)

The conventional specializations of WSNs, embedded systems, control systems and automation (including smart home, smart city, industry and building automation) contribute to facilitating the IoT. The advances in IoT technology facilitate wearable devices which broadly cover health, fitness and entertainment requirements. These devices are installed with sensors which collect the sensitive data about the human beings and transmit these data to the neighboring device, base station or gateway node for further processing and analysis. If the data is security sensitive, only an authentic user should be allowed to pre-process the data to extract essential insights about the patient. With the rise of IoT where the number of sensor devices would grow multi-fold, it would be infeasible for a user to make the system secure using traditional authentication mechanism.
Therefore, it is important to address this concern by devising ways in which multiple advanced factors of authentication and session key establishment would be required to gain access to any smart devices of WSNs/IoT and at the same time its usability would be at high level.

The members of a smart home, city, and office (which has an automated system for monitoring temperature, light, air conditioners, windows, doors, refrigerator, alarms, alerts, etc.) should be given access by configuring the security system. However, to enhance system’s security, it is important to have multiple hierarchies of authentication and session key establishment scheme. Authenticating users who connect to the sensor nodes of WSNs and IoT is a process of validating the identity (based on one or more factors such as user’s inherence, possession, knowledge) using sensor devices. The security of traditional user authentication protocols for WSNs is based on low entropy password which is easy to break through dictionary attacks. However, the biometric information can not be lost, forgotten, guessed easily or shared. Therefore, the biometric based user authentication scheme is more secure and reliable than traditional password based systems.

From last decades, WSNs and IoT have drawn attention in many applications including health-care, battlefield surveillance, smart home, smart banking, financial office and other secure, real-time applications where efficient user authentication and session key establishment is required. A secure and efficient user authentication scheme should provide various security features (e.g., confidentiality, integrity, freshness, etc.) and it should resist various security attacks (e.g., user impersonation, sensor impersonation, stolen smart card and energy exhausting attacks, etc.) with less computation and communication overhead of sensor node. The traditional cryptographic algorithm cannot be implemented on resource constraint sensor nodes for efficient user authentication system. Therefore, we aim to design a secure and light-weight cryptographic mechanism of user authentication and session key establishment for WSNs/IoT.

The significant contributions of our work are as follows:

- In this paper, we first discuss various security issues involved in authenticating the users of WSNs and IoT.
- We perform the security analysis of various existing protocols of user authentication for WSNs. Through security analysis, we show that the existing protocols are vulnerable to various attacks like user impersonation attack, sensor node impersonation attack, attacks based on legitimate users.
- We propose a secure and efficient protocol for authenticating the users of WSNs and IoT considering mutual authentication, session key establishment, data freshness, and confidentiality.
- Through informal security analysis, we show that our proposed protocol resists the stolen smart card, sensor node compromise, gateway node compromise, man-in-the-middle and replay attacks.
- We execute “proof of security” using random oracle model to ensure the correctness of various security features involved in our proposed protocol.
- Subsequently, we verify the proposed protocol on popular and robust security verification tool such as AVISPA and Scyther.
- We use BAN logic to determine whether exchanged messages of the proposed protocol are trustworthy and secure against eavesdropping.
- Finally, we present the comparative analysis of our proposed protocol with other existing protocols based on security and computational overhead.

The remaining portions of this paper are structured as follows: Section 2 appraises the security features and deficiencies of existing user authentication schemes. Section 3 explains the notations and cryptography procedures we used in security analysis and proposed protocol. Section 4 demonstrates the recent protocols of user authentication and their cryptanalysis. Section 5 illustrates our proposed scheme. Section 6 performs the security analysis of our proposed scheme. Section 7 shows the results of comparative study. Section 8 represents the comprehensive analysis. Section 9 concludes our research work.
2. Related Work


The proposed two-factor authentication mechanism based on user’s identity and password is generally not reliable because the user intends to choose a low-entropy password that can be easily cracked by applying simple dictionary attacks.

To improve the security feature of two-factor user authentication mechanism that are vulnerable to password guessing attacks and subject to inefficient password update procedure in WSNs, biometric-based user authentication mechanism, accompanied with user passwords and smart cards, have drawn considerable attention. In 2010, Yuan et al. [13] provided a bio-metric based scheme but it is unprotected from node capture and denial of service attack. In 2012, Yoo et al. [14] designed a scheme that provides secure session key and mutual authentication. In 2013, Xue et al. [15] designed a mutual authentication scheme based on temporal information. However, in 2014, Jiang et al. [16] revealed that Xue et al.’s scheme is susceptible to stolen smart card and privilege insider attack. In 2015, A.K. Das [17] proposed fuzzy extractor based authentication scheme which resists well known security attacks of WSNs and have more security features compared to Althobaiti et al. (2013) [18] scheme. Sharaf et al. [19] proposed (in 2016) an object authentication system in order to exploit device-specific data, known as fingerprints, to authenticate the objects associated with the IoT. In 2016, Alizadeh et al. [20] presented a comprehensive survey of authentication schemes of mobile cloud computing (MCC) to explain MCC authentication and differentiate it with that of cloud computing schemes. However, in this paper we performed the cryptanalysis of A.K.Das [17] scheme and found that it is susceptible to stolen smart card attack. Similarly, we found that Choi et al. [21] (proposed in 2016), Park et al. [22] (introduced in 2016), and Moon et al.’s [23] (proposed in 2017) schemes are also insecure against various security attacks as we have illustrated in Section 4 of this paper.
3. Notations, Assumptions and Cryptography Concepts Used

3.1. Notations

Some important notations used for design and analysis of user authentication protocol for WSNs and IoT are listed in Table 1.

Table 1. Notations used.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p, q$</td>
<td>Two large prime numbers</td>
</tr>
<tr>
<td>$F_p$</td>
<td>A finite field of characteristic $p$</td>
</tr>
<tr>
<td>$E$</td>
<td>Elliptic curve over $F_p$</td>
</tr>
<tr>
<td>$G$</td>
<td>Group of points on $E$</td>
</tr>
<tr>
<td>$P$</td>
<td>Generator point on $E$ with order $q$</td>
</tr>
<tr>
<td>$U_i$</td>
<td>$i$th User of WSNs/IoT</td>
</tr>
<tr>
<td>$ID_{U_i}$</td>
<td>The identity of $U_i$</td>
</tr>
<tr>
<td>$SN_j$</td>
<td>$j$th sensor node</td>
</tr>
<tr>
<td>$PW_{U_i}$</td>
<td>Password of $U_i$</td>
</tr>
<tr>
<td>$ID_{SN_j}$</td>
<td>The identity of $SN_j$</td>
</tr>
<tr>
<td>$SC_i$</td>
<td>$U_i$’s Smart card</td>
</tr>
<tr>
<td>$GWN$</td>
<td>The gateway node</td>
</tr>
<tr>
<td>$x$</td>
<td>Random number</td>
</tr>
<tr>
<td>$h(.)$</td>
<td>Secure hash function</td>
</tr>
<tr>
<td>$Gen(.)$</td>
<td>Fuzzy generator function</td>
</tr>
<tr>
<td>$Rep(.)$</td>
<td>Fuzzy reproduction function</td>
</tr>
<tr>
<td>$Z^+$</td>
<td>Set of positive integers</td>
</tr>
<tr>
<td>$B_i$</td>
<td>Bio-metric information of $U_i$</td>
</tr>
<tr>
<td>$T$</td>
<td>The error tolerance limit</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>Maximum transmission delay</td>
</tr>
<tr>
<td>$T', T'', T'''$</td>
<td>Current time at $GWN$, $SN_j$, and $U_i$</td>
</tr>
<tr>
<td>$Enc_k[s]$</td>
<td>Symmetric encryption of message $s$ using key $k$</td>
</tr>
<tr>
<td>$Dec_k[E_k[s]]$</td>
<td>Symmetric decryption of $E_k[s]$ using key $k$</td>
</tr>
<tr>
<td>$\parallel$</td>
<td>Concatenation operator</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>Bitwise XOR operator</td>
</tr>
<tr>
<td>$\times$</td>
<td>Point multiplication operator of $E$</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>Adversary</td>
</tr>
</tbody>
</table>

3.2. Assumptions

- Sensor node may not fix up with tamper-resistant hardware and if a node is captured by an adversary, all the prominent and confidential information stored in its memory can be accessed by the adversary. If the sensor nodes are tamper-resistant the adversary can know the information stored in the memory by measuring the power consumption of the captured sensor nodes.
- The base station or the gateway node is the trusted entity, and it works both as an authentication as well as a key distribution center.
- The adversary $\mathcal{A}$ can intercept the public communication channel, inject packets and replay the previously transmitted packets.
- The adversary $\mathcal{A}$ can capture the smart card $SC_i$ of user $U_i$ and it can extract the sensitive information stored in the card through simple and differential power analysis techniques [24].
- We assume that the WSNs and IoT consist of few users (with smart card which can be captured or stolen by the adversary $\mathcal{A}$), hundreds of sensor nodes (it can be captured by $\mathcal{A}$) and the trusted gateway node.
The processed data from the sensor nodes are gathered periodically at the gateway node GWN. The gathered data may not always be real-time and fresh at GWN. Therefore, the authentic user should be allowed to access the data directly from the sensor node SNj to make quick decision for secure and real-time applications of WSNs and IoT.

3.3. Cryptography Concepts Used

Some basic cryptography concepts used in the security analysis of existing protocols and also in our proposed protocol are defined as follows:

**Definition 1.** Secure Hash Function [25]: A function $h : \text{In} \rightarrow \text{Out}$, with a binary string $s \in \text{In} \{0, 1\}^*$ of arbitrary length as input and a binary string $d \in \text{Out} \{0, 1\}^m$ of fixed length $m$ as an output, is a secure hash function if the following conditions holds:

- $A$’s advantage to find the collision $\text{Adv}_A^h(t_1) = \text{Pr}(s \neq s', h(s) = h(s'))$ and
- $\text{Adv}_A^h(t_1) \leq \tau$, for any sufficiently small $\tau > 0$.

where $(s, s') \leftarrow_R$ indicates that the pair $(s, s')$ is randomly chosen by $A$ and $\text{Pr}$ represents the probability of the event $(s, s') \leftarrow_R A$ with execution time $t_1$.

**Definition 2.** Secure Encryption Scheme [25]: For any probabilistic, polynomial time adversary $A$, an encryption algorithm $E$ is said to be IND-CPA (indistinguishability of encryption and chosen plaintext attack) secure if $\text{Adv}_{\text{Enc}, A}^{\text{IND-CPA}}(t_2) = 2\text{Pr}[A \leftarrow_O (b_0, b_1 \leftarrow A); \tau \leftarrow_R 0, 1; \gamma \leftarrow_R \{0, 1\}; A(\gamma) = \tau] - 1$ denotes the advantage function of $A$ and $\tau \leftarrow_R 0, 1$ denotes that the bit $\tau$ is a randomly chosen from set $\{0, 1\}$. $t_2$ denotes the execution time.

**Definition 3.** Elliptic Curve Diffie-Hellman [26]: If $p > 3$ be a prime number, the elliptic curve $E_p(a, b)$ considered over the finite field $\mathbb{Z}_p^*$ is represented by the solutions $(x, y) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ of the equation $y^2 = x^3 + ax + b$, along with a point $O$ of infinity, where $4a^3 + 27b^2 \neq 0 \text{ mod } p$. If $P$ be a generator or a base point of a cyclic subgroup $G$ of the elliptic curve $E_p(a, b)$ considered over the finite field $\mathbb{F}_p$, i.e., $G = \langle P \rangle$, the elliptic curve Diffie-Hellman (ECDH) key exchange can be described as follows:

Initially, $U_i$ and $SN_j$ agree on a generator point $P$ and choose their private key as $r_{U_i}$ and $r_{SN_j}$ respectively. Afterwards, they construct and exchange their public keys as $X_{U_i} = r_{U_i} \times P$ and $Y_{SN_j} = r_{SN_j} \times P$. Finally, $U_i$ and $SN_j$ calculate the common secret key as $r_{U_i} \times (r_{SN_j} \times P)$ and $r_{SN_j} \times (r_{U_i} \times P)$ respectively. Where $r_{U_i} \times (r_{SN_j} \times P) = r_{SN_j} \times (r_{U_i} \times P)$ and it is intractable to find $r_{U_i}$ and $r_{SN_j}$ for an adversary $A$ who knows $X_{U_i}$ and $Y_{SN_j}$, i.e.,

The advantage in finding $r_{U_i}$ is defined by $\text{Adv}_A^{\text{ECDH}}(t_3) = \text{Pr}[(r_{U_i}, P) \leftarrow_R A : X_{U_i} = r_{U_i} \times P]$. Where $\text{Adv}_A^{\text{ECDH}}(t_3) \leq \tau$, for any sufficiently small $\tau > 0$ and $(r_{U_i}, P) \leftarrow_R A$ means the pair $(r_{U_i}, P)$ is randomly selected by $A$ with execution time $t_3$, such that $X_{U_i} = r_{U_i} \times P$.

**Definition 4.** Fuzzy Extractor for user authentication: Fuzzy extractor [27] is a cryptography mechanism for securely authenticating a user using bio-metric credentials. Suppose a finite set $M$ is a metric space with a distance function $\text{dis}$ along with an error tolerance limit $\mathcal{T}$ calculated using error correction codes for any particular distance metric (hamming distance, set difference metric, edit distance metric etc.) such that:

- $\text{dis} : M \times M \rightarrow \mathbb{R}^* = [0, \infty)$.
- $\text{dis}(B_i, B_j) = 0 \text{ iff } B_i = B_j$,
- $\text{dis}(B_i, B_j) = \text{dis}(B_i', B_j')$,
- $\text{dis}(B_i, B_j') \leq (\text{dis}(B_i, B_j') + \text{dis}(B_j', B_j''))$, where $B_i, B_j', B_j'' \in M$.

The fuzzy extractor consists of two randomized operations i.e., Generator (Gen) and Reproduction (Rep) with the following characteristics:
• The Gen() operation takes a bio-metric credential \( B_i \in M \) of user \( U_i \) as an input and produces outputs—a secret string \( \sigma_i \in \{0,1\}^l \) and a public accessory string \( \tau_i \in \{0,1\}^*, \) i.e., \( \text{Gen}(B_i) = (\sigma_i, \tau_i) \)

• The Rep() operation takes a noisy bio-metric credential \( B'_i \in M \) of user \( U_i \) and the public accessory string \( \tau_i \) as an input and reproduces the secret string \( \sigma_i \in \{0,1\}^l \) as an output i.e., \( \text{Rep}(B'_i, \tau_i) = \sigma_i \) if and only if \( \text{dis}(B_i, B'_i) \leq T. \)

4. Review and Cryptanalysis of Various Recent Schemes of User Authentication for WSNs

In this section, we concisely review and present the security analysis of the various recently proposed user authentication protocols of WSNs. The security analysis performed in this section illustrates that the existing protocols have various security vulnerability based on the logical proofs and the assumptions considered in the Section 3.2 of this paper. This section provides an awareness of what needs to be fixed and how the user authentication protocol should be design to withstand the miscellaneous attacks incorporated into the WSNs/IoT.

4.1. Review of A.K.Das’s Scheme

A.K.Das [17] performed the security analysis of Althobaiti et al.’s [18] scheme and proposed an improved scheme of user authentication using the fuzzy extractor in order to resist node capture attack, impersonation attack, man-in-the-middle attack. A.K.Das [17] proposed a novel approach (considering the resource constraints of sensor node) for bio-metric based user authentication using the fuzzy extractor. For evaluating the security features of A.K.Das’s Scheme, the user registration phase of Das’s scheme is described in the following Step DR1, Step DR2, Step DR3 and the authentication-key agreement phase is summarized in the Steps DA1, Step DA2, Step DA3 based on the notations of Table 1. We summarize the user registration, authentication and key agreement phase of A.K.Das’s scheme in Tables 2 and 3 respectively.

**Step DR1:** The user \( U_i \) inputs \( ID_{U_i}, PW_{U_i} \) and \( B_i \) and generates 1024 bit random number \( K. \) Subsequently, \( U_i \) calculates \( RPW_i = h(ID_{U_i} || K || PW_{U_i}) \) and selects a key \( e_k_i. \) Then, \( U_i \) transmits \( \langle ID_{U_i}, RPW_i, e_k_i \rangle \) to GWN using secure communication channel.

**Table 2. User registration phase of A.K.Das’s scheme.**

<table>
<thead>
<tr>
<th>Step 1: For User ( (U_i) )</th>
<th>Step 2: For Gateway ( (GWN) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user ( U_i ) inputs ( ID_{U_i}, PW_{U_i} ) ( ) and ( B_i ) ( ) and generates 1024 bit random number ( K. ) Subsequently, ( U_i ) calculates ( RPW_i = h(ID_{U_i}</td>
<td></td>
</tr>
</tbody>
</table>

**Step 3: For User \( (U_i) \)**

\( U_i \) evaluates \( \text{Gen}(B_i) = (\sigma_i, \tau_i) \), \( f^*_i = f_i \oplus h(ID_{U_i} || \sigma_i || K), \) \( r_i = h(ID_{U_i} || \sigma_i) \oplus K, \) \( e_i = h(ID_{U_i} || RPW_i || \sigma_i), \) and \( BE_i = h(ID_{U_i} || \sigma_i) \oplus e_k_i. \) Then, \( U_i \) replaces \( f_i \) with \( f^*_i \) in \( SC_i \). Finally, \( U_i \) stores \( e_i, \tau_i, BE_i, r_i \) into \( SC_i \).
Table 3. Login, authentication and key sharing phase of A.K. Das’s scheme.

<table>
<thead>
<tr>
<th>Step 1: For User ($U_i$)</th>
<th>Step 2: For Gateway ($GWN$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The registered user $U_i$ inserts his/her smart card $SC_i$ into card reader device and provides the $ID_{U_i}$, secret $PW_{U_i}$, $B_i$. Then, Evaluates $r_i' = \text{Rep}(B_i, \tau_i), K_i' = r_i \oplus h({ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td>$\text{else}$</td>
<td>$GWN$ aborts this phase</td>
</tr>
<tr>
<td>$U_i$ transmits $\langle ID_{U_i}, \text{req} \rangle$ to $GWN$ via Public Channel</td>
<td>$\text{else}$</td>
</tr>
<tr>
<td>$\text{else}$</td>
<td>$\text{Reject } U_i$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: For User ($U_i$)</th>
<th>Step 4: For Gateway ($GWN$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>After receiving the Random challenge $R$, $U_i$ evaluates $e_k_i = BE_i \oplus h(ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td>$\text{if } T_1 \text{ and } R \text{ are valid then}$</td>
<td>$\text{if } T_3 \text{ is valid then}$</td>
</tr>
<tr>
<td>$GWN$ computes $f_i^* = h(ID_{U_i} \oplus h(X_s)), f_i^{**} = h(ID_{SN_j}</td>
<td></td>
</tr>
<tr>
<td>$\text{else}$</td>
<td>$\text{else}$</td>
</tr>
<tr>
<td>$\text{Reject } U_i$</td>
<td>$\text{Reject } U_i$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5: For Sensor Node ($SN_j$)</th>
<th>Step 6: For User ($U_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SN_j$ Retrieves $(ID_{U_i}, ID_{SN_j}, T_1, T_2, f_i^{<strong>})$ as $(ID_{U_i}', ID_{SN_j}', T_1', T_2', f_i^{</strong>}')$. $SN_j$ Evaluates the session key $SK_i = h(f_i^*</td>
<td></td>
</tr>
<tr>
<td>$\text{if } T_2 \text{ and } ID_{U_i} \text{ are valid then}$</td>
<td>$GWN$ sends $\langle SC_i \rangle$ to $U_i$ using secure communication channel.</td>
</tr>
<tr>
<td>$SN_j$ sends $h(SK_i)$ to $U_i$</td>
<td>$\text{else}$</td>
</tr>
<tr>
<td>$\text{else}$</td>
<td>$\text{Reject } U_i$</td>
</tr>
<tr>
<td>$\text{Reject } U_i$</td>
<td>$\text{Reject } U_i$</td>
</tr>
</tbody>
</table>

Step DR2: After receiving the message $\langle ID_{U_i}, RPW_{i}, e_k_i \rangle$, the gateway node $GWN$ generates 1024 bit key $X_s$, evaluates $f_i = h(ID_{U_i} \oplus h(X_s))$ and stores $(h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot), f_i, T)$ into $SC_i$. Then, $GWN$ sends $\langle SC_i \rangle$ to $U_i$ using secure communication channel.

Step DR3: After receiving $SC_i$, the user $U_i$ evaluates $\text{Gen}(B_i) = (\sigma_i, \tau_i), f_i^* = f_i \oplus h(ID_{U_i}||\sigma_i)||K_i$, $r_i = h(ID_{U_i}||\sigma_i) \oplus K_i, e_i = h(ID_{U_i}||RPW_{i}||\sigma_i)$, and $BE_i = h(ID_{U_i}||\sigma_i) \oplus ek_i$. Finally, $U_i$ replaces $f_i$ with $f_i^*$ in $SC_i$ and stores $e_i, \tau_i, BE_i, r_i$ into $SC_i$.
Step DA1: The registered user \( U_i \) inserts his/her smart card \( SC_i \) into the card reader device and provides the \( ID_{U_i} \), secret \( PW_{U_i} \), bio-metric information \( B_i \). Then, evaluates \( e_i' = \text{Rep}(B_i, \tau_i) \), \( K' = r_i \oplus h(\text{ID}_{U_i} || e_i') \), \( RPW_i' = h(\text{ID}_{U_i} || PW_{U_i} || K') \), \( e_i'' = h(\text{ID}_{U_i} || RPW_i' || e_i') \). If \( e_i'' = e_i' \), \( U_i \) transmits \( \langle \text{ID}_{U_i}, \text{req} \rangle \) to GWN via public communication channel. Otherwise, \( U_i \) aborts this phase.

Step DA2: After receiving the message \( \langle \text{ID}_{U_i}, \text{req} \rangle \), GWN verifies the message. If \( \text{ID}_{U_i} \) is valid, GWN sends a Random challenge \( R \) to \( U_i \) via public communication channel. Otherwise, GWN aborts this phase.

Step DA3: After receiving the Random challenge \( R \), \( U_i \) evaluates \( e_k = BE_i \oplus h(\text{ID}_{U_i} || e_i') \).

Finds the current time-stamp \( T_1 \). Then, \( U_i \) transmits \( \langle \text{Enc}_{e_k}(R, T_1, \text{ID}_{SN_j}) \rangle \) to GWN via public communication channel.

Step DA4: GWN evaluates \( R, T_1, \text{ID}_{SN_j} \) using decryption operation based on key \( e_k \). If \( T_1 \) is fresh and \( R \) is valid, GWN computes \( f_i^* = h(\text{ID}_{U_i} \oplus h(X_i)), f_i^{**} = h(\text{ID}_{SN_j} || f_i^*) \), finds the current time-stamp \( T_2 \) and computes \( Y_j = \text{Enc}_{K_f}[\text{ID}_{U_i}, \text{ID}_{SN_j}, T_1, T_2, f_i^{**}] \). Finally, GWN transmits \( \langle \text{ID}_{U_i}, Y_j \rangle \) to \( SN_j \) via public communication channel. Otherwise, GWN aborts this phase immediately.

Step DA5: \( SN_j \) retrieves \( \langle \text{ID}_{U_i}, \text{ID}_{SN_j}, T_1, T_2, f_i^{**} \rangle \) as \( \langle \text{ID}_{U_i}', \text{ID}_{SN_j}', T_1', T_2', f_i^{**} \rangle \) using decryption operation on \( \langle \text{ID}_{U_i}, Y_j \rangle \) based on key \( K_f \). If \( T_2'' \) is fresh and \( \text{ID}_{U_i} \) is valid, \( SN_j \) finds the current time-stamp \( T_3 \) and evaluates the session key \( SK_{ij} = h(f_i' || \text{ID}_{U_i} || \text{ID}_{SN_j} || T_1', T_3) \). Then, \( SN_j \) sends \( h(SK_{ij}), T_3 \) to \( U_i \) via public communication channel and stores \( SK_{ij} \) in its memory. Otherwise, \( SN_j \) aborts this phase immediately. Finally, \( SN_j \) stores \( SK_{ij} \) in its memory.

Step DA6: If \( T_3 \) is fresh, the user \( U_i \) computes \( f_i'' = f_i' \oplus h(\sigma_i || \text{ID}_{U_i} || K') \), \( f_i''' = h(\text{ID}_{SN_j} || f_i''), SK_{ij}' = h(f_i''' || \text{ID}_{SN_j} || T_1 || T_3) \). If \( h(SK_{ij}') = h(SK_{ij}), U_i \) establishes the session key \( SK_{ij}' \) with the sensor node \( SN_j \). Otherwise, \( U_i \) aborts this phase immediately.

4.2. Cryptanalysis of A.K.Das’s Scheme

In this section, we perform the cryptanalysis of the A.K.Das’s scheme and found that A.K.Das’s scheme is also vulnerable. The vulnerabilities involve in A.K.Das’s scheme are elaborated in the following subsection:

4.2.1. Stolen Smart Card Attacks

The adversary \( A \) ascertains the value of \( \{ \tau_i, e_i, r_i, BE_i, f^*, h(\cdot), Gen(\cdot), Rep(\cdot), T \} \) from stolen \( SC_i \) by measuring the power consumption of smart card [24]. Then, \( A \) computes: \( BE_i \oplus r_i = [h(\text{ID}_{U_i} || \sigma_i) \oplus \text{K}] \oplus h(\text{ID}_{U_i} || \sigma_i) \oplus ek_i = K \oplus ek_i \).

Afterwards, the adversary \( A \) find out the value of \( K \) and \( ek_i \) by implementing one of the following three mechanisms:

1. Derives the value of \( K \) and \( ek_i \) using the frequency analysis of stream cipher \( BE_i, r_i \) and \( BE_i \oplus r_i \).

2. Eavesdrops \( R \) and \( E_{ek_i}(R, T_1, \text{ID}_{SN_j}) \) and implements the known plain text attack to find out the value of \( ek_i \). Thereafter, \( A \) find out the value of \( K = ek_i \oplus (K \oplus ek_i) \).

3. Steals the bio-metric information \( B_i \) of \( U_i \) (where \( d(B_i, B_i') \leq T \) ) and find out the value of \( \sigma_i = \text{Rep}(B_i', \tau_i) \). Eavesdrops the value of \( \text{ID}_{U_i} \) from public communication channel and then evaluates the value of \( ek_i = BE_i \oplus h(\text{ID}_{U_i} || \sigma_i), K = r_i \oplus h(\text{ID}_{U_i} || \sigma_i) \). It is possible, because \( ek_i \) is not password \( PW_{U_i} \) protected.

Subsequently, \( A \) chooses its own identity \( \text{ID}_{A} \), password \( PW_{A} \), bio-metric information \( B_A \) and computes:

\[
RPW_A = h(\text{ID}_{A} || \text{K} || PW_A), Gen(B_A) = (\sigma_A, \tau_A), e_A = h(\text{ID}_{A} || RPW_A || \sigma_A), r_A = h(\text{ID}_{A} || \sigma_A) \oplus K \oplus BE_A = h(\text{ID}_{A} || \sigma_A) \oplus ek_i.
\]

Finally, \( A \) replaces the information \( \{ \tau_i, e_i, r_i, BE_i, f^*, h(\cdot), Gen(\cdot), Rep(\cdot), T \} \) of \( SC_i \) with \( \{ \tau_A, e_A, r_A, BE_A, f^*, h(\cdot), Gen(\cdot), Rep(\cdot), T \} \) respectively.
The login phase of the adversary $\mathcal{A}$ is as follows:

- $\mathcal{A}$ inserts $SC_i$ and inputs $ID_A, PW_A$ and imprints $B_A$.
- $\mathcal{A}$ computes $\sigma'_A = Rep(B_A, \tau_A)$, $K' = r_A \oplus h(ID_A \parallel \sigma'_A), RPW'_A = h(ID_A \parallel PW_A \parallel K')$ and $\sigma'_A = h(ID_A \parallel RPW'_A \parallel \sigma'_A)$. Then, it verifies if $\sigma'_A = \sigma_A$.

Afterwards, $U_i$ sends the login message $\langle ID_A, req \rangle$ to GWN via a public channel. However, the adversary $\mathcal{A}$ intercepts the message $\langle ID_A, req \rangle$ and replaces $\langle ID_A, req \rangle$ with $\langle ID_{U_i}, req \rangle$.

Authentication and key agreement phase for the adversary $\mathcal{A}$ is illustrated as follows:

- Since $ID_{U_i}$ is valid, therefore GWN generates a random challenge $R$ and send it to $A$.
- $\mathcal{A}$ selects the login sensor node $SN_j$ and sends $\langle E_{ek}(R, T_1, ID_{SN_j}) \rangle$ to GWN.
- After receiving $\langle E_{ek}(R, T_1, ID_{SN_j}) \rangle$, GWN decrypt it using $ek_j$ and verifies the validity of $T_1$ and $R$. Subsequently, GWN computes $f_i^* = h(ID_U \oplus h(X_i)), f_i^{**} = h(ID_{SN_j} \parallel f_i^*), Y_i = E_{ek_i}(ID_{U_i}, ID_{SN_j}, T_1, T_2, f_i^{**})$ and finally sends $\langle ID_{U_i}, Y_i \rangle$ to the sensor node $SN_j$.
- After receiving $\langle ID_{U_i}, Y_i \rangle$, $SN_j$ computes $SK_{ij} = h(f_i^{**} \parallel ID_{U_i} \parallel ID_{SN_j} \parallel T_1'' \parallel T_3)$ and sends $h(SK_{ij}), T_3$ to $\mathcal{A}$.
- Then, $\mathcal{A}$ computes $f_i' = f_i^* \oplus h(\sigma_i' \parallel ID_{U_i} \parallel K')$ using $ID_{U_i}$, stolen bio-metric and evaluated $K$. It is possible because $f_i^*$ has no password protection.
- Finally, $\mathcal{A}$ computes $f_i'' = h(ID_{SN_j} \parallel f_i')$ and the session key $SK_{ij} = h(f_i'' \parallel ID_{U_i} \parallel ID_{SN_j} \parallel T_1'' \parallel T_3)$ shared with $SN_j$.

4.3. Review of Choi et al.’s Scheme

Choi et al. [21] performed the security analysis of Yoon and Kim’s [28] protocol and proposed an improved protocol (considering the resource constraints of sensor node of WSNs) of user authentication using the fuzzy extractor and biometric information. The Choi et al.’s protocol solves the problems of biometric recognition inaccuracy, user verification difficulty, lack of anonymity, perfect forward secrecy, session key revelation by the GWN, DoS attack, and a revocation problem. In this scheme, the gateway node GWN originates master keys, $x$ and $y$, and allocates $h(ID_{SN_j} \parallel y)$ to the sensor node $SN_j$. The registration phase of this scheme is summarized in Step CR1, Step CR2 and Step CR3. The authentication, and session key establishment phase is summarized in Table 4.

**Step CR1**: The user $U_i$ inputs his/her identity $ID_{U_i}$, biometric information $B_i$ and computes: $(\sigma_i, \tau_i) = Gen(B_i), A_i = h(\sigma_i)$. Then, $U_i$ transmits $\langle ID_{U_i}, A_i \rangle$ to GWN via secure communication channel.

**Step CR1**: After receiving the message $\langle ID_{U_i}, A_i \rangle$, the gateway node GWN generates 1024 bit secret key $x$ and computes $M_{U_i} = h(ID_{U_i} \parallel x) \oplus A_i, N_{U_i} = h(ID_{U_i} \parallel x) \oplus A_i, V_{U_i} = h(ID_{U_i} \parallel A_i)$. Then, GWN stores $\langle ID_{U_i}, M_{U_i}, N_{U_i}, V_{U_i}, h(\cdot) \rangle$ into smart card $SC_i$. Finally, GWN sends the smart card $SC_i$ to the user $U_i$.

**Step CR2**: After receiving the smart card $SC_i$, the user $U_i$ stores $\tau_i$ into $SC_i$.

4.4. Cryptanalysis of Choi et al.’s Scheme

In this section, we perform the cryptanalysis of the Choi et al.’s scheme and found that Choi et al.’s scheme is also vulnerable. The vulnerabilities involve in this scheme are elaborated in the following subsection:
Table 4. Authentication and session key establishment phase of Choi et al. protocol.

<table>
<thead>
<tr>
<th>Step 1: For User (U_i)</th>
<th>Step 2: For Gateway (GWN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The registered user U_i inputs ID_{U_i}, B_i' and computes c_i' = Rep(B_i', t_i), A_i' = h(c_i'), V_i' = h(ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td>( U_i ) generates random number r_i ( X_i = r_i \times P, D_i = M_i \oplus A_i' ), ( h(x</td>
<td></td>
</tr>
<tr>
<td>then, ( U_i ) constructs a message ( M_1 = (AID_i, X_i, C_i, T_i, W_i) )</td>
<td>GWN computes ( ID_{U_i} = AID_i \oplus h(h(x</td>
</tr>
<tr>
<td>Finally, ( U_i ) transmits ( M_1 ) to GWN</td>
<td>GWN finds its current timestamp ( T_g ) and computes ( k_g = h(h(SID_g</td>
</tr>
<tr>
<td>else ( GWN ) aborts this phase.</td>
<td>Then, GWN construct the message ( M_2 = (AID_g, C_g, T_g, W_g) ). Finally, GWN transmits ( M_2 ) to SN_j</td>
</tr>
<tr>
<td>else ( GWN ) aborts this phase.</td>
<td></td>
</tr>
<tr>
<td>else ( GWN ) aborts this phase.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: For Sensor Node (SN_j)</th>
<th>Step 4: For User (U_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>if (T'' - T_g) \leq \Delta T then</td>
<td>if (T''' - T_s) \leq \Delta T then</td>
</tr>
<tr>
<td>if ( W_g = h(h(SID_g</td>
<td></td>
</tr>
<tr>
<td>( SN_j ) computes ( k''_g = h(h(SID_g</td>
<td></td>
</tr>
<tr>
<td>( AID_j</td>
<td></td>
</tr>
<tr>
<td>Generates random number ( r_s ) ( k_{SU} = r_s \times X_i' ), ( Y_i = r_s \times P ) Computes ( sk = h(AID_j</td>
<td></td>
</tr>
<tr>
<td>Sk</td>
<td>else ( GWN ) aborts this phase.</td>
</tr>
<tr>
<td>else ( GWN ) aborts this phase.</td>
<td></td>
</tr>
<tr>
<td>else ( GWN ) aborts this phase.</td>
<td>else ( GWN ) aborts this phase.</td>
</tr>
</tbody>
</table>
4.4.1. Attack Based on Legitimate User

In this scheme, a legitimate user $U_l$ can be an adversary $U_A$, because $U_l$ can find out the hashed master key $h(x||y)$ and then it can derive the secret information of user $U_i$ as follows:

- $U_A$ inputs $ID_{U_A}$, imprints $B'_A$, computes $c'_A = Rep(B'_A, \tau_A)$, $A'_A = h(c'_A)$, $V'_A = h(ID_A||A'_A)$ and finally verifies $V_A = V'_A$.
- If verification succeeds, $U_A$ generate random number $r_A$, and computes $X_A = r_A \times P$, $D_A = M_A \oplus A'_A$, $h(x||y) = N_A \oplus A'_A$.
- $A$ intercepts the message $M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle$ of $U_i$ and find out: $ID_{U_l} = AID_i \oplus h(h(x||y)||T_i)$.

Therefore, we find that Choi et al. scheme does not provide user anonymity i.e., an adversary $A$ can compute user $U_i$’s identification $ID_{U_l}$. However, Choi et al. claimed that their protocol provides user anonymity.

- Furthermore $A$ intercepts the cipher text $C_i = E_k(\langle ID_{U_l}, X_i \rangle)$ and derives the plain-text $(ID_{U_l}, X_i)$, therefore Choi et al. scheme is vulnerable to known plain-text attack.

4.4.2. User Impersonation Attack

An adversary $A$ with an stolen smart card $SC_i$ can impersonate a legitimate user $U_i$ as follows:

- $A$ extracts $\langle ID_{U_l}, M_{U_l}, N_{U_l}, V_{U_l}, h(\cdot), \tau_i \rangle$ from the smart card $SC_i$ of the user $U_i$ and computes $A^*_i = N_i \oplus h(x||y)$, $V^*_i = h(ID_1||A^*_i)$ and verify the computed $V^*_i$ with the stored $V_i$.
- $A$ generates a random number $r_A$, calculates $X_A = r_A \times P$, $D^*_A = M_i \oplus \tau^*_i$. Find out the current timestamps $T_A$, computes $k_A = h(D^*_A||T_A)$, $C_A = E_k_A(ID_{U_l}||X_A)$, $AID_A = ID_{U_l} \oplus h(h(x||y)||T_A)$, $W_A = h(h(x||y)||AID_A||X_A||C_A||T_A)$.
- $A$ sends the message $M_A^1 = \langle AID_A, X_A, C_A, T_A, W_A \rangle$ to $GWN$. Subsequently, $A$ establishes the session key $sk = h(AID_A\oplus r_A \times r_s \times P)$ with $SN_j$ using Steps 2–4 of authentication and session key establishment phase of Choi et al. protocol.

4.5. Review of Park et al.’s Scheme

Park et al. [22] performed the security analysis of Chang et al.’s [29] scheme. Then, Park et al. proposed an improved scheme of user authentication using the fuzzy extractor and biometric information in order to provide forward secrecy, accurate password update phase and resist off-line password guessing attacks. In this scheme the gateway node $GWN$ originates master keys, $x$ and $y$, and allocates a key $h(ID_{SN_l}||y)$ to the sensor node $SN_j$. Afterwards, the scheme follows the registration, login and authentication phase as shown in Tables 5 and 6.

4.6. Cryptanalysis of Park et al.’s Scheme

In this section, we perform the cryptanalysis of the Park et al.’s scheme and found that Park et al.’s scheme is also vulnerable and it has the following security vulnerabilities:

4.6.1. Sensor Node Impersonation Attack

According to Park et al., to impersonate a sensor node $SN_j$, an adversary $A$ need to have the key $k_{GWN} = h(h(ID_{SN_l}||y)||T_{GWN})$. Although, an adversary $A$ can impersonate the sensor node $SN_j$ without having $k_{GWN}$ with the help of following steps:

- The adversary $A$ intercepts the message $M_1 = \langle AID_{U_l}, X_{U_l}, C_{U_l}, T_{U_l}, W_{U_l} \rangle$, $M$ and $M_2 = \langle AID_{GWN}, C_{GWN}, T_{GWN}, W_{GWN} \rangle$.
- Then, $A$ generates a random number $r_A$, finds current times-stamp $T_A$ and computes: $K_{AU_l} = r_A \times X_{U_l}$, $Y_A = r_A \times P$, $sk = h(AID_{U_l}||K_{AU_l}||T_A)$, $RM = Query response$ and $V_A = h(AID_{U_l}||X_{U_l}||Y_A||RM||T_A)$. 
• Afterwards, \( A \) sends \( M_3 = \langle RM, Y_A, V_A, T_A \rangle \) to \( U_i \).

• After receiving \( M_3 \), \( U_i \) computes: \( V_A = h(\text{AID}_U || X_U || Y_A || RM || T_A) \). If \( V_A^* = V_A \), \( A \) computes \( K_{U,A} = r_{U_i} \times Y_A, sk = h(\text{AID}_U || K_{U,A} || T_A) \).

Therefore, the adversary \( A \) succeeds in impersonating the sensor node \( SN_j \) and establishing the session key \( sk \) with the user \( U_i \).

### 4.6.2. User Impersonation Attack

In Park et al.'s scheme, a legitimate user \( U_k \) can be an adversary \( U_A \) to impersonate the user \( U_i \) because \( U_k \) can find out the hashed master key \( h(x||y) \) and then it can derive the secret information of user \( U_i \) as follows:

• First, the adversary \( A \) extract the information \( \langle V_{U_i}, N_{U_i}, C_{U_i}, h(.), P_{U_i} \rangle \) from the smart card.

• Then, \( A \) imprints its bio-metric information \( B'_i \) and computes: \( \sigma'_i = \text{Rep}(B'_i, P_{U_i}) \) and \( A'_{U_i} = h(\sigma'_i) \), \( M_{U_i} = N_{U_i} \oplus A'_{U_i} \).

• Afterwards, \( A \) generates random number \( r_{A_i} \), selects an identity \( ID_{U_i} \) and computes: \( X_A = X_{U_i} \times P, \text{AID}_U = ID_{U_i} \oplus h(M_A||T_A) \) and \( W_A = W_{U_i} = h(M_A||ID_{U_i}||X_A||T_A) \). Finally, \( A \) sends \( M_1 = \langle \text{AID}_U, X_A, C_{U_i}, T_{U_i}, W_{U_i} \rangle \) to GWN.

• After receiving \( M_1 \), if \( T' - T_{U_i} \leq \Delta T \), GWN computes \( A'_{U_i} \oplus W_{U_i} = \text{Dec}_x(C_{U_i}) \), \( M'_{U_i} = h(x||y||A'_{U_i}) \), \( \text{ID}'_{U_i} = \text{AID}_U \oplus h(M'_{U_i}||T_{U_i}) \), \( W'_{U_i} = h(M_{U_i}||\text{ID}'_{U_i}||X'_{U_i}||T_{U_i}) \).

• If \( W_{U_i} = W'_{U_i} \), the GWN finds the current time stamp \( T_{GWN} \) and computes: \( k_{GWN} = h(h(ID_{SN}||y)||T_{GWN}) \), \( C_{GWN} = Enc_{GWN}(\text{AID}_U||X_{U_i}) \), \( W_{GWN} = h(h(ID_{SN}||y)||\text{AID}_U||C_{GWN}||T_{GWN}) \).

• Finally, GWN sends \( M_2 = \langle \text{AID}_{U_k}, W_{GWN}, C_{GWN}, T_{GWN} \rangle \) to \( SN_j \).

• After receiving \( M_2 \), if \( T'' - T_{GWN} \leq \Delta T \) and \( W_{GWN} = h(h(ID_{SN}||y)||\text{AID}_U||C_{GWN}||T_{GWN}) \), \( SN_j \) computes: \( k'_{SN_j} = h(h(ID_{SN}||y)||T_{GWN}) \) and \( \text{AID}'_{SN_j} = h(\text{AID}_{SN_j}||X_{SN_j}||T_{SN_j}) \).

• If \( \text{AID}_{U_k} = \text{AID}'_{SN_j} \), \( SN_j \) generates a random number \( r_{SN_j} \) and computes: \( K_{SU} = r_{SN_j} \times X'_{SN_j}, Y_{U_i} = r_{SN_j} \times P, sk = h(\text{AID}_U || K_{SU} || T_{SN_j}), RM = \text{Query Response}, V_{SN_j} = h(\text{AID}_U || X_{U_i} || Y_{U_i} || RM || T_{SN_j}) \).

• Then, \( SN_j \) sends \( M_3 = \langle RM, Y_{U_i}, V_{SN_j}, T_{SN_j} \rangle \) to the adversary \( A \).


Table 6. \(U_i\)'s authentication and session key sharing phase of Park et al. protocol.

<table>
<thead>
<tr>
<th>Step 1: For User ((U_i))</th>
<th>Step 2: For Gateway ((GWN))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U_i) inserts the smart card (SC_i), inputs (ID_{U_i}) and imprints (B'<em>i). Then, computes (\sigma'</em>{U_i} = R(ep(B'<em>i, \tau</em>{U_i}), A'<em>{U_i} = h(\sigma'</em>{U_i}), V'<em>{U_i} = h(ID</em>{U_i}</td>
<td></td>
</tr>
<tr>
<td>else Abort this phase.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: For Sensor Node ((SN_j))</th>
<th>Step 4: For User ((U_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>if ((T'' - T_{GWN}) \leq \Delta T) then (k_{GWN}' = h(h(ID_{SN_j}</td>
<td></td>
</tr>
<tr>
<td>else Abort this phase.</td>
<td>else Abort this phase.</td>
</tr>
</tbody>
</table>
which is known only to the GWN. The phase are described in following Steps R1, R2, R3 and summarized in Table 7 (using Steps 1–3).

5. Proposed Protocol

In our proposed protocol, we consider that the WSNs and IoT consist of few users (with the smart card which can be captured or stolen by the adversary A), hundreds of sensor nodes (these nodes can be captured by A) and trusted gateway node. Considering these entities, we design the protocol which consists of four critical components (i) Set-up before the deployment of WSNs/IoT (ii) Registration of U_i by the GWN (iii) U_i’s authentication and session key establishment phase (iv) U_i’s credentials update phase.

5.1. Set-Up before the Deployment of WSNs/IoT

In this phase, we select a high-performance and trusted computing node as a gateway GWN. The GWN assigns a unique identity ID_SN_j to each sensor node SN_j and loads a unique secret key K_{GSN_j} = h(ID_{SN_j}||K_{GWN}) into the memory of SN_j.

5.2. Registration of U_i by the GWN Using Secure Communication Channel

In this phase, a legitimate user U_i sends the hashed secret credential to GWN using a secure communication channel and the GWN provides a smart card (consisting of some secret parameter which is known only to the GWN) SC_i to U_i. The steps associated with the proposed user registration phase are described in following Steps R1, R2, R3 and summarized in Table 7 (using Steps 1–3).

Table 7. User registration phase of proposed protocol.

<table>
<thead>
<tr>
<th>Step 1: For User (U_i)</th>
<th>Step 2: For Gateway (GWN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U_i inputs ID_{U_i}, PW_{U_i} and B_i</td>
<td>GWN computes 1024 bit secret key x</td>
</tr>
<tr>
<td>Computes:</td>
<td>and Computes:</td>
</tr>
<tr>
<td>Gen(B_i) = (σ_{U_i}, τ_{U_i}), PB_{U_i} = h(PW_{U_i}</td>
<td></td>
</tr>
<tr>
<td>U_i transmits (ID_{U_i}, PB_i) to GWN</td>
<td>A_{U_i} = PB_i ⊕ h(ID_{U_i} ⊕ x),</td>
</tr>
<tr>
<td>ViaSecureChannel</td>
<td>B_{U_i} = h(ID_{U_i}</td>
</tr>
<tr>
<td></td>
<td>W_{U_i} = h(ID_{U_i}</td>
</tr>
<tr>
<td></td>
<td>GWN stores the value of P, A_{U_i}, B_{U_i},</td>
</tr>
<tr>
<td></td>
<td>W_{U_i} into SC_i.</td>
</tr>
<tr>
<td></td>
<td>GWN transmits (SC_i) to U_i</td>
</tr>
</tbody>
</table>

Step R1: A legitimate user U_i selects her identity ID_{U_i}, password PW_{U_i} and inputs his/her biometric information B_i into the generator function Gen() which generates a secret information σ_i and a public reproduction parameter τ_i. Then, U_i calculates PB_i = h(PW_{U_i}||σ_i) using secure hash function h() and sends ID_{U_i}, PB_i to the gateway node GWN.

Step R2: GWN generates a secret key x, selects a generator point P of G with order q and computes: K_{U_i} = h(ID_{U_i}||x) × P (where “×” is the scalar multiplication operator of elliptic curve),
A_{U_i} = PB_i ⊕ h(ID_{U_i} ⊕ x),
B_{U_i} = h(ID_{U_i}||PB_i)||h(ID_{U_i} ⊕ x),
W_{U_i} = h(ID_{U_i}||PB_i)||K_{U_i}.
After receiving the $SK_U$ the user $U_i$ computes $\sigma_i$ and sends $SC_i$ to the user $U_i$.

**Step R3:** After receiving the $SC_i$ from $GWN$, the user $U_i$ stores function $h()$, $Gen()$, $Rep()$ and the values of $T, \tau_i$ into $SC_i$.

### 5.3. User Authentication and Session Key Establishment Phase

In this module, we use the reproduction procedure $Rep(.)$ of fuzzy extractor for authentication the user $U_i$ with its noisy biometric credential $B'_i$ and we use Elliptic curve Diffie-Hellman procedure for sharing the common session key $SK$ between user $U_i$ and sensor node $SN_j$. The detail descriptions of this phase are illustrated in following Steps A1–A4 and summarized in Table 8 (using Steps 1–4).

**Table 8.** User authentication and session key establishment phase of the proposed protocol.

**Step 1: For User ($U_i$)**

- $U_i$ inputs $ID_{U_i}, PW_{U_i}$ and $B'_i$, Computes $\sigma_i = Rep(B'_i, \tau_i)$, $PB'_i = h(PW_{U_i}|\sigma_i)$, $h'_i(ID_{U_i} | x) = A_{U_i} \oplus PB'_i$, $B_{U_i} = h(ID_{U_i}|PB'_i|h'_i(ID_{U_i} | x))$. If $B'_{U_i} = B_{U_i}$ then
  
  - Evaluate $K_{U_i} = W_{U_i} \oplus h'(ID_{U_i}|PB'_i)$.
  - Generate $r_{U_i} \in Z^*_q$.
  - Find current time stamp $T_{U_i}$.
  - $X_{U_i} = r_{U_i} \times P, X'_{U_i} = r_{U_i} \times K_{U_i}$.
  - $\alpha = Enc_{X'_{U_i}[ID_{SN_j}|T_{U_i}]}$.
  - Construct a message $M_1 = \langle ID_{U_i}, X_{U_i}, \alpha \rangle$.
  
else

- $U_i$ is unauthenticated, abort this phase.

**Step 2: For Gateway ($GWN$)**

- $GWN$ computes $X'_{U_i} = h(ID_{U_i}|x) \times X_{U_i}$, $[ID_{SN_j}|T_{U_i}] = Dec_{X'_{U_i}[\alpha]}$.

if $T' - T_{U_i} \leq \Delta T$ then

- Generates $r_{SN_j} \in Z^*_q$.
- Calculate $Y_{SN_j} = r_{SN_j} \times P$.
- Session key $sk = r_{SN_j} \times X_{U_i}$.
- Find Current time-stamp $T_{GWN}$.
- $\beta = Enc_{X'_{U_i}[ID_{SN_j}|Y_{SN_j}|T_{GWN}]}$.
- $\gamma = Enc_{K_{GWN}}[ID_{U_i}|sk|\beta|T_{GWN}]$.
- Construct the message $M_2 = \langle \gamma \rangle$.

ViaPublicChannel

GWN transmits $M_2$ to $SN_j$.

else

- Replay and energy exhausting attack possible. Abort this phase.

**Step 3: For Sensor Node ($SN_j$)**

SN$_j$ computes $[ID_{U_i}|sk||\beta||T_{GWN}] = Dec_{K_{SN_j}}[\gamma]$.

if $T'' - T_{GWN} \leq \Delta T$ then

- Store the session key $sk$.

- Construct the message $M_3 = \langle \beta \rangle$.

ViaPublicChannel

SN$_j$ transmits $M_3$ to $U_i$.

else

- Replay and energy exhausting attack possible. Abort this phase.

**Step 4: For User ($U_i$)**

- $U_i$ computes $[ID_{SN_j}|Y_{SN_j}|T_{GWN}] = D_{X'_{U_i}}[\beta]$.

if $T''' - T_{GWN} \leq 2\Delta T$ then

- Establish the session key $sk = r_{U_i} \times Y_{SN_j}$ with $SN_j$. Where $r_{U_i} \times Y_{SN_j} = r_{SN_j} \times X_{U_i}$ based on ECDH.

else

- Replay and energy exhausting attack possible. Abort this phase.
Step A1: \( U_i \) inputs \( ID_{U_i}, PW_{U_i}, B_i' \) and computes \( \sigma_i' = \text{Rep}(B_i', \tau_i) \) using reproduction function of fuzzy extractor as described in Definition 4. Then, \( U_i \) calculates \( PB_i' = h(PW_{U_i} || \sigma_i'), h'(ID_{U_i} \oplus x) = A_{U_i} \oplus PB_i', B_i'_{U_i} = h(ID_{U_i} || PB_i' || h'(ID_{U_i} \oplus x)) \).

If the equivalent condition \( B_i'_{U_i} = B_i \) does not fulfill; abort the protocol. Otherwise, \( U_i \) evaluates \( K_{U_i} = W_{U_i} \oplus h'(ID_{U_i} || PB_i'), \) generates a random number \( r_{U_i} \in \mathbb{Z}_q \), and find out her current time stamp \( T_{U_i} \). Then, the user \( U_i \) calculates \( X_{U_i} = r_{U_i} \times P, X_{U_i}' = r_{U_i} \times K_{U_i} \) (where \( \times \) is the scalar multiplication operator of elliptic curve) and encrypts the message \( (ID_{SN_j} || T_{U_i}) \) considering \( X_{U_i}' \) as a symmetric key to find: \( \alpha = Enc_{X_{U_i}'}(ID_{SN_j} || T_{U_i}) \). Finally, \( U_i \) Construct a message \( M_1 = (ID_{U_i}, X_{U_i}, \alpha) \) and sends \( M_1 \) to the gateway node GWN.

Step A2: After receiving the message \( M_1 \), the gateway node GWN computes \( X_{U_i} = h(ID_{U_i} || x) \times X_{U_i} \) and decrypts the cipher text \( \alpha \) considering \( X_{U_i} \) as a symmetric key to find: \( [ID_{SN_j} || T_{U_i}] = \text{Dec}_{X_{U_i}'}(\alpha) \). And if the condition \( T' - T_{U_i} \leq \Delta T \) does not fulfill; the GWN aborts the protocol. Otherwise, the gateway node GWN generates a random number \( r_{SN_j} \in \mathbb{Z}_q \) and calculates \( Y_{SN_j} = r_{SN_j} \times P \), the session key \( sk = r_{SN_j} \times X_{U_i} \) (where \( \times \) is the scalar multiplication operator of elliptic curve). Then, the gateway node GWN finds its current time stamp \( T_{GWN} \) and calculates: \( \beta = Enc_{X_{U_i}}(ID_{SN_j} || Y_{SN_j} || T_{GWN}), \gamma = Enc_{K_{GWN}}([ID_{U_i} || sk] || \beta || T_{GWN}) \). Finally, GWN construct the message \( M_2 = (\gamma) \) and sends \( M_2 \) to the sensor node \( SN_j \).

Step A3: After receiving the message \( M_2 \), the sensor node \( SN_j \) decrypts the cipher text \( \beta \) using symmetric key \( K_{GWN} \) to find out: \( [ID_{U_i} || sk] || \beta || T_{GWN} = \text{Dec}_{K_{GWN}}(\gamma) \). If the condition \( T'' - T_{GWN} \leq 2\Delta T \) fulfills, \( SN_j \) stores the session key \( sk \) and finally transmits \( \beta \) to \( U_i \).

Step A4: After receiving the message \( M_3 \), the user \( U_i \) decrypts the message \( \beta \) considering \( X_{U_i} \) as a symmetric key and find out: \( [ID_{SN_j} || Y_{SN_j} || T_{GWN}] = \text{Dec}_{X_{U_i}'}(\beta) \). Once the condition \( T''' - T_{GWN} \leq 2\Delta T \) fulfills, the user \( U_i \) establishes the session key \( sk = r_{U_i} \times Y_{SN_j} \) with \( SN_j \).

Where \( r_{U_i} \times Y_{SN_j} = r_{SN_j} \times X_{U_i} \) based on ECDH problem.

5.4. User’s Credential Update Phase

If a legitimate user gets authenticated using her identity \( ID_{U_i} \), password \( PW_{U_i} \), biometric information \( B_i \) and the smart card \( SC_i \), she can update her password and biometric information using the mechanism described in Table 9.

Table 9. User’s credential update phase of proposed protocol.

<table>
<thead>
<tr>
<th>Step</th>
<th>User’s credential update phase of proposed protocol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>U_i inserts ( SC_i ) into the card reader and inputs ( ID_{U_i}, PW_{U_i}, B_i' ).</td>
<td></td>
</tr>
<tr>
<td>Then, ( U_i ) computes ( \sigma_i' = \text{Rep}(B_i', \tau_i) ) [Using fuzzy extractor].</td>
<td></td>
</tr>
<tr>
<td>( PB_i' = h(PW_{U_i}</td>
<td></td>
</tr>
<tr>
<td>( B_i' = h(ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td><strong>if</strong> ( B_i' = B_i ) <strong>then</strong></td>
<td></td>
</tr>
<tr>
<td>( U_i ) calculates ( h(ID_{U_i} \oplus x) = A_{U_i} \oplus PB_i' ), ( K_{U_i} = W_{U_i} \oplus h(ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td>( U_i ) inputs new ( PW_{new}, B_{new} ),</td>
<td></td>
</tr>
<tr>
<td>Then, ( U_i ) computes ( \text{Gen}(B_{new}) = (\sigma_{new}, \tau_{new}) ) [Using fuzzy extractor].</td>
<td></td>
</tr>
<tr>
<td>( PB_{new} = h(PW_{new}</td>
<td></td>
</tr>
<tr>
<td>( A_{new} = PB_{new} \oplus h(ID_{U_i} \oplus x), )</td>
<td></td>
</tr>
<tr>
<td>( B_{new} = h(ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td>( W_{new} = h(ID_{U_i}</td>
<td></td>
</tr>
<tr>
<td>Finally, replaces the value of ( A_{U_i}, B_{U_i}, W_{U_i} ) with ( A_{new}, B_{new}, W_{new} ) into ( SC_i )</td>
<td></td>
</tr>
<tr>
<td><strong>else</strong></td>
<td></td>
</tr>
<tr>
<td>User ( U_i ) is unauthenticated. Abort protocol to avoid stolen smart card attack.</td>
<td></td>
</tr>
</tbody>
</table>
6. Security Analysis:

To estimate the security strength of our proposed protocol, we perform the informal and formal analysis of security features.

6.1. Informal Analysis

Our proposed protocol can withstand various known security attacks as illustrated in the following propositions.

**Proposition 1.** The proposed protocol is secure against Stolen Smart Card Attack.

**Proof.** An adversary $A$ who have stolen the smart card $SC_i$ can extract the intimate data such as $A_Ui, B_Ui, W_Ui, h(.), Rep(.), Gen(.)$. However, in our protocol the most important private information such as $A_Ui$ is stored in well-protected form. If $A$ succeed to find out $A_Ui$, it can not find out $PB_i$ or $h(ID_Ui \oplus x)$ using frequency analysis attack. The private information $A_Ui$ also can not be extracted by $A$ because it is hashed after concatenated with $PW_Ui$.

**Proposition 2.** The proposed protocol is secure against node compromise attack.

**Proof.** According to our presumption, the sensor node $SN_j$ is not fixed with tamper resistant hardware, therefore an adversary $A$ can capture the sensor node $SN_j$ and find out the value of the key $K_{GSN_j}$ and session key $sk$. However, $A$ can not use the same session key at next session because we made the session key unique using the random number $r_{Ul}$ and $r_{SN_j}$. If $A$ captures the key $K_{GSN_j}$ from $SN_j$, it can establish a session key with any user who wants to access data from $SN_j$ but it can not establish a session key with any other user associated with non-compromised sensor node because the key $K_{GSN_j}$ is uniquely given to $SN_j$.

**Proposition 3.** The proposed protocol is secure against Man-in-the-middle attack.

**Proof.** Suppose an adversary $A$ eavesdrops the message $M_1$ during user authentication and session key establishment phase, generates a random number $r_A$ and the current time-stamp $T_A$. However, $A$ can not evaluate the value of $X'_{Ul}$ without knowing the bio-metric information and smart card credentials of $U_i$ in order to decrypt and modify the value of $a$. Likewise, it is computationally infeasible for an adversary $A$ to modify the value of $\gamma$ and $\beta$ without knowing the key $K_{GSN_j}$ and $X_{Ul}$ respectively. Therefore, our scheme is secure against the Man-in-the-middle attack.

**Proposition 4.** The proposed protocol is secure against replay attack.

**Proof.** Suppose an adversary $A$ intercepts the message $M_1 = \left< ID_{Ul}, X_{Ul}, a = Enc_{X'_{Ul}} [ID_{SN_j} || T_{Ul}] \right>$ from the public communication channel established between Step 1 and Step 2 of user authentication and session key establishment phase of our proposed protocol. Sometime later, $A$ resends $M_1$ to the gateway node $GWN$. At the gateway node $GWN$, the message $M_1$ will be declared as replayed because the time-stamp $T_{Ul}$ will not be fresh and the condition $T' - T_{Ul} \leq \Delta T$ will not be satisfied. Similarly, if the adversary $A$ intercepts and replays the messages $M_2$ and $M_3$ from the public communication channels of user authentication and session key establishment phase, they will be declared (after time-stamp verification) as replayed messages by the sensor node $SN_j$ and the user $U_i$ respectively. Therefore, our scheme is secure against the replay attack.

**Proposition 5.** The proposed protocol is resilience against gateway node capture attack.
Theorem 1. If the hash function $h$ can be represented as:

$$h = h(PW_{U_i}||\sigma_{U_i}, X_{U_i})$$

then it is not possible to find out the value of user’s password $PW_{U_i}$ and biometric information $B_i$ from the captured Gateway node $GWN$. Then, $A$ can not impersonate the user $U_i$ based on the authentication phase of our proposed protocol. Hence, our proposed protocol is resilience against gateway node capture attack.

Proof. In the registration phase of our proposed protocol, the user $U_i$ transmits only the value of $PB_{U_i} = h(PW_{U_i}||\sigma_{U_i})$, instead of sending the original biometric information $B_i$, to the gateway node $GWN$. Where, $\sigma_{U_i}$ is generated using Fuzzy extractor and the function $h(.)$ is a secure one-way hash function. Therefore, for an adversary $A$, it is not possible to find out the value of legitimate user’s identity $ID$, password $PW_{U_i}$, and biometric information $B_i$ from the captured Gateway node $GWN$. Then, $A$ can not impersonate the user $U_i$ based on the authentication phase of our proposed protocol. Hence, our proposed protocol is resilience against gateway node capture attack.

6.2. Formal Security Analysis

In this section, we first use random oracle model to perform the formal security analysis of our proposed protocol. Then, we use Sczyter tool [30] to verify all the security claims specified in different roles. Afterwards, we automatically validate the safety of our protocol using AVISPA [31] (version v1.1) tool based on Dolev-Yao intruder model with OFMC and CL-AtSe back-ends. We do logical verification using BAN logic to ensure that our protocol works correctly and achieve the specified security feature.

6.2.1. Formal Security Verification Using Random Oracle Model

The random oracle model (ROM) is a robust tool proposed by Bellare and Rogaway in [32] to make it possible to execute meticulous “proofs of security” for particular fundamental cryptographic protocols.

A random oracle is a theoretical black box that responds to every individual query with an accurate random response chosen uniformly from its output domain. If a query is occurring several times, it responds the same way every time that query is performed.

Based on random oracle model, the following Theorem 1 shows that our protocol can resist various security attacks.

With the help of random oracle model we prove that for an adversary $A$ it is not possible to obtain the value of legitimate user’s identity $ID_{U_i}$, password $PW_{U_i}$, biometric information $B_i$, and the session key $sk$. Considering the method of contraction, we assume that there exist some random oracles as illustrated in following Definitions 5–7.

Definition 5. Reveal1: Given a hash value $y = h(s)$, this oracle unconditionally outputs the string $s$.

Definition 6. Reveal2: Given an encrypted value $Enc_{K}[s]$, this oracle unconditionally outputs the string $s$ without knowing the key $k$.

Definition 7. Reveal3: Given $P \in E_p(a, b)$ and the public parameter $X = r \times P \in E_p(a, b)$, this oracle outputs the private key $r$.

Theorem 1. If the hash function $h()$, encryption mechanism $Enc$, and elliptic curve Diffie-Hellman problem $ECDH$ follows the random oracle Reveal1, Reveal2 and Reveal3 respectively; our scheme resist the adversary $A$ for deriving the values of user $U_i$’s secret parameters $PW_{U_i}, \sigma_{U_i}, K_{U_i}$ and $X_{U_i}$.

Proof of Theorem 1. If we assume that, there exist the oracle Reveal1, Reveal2, Reveal3 which can derive string $s$ from the hash digest $d = h(s)$, string $s$ from the cipher-text $Enc_{K}[s]$ and private key $r$ from the public parameter $X = r \times P$ respectively. Then, the adversary $A$ can design an procedure $Exp_{ECDH}^{i} (A)$ as shown in Algorithm 1 such that probability of success of $Exp_{ECDH}^{i} (A)$ is $Success_{ECDH}^{A} = |P[Exp_{ECDH}^{i} (A) = 1] - 1|$. The advantage function for $Exp_{ECDH}^{i} (A)$ can be represented as:

$$Adv_{ECDH}^{A}((t_1 + t_2 + t_3), (q_{R_1} + q_{R_2} + q_{R_3})) = [Adv_{ECDH}^{A}(t_1) \cdot Adv_{ECDH}^{A}(t_2) \cdot Adv_{ECDH}^{A}(t_3)].$$
According to Algorithm 1, there exist oracle Reveal1, Reveal2, Reveal3 capable of finding the preimage of \( h() \), the plain-text \( s \) from the cipher-text \( Enc_k[s] \) and private key \( r \) from the public parameter \( X = r \times P \).

**Algorithm 1: EXP^Adv\_{Exp-EC\text{-}ECDH}**

1. Extract \( \{P, A_{U_i}, B_{U_i}, W_{U_i}, \tau_i, T, h(), Gen(), Rep()\} \) from \( SC_i \) using simple and differential power analysis attacks. Where
2. \( A_{U_i} = PB_i \oplus h(ID_{U_i} \oplus x) \)
3. \( B_{U_i} = h(ID_{U_i}||PB_i||h(ID_{U_i} \oplus x)) \)
4. \( W_{U_i} = h(ID_{U_i}||PB_i) \oplus K_{U_i} \)
5. Call Reveal1 oracle on input \( B_{U_i} \) to retrieve the information of 
   \( ID_{U_i}, PB_i, h(ID_{U_i}||x) \) as \( ID'_{U_i}||PB'_i||h(ID_{U_i}||x)' \) \( \leftarrow \) Reveal1(\( B_{U_i} \))
6. Call Reveal1 oracle on input \( PB'_i \) to retrieve the information of 
   \( PW_{U_i}, \sigma_i' = (PW_{U_i}', \sigma_i') \) \( \leftarrow \) Reveal1(\( PB'_i \))
7. Compute \( h(ID'_{U_i}||PB'_i) \)
8. Compute \( K'_U_i = W_{U_i} \oplus h(ID'_{U_i}||PB'_i) \)
9. Intercept the message \( M_1 = \langle ID_{U_i}, X_{U_i}, \alpha \rangle \)
10. **if** \( (ID'_{U_i} = ID_{U_i}) \) **then**
    
    Call Reveal3 oracle on input \( X_{U_i} \) to retrieve the private information \( r_{U_i} \) as \( r_{U_i} \leftarrow \) Reveal3(\( X_{U_i} \)),
    
    Compute the established secret \( X'_{U_i} = r'_{U_i} \times K'_{U_i} \),
    
    Call Reveal2 oracle on input \( \alpha \) to retrieve the information
    
    \( IDS_{SN}, T_{U_i} \) as \( ID'_{SN}||T'_{U_i} \leftarrow \) Reveal2
    
    **if** \( (E_{X'_{U_i}}[ID'_{SN}||T'_{U_i}] = \alpha) \) **then**
    
    Accept the derived \( ID'_{U_i}, PW_{U_i}', \sigma'_{i} and X'_{U_i} \) as the correct identity, password, secret biometric data and the established secret information of the user \( U_i \)
    
    Return 1 (Success)
    
    **else**
    
    Return 0 (Failure)
    
    **else**
    
    Return 0 (Failure)

Therefore, the adversary \( A \) can get the values of \( PW_{U_i}, \sigma_i, K_{U_i}, X'_{U_i} \). However, according to Definitions 1–3 (defined in Section 3) we have

\[
Adv^\text{Adv}_{\text{Exp}}(t_1) = Pr[(s, s') \leftarrow_R A : s \neq s', h(s) = h(s')],
\]

\[
Adv^\text{IND-CPA}_{Enc,A}(t_2) = 2Pr[A \leftarrow O_k; (b_0, b_1 \leftarrow A); \gamma \leftarrow_R 0, 1; \gamma \leftarrow_R O_k(b_\gamma) : A(\gamma) = \tau - 1,
\]

\[
Adv^\text{ECDH}_{A}(t_3) = Pr[(r_{U_i}, P) \leftarrow_R A : X_{U_i} = r_{U_i} \times P]
\]

Where \( Adv^\text{ECDH}_{A}(t_1) \leq \tau, Adv^\text{IND-CPA}_{Enc,A}(t_2) \leq \tau, Adv^\text{ECDH}_{A}(t_3) \leq \tau. \)
Therefore, $\text{Adv}^{h_{\text{Enc}}-\text{ECDH}}_A((t_1 + t_2 + t_3), (qR_1 + qR_2 + qR_3)) \leq \tau$.

which indicates that $\text{Adv}^{h_{\text{Enc}}-\text{ECDH}}_A((t_1 + t_2 + t_3), (qR_1 + qR_2 + qR_3))$ is negligible for any probabilistic polynomial time adversary $A$. Now, we find that the secure hash function $h()$, encryption mechanism $\text{Enc}_k[s]$ and elliptic curve Diffie-Hellman problem $\text{ECDH}$ defined in Section 3 contradicts the oracle Reveal1, Reveal2 and Reveal3 respectively considered in Algorithm 1. This indicates that our scheme resist the adversary $A$ for deriving the values of the secret parameters $PW_{U_i}, c_i, K_{U_i}$, and $X'_{U_i}$. Hence, the theorem is proved. $\square$

Where $q_{R_1}, q_{R_2}, q_{R_3}$ represents the total number of queries made to the Reveal1, Reveal2, Reveal3 oracle respectively.

6.2.2. Verification Using Scyther tool

The Scyther tool algorithm provides some novel features, including:

- Guaranteed termination, after which the result is either unbounded correctness, falsification, or bounded correctness.
- Efficient generation of a finite representation of an infinite set of traces concerning patterns, also known as a complete characterization.
- State-of-the-art performance, which has made new types of protocol analysis feasible, such as multi-protocol analysis.

The proposed protocol is specified in Security Protocol Description Language (SPDL). The protocol specification defines sequence of roles of $U_i$, $\text{GWN}$ and $\text{SN}_j$. Every role encompasses sequences of events (i.e., send, receive, declarations and claim events). The protocol specification and the roles of $U_i$, $\text{GWN}$ and $\text{SN}_j$ are represented in Tables 10–13 respectively. The verification result obtained using Scyther tool is shown in Figure 2. The result indicates that no attacks found on each of the claims specified in our protocol.

6.3. Verification Using AVISPA Tool

In this section, we first explain the setup procedure and some basic features of AVISPA tool which we use for the formal security analysis of our protocol. Afterwards, we describe the implementation of our protocol using High-Level Protocol Specification Language (HLPSL). Finally, we discuss about the results obtained.

6.3.1. Experimental Setup and the Size of the Entities Involved in WSNs/IoT for the Simulation of Proposed Protocol Using AVISPA Tool

In order to simulate the proposed protocol on AVISPA v1.1, we use a Security Protocol ANimator (SPAN) Version 1.6 on a computer system having ubuntu 16.04 LTS operating system (64 bit), Intel (R) core (TM) i7-6500U CPU @ 2.50 GHz x4 processor, and 8 GB RAM. We extract the archive avispa-package-1.1_Linux-i686.tgz, set up the environment variable AVISPA_PACKAGE and keep the script of the avispa protocol in the execution path. We implement our protocol considering minimal number of entities involved in WSNs/IoT (i.e, one user $U_i$, one sensor node $SN_j$ and one gateway node $\text{GWN}$) using Dolev-Yao model [33] with a bounded number of sessions, specified goal, On-the-Fly Model-Checker (OFMC) and Constraint-Logic based Attack Searcher (CL-AtSe) backend.

6.3.2. Basic Features of AVISPA Tool

AVISPA is a broadly accepted and robust software tool for automatically validating (using push-button mechanism) the security features of the protocols used in Internet of Things. The architecture of AVISPA tool is shown in following Figure 3.
Table 10. Specification of the proposed protocol in SPDL.

| hashfunction h; /*Secure hash function */ |
| const XOR: Function; /*XOR operation */ |
| const Concat: Function; /*Concatenation Function */ |
| const EccMul: Function; /*Scalar Point Multiplication Operation of ECC */ |
| const Gen: Function; /*Generator function of Fuzzy Extractor*/ |
| const Rep: Function; /*Reproduction function of Fuzzy Extractor*/ |
| const Enc: Function; /*Encryption Function*/ |
| const Dec: Function; /*Decryption Function*/ |

/*IDui, PWui, Bi represents the identity, password and bio-metric information of the user Ui respectively. Kgsnj denotes the secret key shared between sensor and gateway node. Tui, Tgwn denotes the current time-stamp of user, gateway respectively. Rui and Rsnj represents the random number generated at user Ui and sensor node SNj respectively. */

protocol Protocol(Ui, GWN, SNj)
{ macro SIGi = Gen(Bi); /*macro defines abbreviations for particular term */
  macro PBi = h(Concat (PWui, SIGi));

  macro SIGi' = Rep(Bi', TAUi);
  macro PBi' = h(Concat(PWui, SIGi'));

  macro Kui = EccMul(h(Concat(IDui,x)),P);
  macro Aui = XOR(PBi, h(XOR(IDui,x)));
  macro Bui = h(Concat(IDui,PBi,h(XOR(IDui,x))));
  macro Wui = XOR(h(Concat(IDui,PBi)), Kui);

  macro Xui = EccMul(Rui, P);
  macro Xui'= EccMul(Rui, Kui);

  macro Ysnj = EccMul(Rsnj,P);
  macro sk = EccMul(Rui, Ysnj);

  macro Alpha = Enc(Xui', Concat(IDsnj,Tui));
  macro Alpha' = Dec(Xui', Enc(Xui', Concat(IDsnj,Tui)));

  macro Beta = Enc(Xui', Concat(IDsnj,Ysnj,Tgwn));
  macro Beta' = Dec(Enc(Xui', Concat(IDsnj,Ysnj,Tgwn)));

  macro Gamma = Enc(Kgsnj, Concat(IDui,sk,Beta,Tgwn));
  macro Gamma' = Dec(Enc(Kgsnj, Concat(IDui,sk,Beta,Tgwn)));

AVISPA involves HLPSL to specify the protocol in a file with.hlpsl extension. It performs a static analysis to verify the executability of the protocol. A HLPSL2IF translator is used to translate the HLPSL specification into an Intermediate Formate (IF) specification, which is tool-independent language and compatible for automated deduction. The IF specifications are provided as an input to one of the four back-ends. The back-ends are as follows:

1. On-the-fly model-checker (OFMC)
2. Constraint-logic based attack searcher (CL-AtSe)
3. SAT-based model-checker (SATMC)
4. Tree automata based on automatic approximation for the analysis of security protocols (TA4SP).
Table 11. Specification of the user’s role in SPDL.

```plaintext
role Ui
{
    var Tsnj, Tgwn: Nonce;
    fresh Tui: Nonce; /* Time-stamp Tui is freshly generated */
    const IDui, PWui, Bi, Bi', PBi, IDsnj, Rui, Rsnj, Kgsnj, Xui, Xui', x, Tui, Tgwn, P, TAUi: Ticket;

    send_1(Ui, GWN, IDui, PBi); /* Ui sends IDui, PBi to GWN */
    recv_2(GWN, Ui, P, Aui, Bui, Wui); /* Ui received P, Aui, Bui, Wui from GWN */

    send_3(Ui, GWN, Xui, Alpha);
    recv_5(SNj, Ui, Beta);
    match(Beta', Beta); /* Test the equality of Beta’ and Beta */

    claim_Ui1(Ui, Secret, Bi); /* Bi should be secret for Ui */
    claim_Ui2(Ui, Secret, PWui);

    claim_Ui3(Ui, Secret, x);
    claim_Ui4(Ui, Secret, Xui');

    claim_Ui5(Ui, Secret, Tui);
    claim_Ui6(Ui, SKR, sk); /* Session key sk should be secret */

    claim_Ui7(Ui, Niagree); /* Non-injective agreement */
    claim_Ui78(Ui, Nisynch); /* Non-injective synchronization */
}
```

Table 12. Specification of the gateway node’s role in SPDL.

```plaintext
role GWN
{
    fresh Tgwn: Nonce;
    var Tui: Nonce;
    const IDui, PWui, IDsnj, Bi, P, x, Rui, Tui, Bi, PWui: Ticket;

    recv_1(Ui, GWN, IDui, IPBi);
    send_2(GWN, Ui, P, Aui, Bui, Wui);

    recv_3(Ui, GWN, IDui, IDsnj, Xui, TSui, Alpha);
    match (Alpha, Alpha');

    send_4(GWN, SNj, Beta, Gamma, Xui, TGgwn, TUgwn);
    claim_GWN1(GWN, Secret, Tgwn);

    claim_GWN2(GWN, Secret, x);
    claim_GWN3(GWN, Secret, k(GWN, SNj));

    claim_GWN4(GWN, Secret, Kui);
    claim_GWN5(GWN, Secret, Xui');
}
```
Table 13. Specification of the sensor’s role in SPDL.

```plaintext
role SNj
{
  var Tgwn: Nonce;
  fresh Tsnj: Nonce;

  const IDui, IDsnj, x, Rui, Tui, P, Bi, PWui, Rsnj: Ticket;
  recv_4(GWN, SNj, Beta, Gamma, Xui, TGgwn, T Ugwn);
  match(Beta, Beta');
  send_5(SNj, Ui, Delta, Gamma, Ysnj, Tsnj, T Ugwn);
  claim_SNj1(SNj, Secret, Tgwn);
  claim_SNj2(SNj, Secret, Rsnj);

  claim_SNj3(SNj, Secret, Tsnj);
  claim_SNj4(SNj, Secret, k(GWN, SNj));

  claim_SNj5(SNj, SKR, h(EccMul(Rsnj, Xui)));
}
```

Figure 2. Security verification result obtained using Scyther tool.
6.4. Implementation of the Proposed Protocol Using HLPSL

The HLPSL specification of the protocol consist of some important section as follow:

1. Basic Role: Basic role explains the activity of the entities (e.g., User $U_i$, Gateway $GWN$ and Sensor node $SN_j$) involve in the protocol.
   - Each role may have some parameter like $U_i$, $GWN$, $SN_j$ of type agent and $K_{ui1}$, $K_{gsnj}$ of type symmetric_key.
   - The parameter RCV and SND denotes the agent’s communication channels for receiving and sending the information.
   - The parameter (dy) represents the Dolev-Yao intruder model for the channel.
   - The function H, Gen, Rep, EccMul, Enc, Dec and XOR corresponding to the hash function, fuzzy extractor’s generator, fuzzy extractor’s reproduction, elliptic curve scalar multiplication, encryption, decryption and logical XOR operations respectively.
   - The term hash_func represents all the functions which are not easily invertible because the random non-invertible arithmetic operators are not supportable in HLPSL.
   - The term “played_by $U_i$” denotes that the role User is played by $U_i$.

The HLPSL specification of roles of $U_i$, $GWN$ and $SN_j$ are shown in Tables 14–16 respectively.

2. Transitions: The transitions are declared in steps. It consist of trigger which fires when an event occurs. For any States in a transition if a message received on channel RCV, then transition fires and allocates a new value to the State.

3. Composed Roles: It makes one or more basic roles to execute together and represent the sessions involve in the protocol. The operator $\land$ represents the parallel execution of the roles.

The HLPSL specification of proposed protocol’s session is shown in Table 17.

4. Environment: It consist of global constant and session composition, where the adversary may execute some role as a authorized user.

The HLPSL specification of proposed protocol’s environment is shown in Table 18.

5. Security Goal: This module specifies the security Goal of the protocol. Some important predicates used in this module are as follows:

![Figure 3. AVISPA Architecture.](image-url)
• secret([PW_i,Bi,SIGi'], sub1, Ui): It indicates that the information [PW_i,Bi,SIGi'] is secretly shared to Ui and it can be recognized with a constant identity sub1 in goal section.

• witness(Ui, GWN, gateway_user_gu, Tui, Alpha'): It represents the weak authenticity of Ui by GWN and Ui is the witness for the data [Tui', Alpha']. The identity of this goal is represented as gateway_user_gu in goal section.

• request(Ui,SNj, user_sensor_us, Skey'): It represents the strong authenticity of Ui by SNj on Skey with an identity user_sensor_us.

• Symbols: Concatenation (.) is used for message composition (e.g., SND (IDi.PBi')) and Commas (,) is used in case of multiple arguments of events or functions (e.g., secret([PW_i,Bi,SIGi', sub1, Ui]).

Table 14. Specification of Ui's role in HLPSL.

| role user(Ui, GWN, SNj: agent, Xui1, Kgsnj: symmetric_key, H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func, SND, RCV: channel(dy)) played_by Ui def= local State: nat, IDui, IDsnj, PWui, Bi, Bi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui, Bui, Wui, Alpha, Beta, Gamma, Ysnj, Ysnj1, Tui, Tgwn, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8, gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id init State: = 0 transition 0. State = 0 ∧ RCV (start) = ⊥ State': = 2 ∧ SIGi': = Gen(Bi) ∧ PBi': = H(PWui.SIGi') ∧ secret([PWui,Bi,SIGi', sub1, Ui]) ∧ SND (IDui.PBi') 2. State = 2 ∧ RCV (P.Aui'.Bui'.Wui') = ⊥ State': = 5 ∧ Rui': = new() ∧ Tui': = new() ∧ secret(Rui', sub2, Ui) ∧ SIGi1': = Rep(Bi1.TAUi) ∧ PBi1': = H(PWui.SIGi1') ∧ Kui1': = XOR(Wui, H(IDui.PBi1')) ∧ Xui': = EccMul(Rui'.P) ∧ Xui1': = EccMul(Rui'.Kui1') ∧ secret(Xui1', sub3, Ui, GWN) ∧ Alpha': = Enc(IDsnj,Tui) ∧ SND(IDui.Xui'.Alpha') ∧ witness(Ui, GWN, gateway_user_gu, Tui, Alpha') 6. State = 5 ∧ RCV(Beta1') = ⊥ State': = 6 ∧ Ysnj1': = Dec(Beta1') ∧ Skey': = EccMul(Rui'.Ysnj1') ∧ request(Ui,SNj, user_sensor_us, Skey') end role

The HLPSL specification of proposed protocol’s goal is shown in Table 19.
Table 15. Specification of GWN’s role in HLPSL.

```plaintext
role gateway(Ui, GWN, SNj: agent,
Xui1, Kgsnj: symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by GWN def=
local
State: nat,
IDui, IDsnj, PWui, Bi, Bi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui, Bui, Wui, Alpha, Beta,
Gamma, Ysnj, Tui, Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 1
transition
1. State = 1 ∧ RCV (IDui.PBi') = ⊥
State': = 3 ∧ X': = new()
   ∧ Kui': = EccMul(H(IDui.X').P)
   ∧ Aui': = XOR(PBi'.H(XOR(IDui.X')))
   ∧ Bui': = H(IDui.PBi'.XOR(IDui.X'))
   ∧ secret(X',sub4, GWN)
   ∧ Wui': = XOR(H(IDui.PBi).Kui')
   ∧ secret(Kui', sub5, GWN,Ui)
   ∧ SND(P,Aui'.Bui'.Wui')
3. State = 3 ∧ RCV(IDui.Xui'.Alpha') = ⊥
State': = 4 ∧ Tgwn'. =new()
   ∧ request(GWN, Ui, gateway_user_gu, Alpha')
   ∧ IDsnj': = Dec(Alpha')
   ∧ Rsnj': = new()
   ∧ Ysnj': = EccMul(Rsnj'.P)
   ∧ Beta': = Enc(IDsnj'.Ysnj'.Tgwn)
   ∧ secret(Kgsnj, sub6, GWN,SNj)
   ∧ Gamma': = Enc(IDui.Skey'.Beta'.Tgwn')
   ∧ SND(Gamma')
   ∧ witness(GWN, Ui, gateway_user_gu, Tgwn')
end role
```

6.5. Description of the Output Format Generated by AVISPA Tool

The output generated by AVISPA tool describes the final result obtained under various conditions after the security analysis of the protocol. The output produced by the AVISPA tool consist of following sections and subsections:

- **Summary:** This section specifies the security reliability of the protocol regarding safe, unsafe or inconclusive.
- **Details:** In this portion, the output specifies the environment and the context under which the protocol is claimed to be safe, unsafe or inconclusive.
- **Protocol:** It indicates the name of the protocol given as an input for security verification.
- **Goal:** This section represents the specified security goal of the protocol.
- **Backend:** This section represents one of the four back-ends used for the analysis of the protocol.

The verification result of AVISPA [31] tool is shown in Table 20 which represents that the proposed protocol is safe from various attacks (like man-in-the-middle attack, replay attack etc.) using Dolev-Yao model [33] with bounded number of sessions, specified goal, On-the-Fly Model-Checker(OFMC) and Constraint-Logic based Attack Searcher (CL-AtSe) backend.
Table 16. Specification of SNj’s role in HLPSL.

role sensor(Ui, GWN, SNj: agent,  
Xui1, Kgsnj: symmetric_key,  
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,  
SND, RCV: channel(dy))  
played_by SNj def=  
local  
State: nat,  
IDui, IDsnj, PWui, Bi, Bi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui, Bui, Wui, Alpha, Beta,  
Gamma, Ysnj, Tui, Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text  
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,  
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id  
init  
State: = 4  
transition  
4. State = 4 ∧ RCV (Gamma') = ⊥  
State': = 5 ∧ Skey1' = Dec(Gamma'.Kgsnj)   
∧ secret(Skey1', sub7, SNj)   
∧ Beta1' = Dec(Gamma')   
∧ secret(Skey1', sub8, SNj)   
∧ SND(Beta1')  
end role

Table 17. Specification of proposed protocol’s session in HLPSL.

role session(Ui, GWN, SNj: agent,  
Xui1, Kgsnj: symmetric_key,  
H, Gen, Rep, EccMul, Enc, Dec, XOR: hash_func)  
def=  
local GWNUi, RUi, GWNSNj, RSNj, GWNGWN, RGWN: channel(dy)  
composition  
user(Ui, GWN, SNj, Xui1, Kgsnj, H, Gen, Rep, EccMul, Enc, Dec, XOR, GWNUi, RUi)  
∧ sensor(Ui, GWN, SNj, Xui1, Kgsnj, H, Gen, Rep, EccMul, Enc, Dec, XOR, GWNSNj,  
RSNj)  
∧ gateway(Ui, GWN, SNj, Xui1, Kgsnj, H, Gen, Rep, EccMul, Enc, Dec, XOR,  
GWNGWN, RGWN)  
end role

Table 18. Specification of proposed protocol’s environment in HLPSL.

role environment()  
def=  
const ui, gwn, snj: agent,  
xui1, kgsnj, kig: symmetric_key,  
h, gen, rep, eccMul, enc, dec, XOR: hash_func,  
sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,  
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id  
intruder_knowledge = ui,gwn,snj,kig  
composition  
session(ui, snj, gwn, xui1, kig, h, gen, rep, eccMul, enc, dec, XOR)  
∧ session(ui, snj, gwn, kgsnj, kig, h, gen, rep, eccMul, enc, dec, XOR)  
∧ session(ui, snj, gwn, kig, kgsnj, h, gen, rep, eccMul, enc, dec, XOR)  
end role
Table 19. Specification of proposed protocol’s goal in HLPSL.

```
goal
secrecy_of sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8
authentication_on gateway_sensor_gs, gateway_user_gu, user_sensor_us
end goal
environment()
```

Table 20. Security verification result obtained using AVISPA tool.

<table>
<thead>
<tr>
<th>Using OFMC BACKEND</th>
<th>Using CL-AtSe BACKEND</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SUMMARY</strong></td>
<td><strong>SUMMARY</strong></td>
</tr>
<tr>
<td>SAFE</td>
<td>SAFE</td>
</tr>
<tr>
<td><strong>DETAILS</strong></td>
<td><strong>DETAILS</strong></td>
</tr>
<tr>
<td>BOUNDED_NUMBER_OF_SESSIONS</td>
<td>BOUNDED_NUMBER_OF_SESSIONS</td>
</tr>
<tr>
<td><strong>PROTOCOL</strong></td>
<td><strong>PROTOCOL</strong></td>
</tr>
<tr>
<td>/home/cmb-lab-22/Desktop/Proto.if</td>
<td>/home/cmb-lab-22/Desktop/Proto.if</td>
</tr>
<tr>
<td><strong>GOAL</strong></td>
<td><strong>GOAL</strong></td>
</tr>
<tr>
<td>as_specified</td>
<td>as_specified</td>
</tr>
<tr>
<td><strong>BACKEND</strong></td>
<td><strong>BACKEND</strong></td>
</tr>
<tr>
<td>OFMC</td>
<td>CL-AtSe</td>
</tr>
<tr>
<td><strong>STATISTICS</strong></td>
<td><strong>STATISTICS</strong></td>
</tr>
<tr>
<td>Time: 984 ms</td>
<td>Analysed: 1956 states</td>
</tr>
<tr>
<td>parseTime: 0 ms</td>
<td>Reachable: 1956 states</td>
</tr>
<tr>
<td>visitedNodes: 456 nodes</td>
<td>Translation: 0.06 s</td>
</tr>
<tr>
<td>depth: 9 piles</td>
<td>Computation: 0.01 s</td>
</tr>
</tbody>
</table>

6.5.1. Logical Verification Using BAN Logic

In this subsection, we use BAN logic [34] to verify the freshness of time-stamp to avoid replay attack and we validate the message origin to achieve authenticity. The notation we use for logical verification is shown in Table 21.
Table 21. Notations used in verification using BAN logic.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{BAN}, Q_{BAN}$</td>
<td>Principals like $U_i$, GWN, and $SN_j$</td>
</tr>
<tr>
<td>$S$</td>
<td>Statements like $T_{U_i}, T_{GWN}, \alpha, \beta$ etc.</td>
</tr>
<tr>
<td>$K$</td>
<td>Secret key or secret informations like $K_{GSN_i}, X_{U_i}$ etc.</td>
</tr>
<tr>
<td>$P_{BAN} \equiv S$</td>
<td>$P_{BAN}$ believes $S$, or $P_{BAN}$ believes $S$ is true.</td>
</tr>
<tr>
<td>$P_{BAN} \leq S$</td>
<td>$P_{BAN}$ has received a information containing $S$ and it can read or repeat $S$</td>
</tr>
<tr>
<td>$P_{BAN} \sim S$</td>
<td>$P_{BAN}$ once said $S$. $P_{BAN}$ sent a data containing $S$ and it could be a fresh or old data.</td>
</tr>
<tr>
<td>$P_{BAN} \Rightarrow S$</td>
<td>$P_{BAN}$ has jurisdiction over $S$. That is $P_{BAN}$’s beliefs about $S$ should be trusted</td>
</tr>
<tr>
<td>$#(S)$</td>
<td>The information $S$ is fresh and it has not been sent before.</td>
</tr>
<tr>
<td>$P_{BAN} \equiv S_{Q_{BAN}}$</td>
<td>$S$ is a secret data and it is only known to $P_{BAN}$ or $Q_{BAN}$ and perhaps to the trusted principals</td>
</tr>
<tr>
<td>$&lt; S &gt;_{SI}$</td>
<td>$S_1$ is a secret and its presence gives the identity of whoever generates $&lt; S &gt;_{S_1}$</td>
</tr>
</tbody>
</table>

**Rule 1** Message meaning rule: \[ \frac{P_{BAN} \equiv P_{BAN} \equiv S_{Q_{BAN}} \equiv \{S\} \equiv S_{Q_{BAN}} \equiv S}{P_{BAN} \equiv Q_{BAN} \equiv P_{BAN} \equiv \{S\} \equiv S_{Q_{BAN}} \equiv S} \]. That is, if $P_{BAN}$ believes that she shared the key $K$ with $Q_{BAN}$, and $P_{BAN}$ sees the message $\{S\}$ encrypted with key $K$, $P_{BAN}$ believes that $Q_{BAN}$ once said $S$.

**Rule 2** Nonce verification rule: \[ \frac{P_{BAN} \equiv S_{Q_{BAN}} \equiv P_{BAN} \equiv \{S\} \equiv S_{Q_{BAN}} \equiv S}{P_{BAN} \equiv Q_{BAN} \equiv P_{BAN} \equiv \{S\} \equiv S_{Q_{BAN}} \equiv S} \]. That is, if $P_{BAN}$ believes $S$ is fresh and $Q_{BAN}$ once said $S$, $P_{BAN}$ believes $Q_{BAN}$ believes $S$.

**Rule 3** Jurisdiction rule: \[ \frac{P_{BAN} \equiv S_{Q_{BAN}} \equiv S_{Q_{BAN}} \equiv P_{BAN} \equiv \{S\} \equiv S_{Q_{BAN}} \equiv S}{P_{BAN} \equiv Q_{BAN} \equiv P_{BAN} \equiv \{S\} \equiv S_{Q_{BAN}} \equiv S} \]. That is, if $P_{BAN}$ believes that $Q_{BAN}$ had jurisdiction right to $S$ and believes $Q_{BAN}$ believes $S$, $P_{BAN}$ believes $S$.

In order to achieve the better security features, the proposed protocol should achieve the security Goals as defined in Table 22.

**Table 22.** Goals: The goals made to analyze the proposed scheme.

- **Goal 1** $U_i | \equiv K_{U_i}$
- **Goal 5** $U_i | \equiv GWN | \equiv T_{GWN}$
- **Goal 2** $SN_j | \equiv K_{GSN_j}$
- **Goal 6** $GWN | \equiv U_i | \sim ID_{SN_j}$
- **Goal 3** $GWN | \equiv U_i | \equiv T_{U_i}$
- **Goal 7** $SN_j | \equiv GWN | \sim ID_{U_i}$
- **Goal 4** $SN_j | \equiv GWN | \equiv T_{GWN}$
- **Goal 8** $U_i | \equiv GWN | \sim Y_{SN_j}$

**Message 1** $U_i \rightarrow GWN : ID_{U_i}, X_{U_i}, \langle ID_{SN_j} || T_{U_i} \rangle X_{U_i}$

**Message 2** $GWN \rightarrow SN_j : \langle ID_{U_i} || sk || \langle ID_{SN_j} || Y_{SN_j} || T_{GWN} \rangle X_{U_i} || T_{GWN} \rangle X_{U_i}$

**Message 3** $SN_j \rightarrow U_i : \langle ID_{SN_j} || Y_{SN_j} || T_{GWN} \rangle X_{U_i}$

Hypotheses: Some important assumptions (as shown in Table 23) about the initial state are made to analyze the proposed scheme.

Now, based on the hypothesis as described in Table 23 and the rules of the BAN logic, we validate that the proposed protocol can accomplish the intended goals and the clear explanations are as follows:
Table 23. Hypotheses: The assumptions made to analyze the proposed scheme.

1. Derivation of user $U_i$'s trusts on the truth of secret information $K_{U_i}$:

   \[
   \frac{U_i \equiv GWN \Rightarrow K_{U_i}, U_i \equiv GWN \equiv K_{U_i}}{X'_{U_i}}
   \]

   That is, if $U_i$ believes that GWN has jurisdiction over $K_{U_i}$ then $U_i$ trusts GWN on the truth of $K_{U_i}$. Therefore, we achieve Goal 1.

2. Derivation of sensor node $SN_j$'s trusts on the truth of secret information $K_{GSN_j}$:

   \[
   \frac{SN_j \equiv GWN \Rightarrow K_{GSN_j}, SN_j \equiv GWN \equiv K_{GSN_j}}{K_{GSN_j}}
   \]

   That is, if sensor node $SN_j$ believes that the gateway node GWN has jurisdiction over $K_{GSN_j}$ then $SN_j$ trusts GWN on the truth of $K_{GSN_j}$. Therefore, we achieve Goal 2.

3. Verification of freshness of user's time-stamp $T_{U_i}$ on the gateway node GWN (using message-meaning and nonce verification rule):

   \[
   \frac{X'_{U_i}}{GWN \equiv U_i \Rightarrow GWN \equiv \langle T_{U_i} \rangle X'_{U_i}}
   \]

   That is, if GWN believes the secret $X'_{U_i}$ is shared with $U_i$ and sees $< T_{U_i} > X'_{U_i}$, then GWN believes $U_i$ once said $T_{U_i}$.

4. Verification of freshness of gateway node's time-stamp $T_{GWN}$ on the sensor node $SN_j$ (using message-meaning and nonce verification rule):
Information 2017, 8, 136

That is, if \( S_{N_j} \) believes the secret \( K_{GSN_j} \) is shared with \( GWN \) and sees \( <T_{GWN}>_{K_{GSN_j}} \), then \( S_{N_j} \) believes \( GWN \) once said \( T_{GWN} \).

\[
\frac{S_{N_j}| \equiv GWN \iff S_{N_j}, S_{N_j} \triangleleft (T_{GWN})_{K_{GSN_j}}}{S_{N_j}| \equiv GWN| \sim T_{GWN}} \quad \text{(Based on message-meaning rule)}
\]

That is, if \( S_{N_j} \) believes that the time-stamp \( T_{GWN} \) is fresh and \( GWN \) once said \( T_{GWN} \), then \( S_{N_j} \) believes \( GWN \) believes \( T_{GWN} \). Therefore, we achieve Goal 4.

5. Verification of freshness of gateway node’s time-stamp \( T_{GWN} \) on user \( U_i \) (using message-meaning and nonce verification rule):

\[
\frac{X_{U_i} \equiv GWN \iff GWN, U_i \triangleleft (T_{GWN})_{X_{U_i}}}{U_i| \equiv GWN| \sim T_{GWN}} \quad \text{(Based on message-meaning rule)}
\]

That is, if \( U_i \) believes the secret \( X_{U_i}' \) is shared with \( GWN \) and sees \( <ID_{SN_j}>_{X_{U_i}} \), then \( U_i \) believes \( GWN \) once said \( T_{GWN} \).

\[
\frac{U_i| \equiv #(T_{GWN}), U_i| \equiv GWN| \sim T_{GWN}}{U_i| \equiv GWN| \equiv T_{GWN}} \quad \text{(Based on nonce-verification rule)}
\]

That is, if \( U_i \) believes that the time-stamp \( T_{GWN} \) is fresh and \( GWN \) once said \( T_{GWN} \), then \( U_i \) believes \( GWN \) believes \( T_{GWN} \). Therefore, we achieve Goal 5.

6. Verification of sensor node’s identity \( ID_{SN_j} \) on the gateway node \( GWN \):

\[
\frac{X_{U_i}' \equiv GWN \iff GWN, GWN \triangleleft (ID_{SN_j})_{X_{U_i}'}}{GWN| \equiv U_i| \sim ID_{SN_j}} \quad \text{(Based on message-meaning rule)}
\]

That is, if \( GWN \) believes the secret \( X_{U_i}' \) is shared with \( U_i \) and sees \( <ID_{SN_j}>_{X_{U_i}'} \), then \( GWN \) believes \( U_i \) once said \( ID_{SN_j} \). Therefore, we achieve Goal 6.

7. Verification of user’s identity \( ID_{U_i} \) on the sensor node \( S_{N_j} \):

\[
\frac{K_{GSN_j} \equiv GWN \iff S_{N_j}, S_{N_j} \triangleleft (ID_{U_i})_{K_{GSN_j}}}{S_{N_j}| \equiv GWN| \sim ID_{U_i}} \quad \text{(Based on message-meaning rule)}
\]

That is, if \( S_{N_j} \) believes the secret \( K_{GSN_j} \) is shared with \( GWN \) and sees \( <ID_{SN_j}>_{X_{U_i}'} \), then \( GWN \) believes \( U_i \) once said \( ID_{U_i} \). Therefore, we achieve Goal 7.

8. Verification of the public key \( Y_{SN_j} \) by user \( U_i \):

\[
\frac{X_{U_i}' \equiv GWN \iff U_i, U_i \triangleleft <Y_{SN_j}>_{X_{U_i}'}}{U_i| \equiv GWN| \sim Y_{SN_j}} \quad \text{(Based on message-meaning rule)}
\]
That is, if $U_i$ believes the secret $X_U'$ is shared with $GWN$ and sees $<y_{SN_j}, X_U'>$, then $U_i$ believes $GWN$ once said $Y_{SN_j}$. Therefore, we achieve Goal 8.

7. Comparative Study Based on Security Features and Computational Overhead

7.1. Relative Security Analysis

Our comparative analysis of security features is based on the popular features which need to be considered and resistant against well-known attacks. Table 24 shows that our scheme overcomes the major attacks and provides more security.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resist stolen smart card attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Resists Replay attack</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resists Man-in-the-middle attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resists user impersonation attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resists sensor impersonation attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resists insider attack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offers mutual authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offers biometric data updating</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offers secure password updating</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offers formal security analysis</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

7.2. Relative Performance Based on Computational Cost

The execution time as considered in [35,36], for the different cryptographic operation (performed by user $U_i$ and the gateway node $GWN$ with a computer system having windows 7 operating system, Intel (R) core (TM) 2 Quad CPU Q8300, @2.50 Hz processor, and 2 GB RAM) are listed in following Table 25. We assumed the time for executing a fuzzy extractor is the same as that for executing a hash function because the fuzzy extractor [27] can be constructed from universal hash functions or error-correcting codes requiring only lightweight operations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Operation</th>
<th>Time Taken (in Millisecond)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_h$</td>
<td>One-way cryptographic hash function</td>
<td>0.5</td>
</tr>
<tr>
<td>$T_e$</td>
<td>Elliptic curve point multiplication</td>
<td>50.3</td>
</tr>
<tr>
<td>$T_f$</td>
<td>Fuzzy extractor used in biometric verification</td>
<td>0.5</td>
</tr>
<tr>
<td>$T_E$</td>
<td>Symmetric key encryption/decryption</td>
<td>8.7</td>
</tr>
</tbody>
</table>

The computational time and energy consumed by the various cryptographic operations (performed by MicaZ sensor node $SN_j$ with 8-bit ATmega128L Atmel processor, 4 K bytes ROM, 128 K bytes ROM, 512 K bytes EEPROM, 2 AA battery with TinyOS [37] and nesC [38] programming language) are listed in following Table 26.

The comparison of user authentication protocols based on computational cost is shown in Table 27. In the proposed protocol, the registration phase has computation costs $T_h \approx 0.50$ millisecond and $4T_h + T_e \approx (4 \times 0.50 + 50.3) = 52.30$ millisecond associated with $U_i$ and $GWN$ respectively; the authenticated session key establishment phase has computational costs $3T_h \approx 1.50$ millisecond, $3T_h + T_e \approx (1.50 + 50.3) = 51.80$ millisecond and $TS_E \approx 5.05$ millisecond associated with $U_i$. 
GWN and SN, respectively. Similarly the computational cost for Das et al. [17], Choi et al. [21], Park et al. [22] and Moon et al.’s [23] schemes are evaluated, represented and compared in Table 27.

Table 26. Execution time and energy consumption on MicaZ sensor node for cryptographic operations.

<table>
<thead>
<tr>
<th>Function</th>
<th>Time (in millisecond)</th>
<th>Energy (in µ Joule)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric Encryption and Decryption (AES-128)</td>
<td>$T_{SE} \approx 5.05$</td>
<td>121.2</td>
</tr>
<tr>
<td>Hashing (SHA-1)</td>
<td>$T_{Sh} \approx 3.63$</td>
<td>87.12</td>
</tr>
<tr>
<td>Elliptic curve Fixed Point Multiplication (MoTE ECC-160)</td>
<td>$T_{Se} \approx 370$</td>
<td>8880</td>
</tr>
</tbody>
</table>

Table 27. Comparison of protocols based on computational cost.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Registration Phase Time (in millisecond)</th>
<th>Authentication and Session Key Establishment Phase Time (in millisecond)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.K. Das [17]</td>
<td>$4T_h + T_f$ 2T_h 6T_h + T_f 3T_h + 2T_e $2TS_h + TS_E$</td>
<td>$\approx 2.50$ 1.00 $12.20$ $18.90$ $\approx 12.31$</td>
</tr>
<tr>
<td>Choi et al. [21]</td>
<td>$T_h + T_f$ 3T_h 10T_h + T_f 10T_h + 2T_e $6TS_h + TS_E$</td>
<td>$\approx 1.00$ 1.50 $114.80$ $22.40$ $\approx 766.83$</td>
</tr>
<tr>
<td>Park et al. [22]</td>
<td>$T_h + T_f$ 5T_h 10T_h + T_f 11T_h $4TS_h + 2TS_e$</td>
<td>$\approx 1.00$ 2.50 $106.10$ $6.50$ $\approx 754.52$</td>
</tr>
<tr>
<td>Moon et al. [23]</td>
<td>$T_h + T_f$ 3T_h + T_e 6T_h + T_f 6T_h + T_e + T_e $4TS_h + TS_E$</td>
<td>$\approx 1.00$ 51.80 $53.80$ $62$ $\approx 759.57$</td>
</tr>
<tr>
<td>Proposed Protocol</td>
<td>$T_h$ 4T_h + T_e 3T_h 3T_h + T_e $TS_E$</td>
<td>$\approx 0.50$ 52.30 $1.50$ $51.80$ $\approx 5.05$</td>
</tr>
</tbody>
</table>

This comparison indicates that the execution time for the sensor node is very less (because we shifted the overload of performance of elliptic curve point multiplication from sensor node to the gateway node with improved security features) in the proposed protocol.

The energy consumption of the cryptographic operations on the sensor node is evaluated based on the following equation:

$$
\text{Energy} = \text{Voltage} \times \text{Current} \times \text{Times}
$$

where current = 8 Milliampere and Voltage = 3.0 Volts for the micaZ sensor node with AA batteries. Therefore, the energy consumption for Das et al. [17], Choi et al. [21], Park et al. [22] and Moon et al.’s [23] schemes are $(8 \times 3.0 \times (2TS_h + TS_E)) = 295.44$, $(8 \times 3.0 \times (6TS_h + TS_E + 2TS_e)) = 18,403.92$, $(8 \times 3.0 \times (4TS_h + 2TS_e)) = 18,108.48$ and $(8 \times 3.0 \times (4TS_h + TS_E)) = 18,229.68$ respectively. For the proposed protocol the energy consumption is $(8 \times 3.0 \times (TS_E) = 5.05)$. The comparison of user authentication protocols based on energy consumption is shown in Table 28 which illustrate that the proposed protocol consumes less energy compared to other existing protocols.

For the comparative analysis of communicational overhead, we assume that $ID_{UI}$, message request $req$, message response $R/RM$, encrypted message $Enc_k[s]$, time-stamp $T_{UL}/T_{GWN}/T_{SN}$, hash function $h(.)$ and the point on elliptic curve take 160, 32, 32, 128, 32, 160 and 160 bits respectively. In our proposed protocol, during the authentication and session key establishment phase, the message $ID_{UL}, X_{UL}, \alpha$ requires $(160 + 160 + 128 = 448)$ bits, whereas the messages $(\beta)$ and $\alpha$ require $(128 + 128 = 256)$ bits. As a result, the total communication overhead of our proposed protocol becomes 713 bits based on 3 communicated messages. For A.K.Das’s [17] protocol, in the login phase, the message $(ID_{UL}, req)$ requires $(160 + 32) = 192$ bits, whereas in the authentication and key agreement phase, the messages $R, Enc_{k1}(R, T_1, ID_{SN}), (ID_{UL}, Y_i)$ and $(h(SK_{ij}), T_3)$ require 32, 128, 288,
and 352 bits, respectively. As a result, the total communication overhead of A.K.Das’s scheme becomes 832 bits. Similarly the communicational overhead for Choi et al. [21], Park et al. [22] and Moon et al.’s [23] schemes are evaluated, represented and compared in Table 29. The comparative analysis of Table 29 illustrates that the proposed protocol has less communication overhead (which saves communication energy and bandwidth) compared to other existing protocols.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>295.44</td>
<td>18,403.92</td>
<td>18,108.48</td>
<td>18,229.68</td>
<td>121.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>832</td>
<td>1504</td>
<td>1696</td>
<td>1920</td>
<td>713</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

8. Comprehensive Analysis and Lessons Learnt

The security analysis of existing user authentication protocols of the literature demonstrates that the protocols are vulnerable to various attacks like user impersonation attack, sensor node impersonation attack and attacks based on legitimate users. The performance analysis illustrates that the existing protocols are inefficient considering the computational cost. Whereas, the comparative security and performance analysis indicate that our proposed protocol is secure against stolen smart card attack, user impersonation attack, sensor node impersonation attack, sensor node capture attack, replay attack, man-in-the-middle attack. The proposed authentication protocol provides various security features such as mutual authentication, three-factor authentication, secure password and biometric information update, confidentiality, integrity, freshness. The proposed protocol is efficient concerning the computational cost of the resource-constrained sensor nodes, and it saves communication energy, bandwidth. As a result, the protocol is appropriate for applications of resource-constrained ubiquitous computing devices. Therefore, the proposed protocol can be used in various real-world applications consisting of resource constraint sensor devices of WSNs and IoT where bio-metric based secure user authentication and efficient session key establishment is required. The proposed protocol can be used for the implementation of bio-metric based secure authentic banking and financial transactions using the smart card, automated teller machines (ATM), point-of-sale (POS) machines.

9. Conclusions and Future Work

In this paper, we have discussed the security issues involved with the sensor nodes of WSNs and performed the security analysis of various existing protocols of user authentication for WSNs. We have proposed an efficient user authentication, session key establishment protocol for WSNs and IoT using the smart card, fuzzy extractor, ECDH techniques. We have presented security proof using random oracle model and BAN logic to ensure the correctness of various security features involved in the proposed protocol. Afterwards, we have performed the security analysis and verification using well-known and robust tools such as AVISPA and Scyther. Through the precise security analysis using mathematical functions and simulation tools, we have demonstrated that the proposed protocol fulfills the desirable security requirements and withstands the security drawbacks found in existing protocols of user authentication for WSNs. Finally, we have presented
the comparative analysis of our protocol with other existing protocols based on security features and computational overhead which justify that our proposed protocol is secure, efficient and suitable for WSNs/IoT. In future, we would like to propose hyper-elliptic curve cryptography based authenticated key exchange protocol suitable for WSNs and IoT.

Acknowledgments: This work was supported by IDRBT.

Author Contributions: The authors worked jointly in the security analysis and development of the proposed protocol.

Conflicts of Interest: The authors declare no conflict of interest.

References


