information

Article

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

Anup Kumar Maurya [1,2,*] and V. N. Sastry [1]

[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.

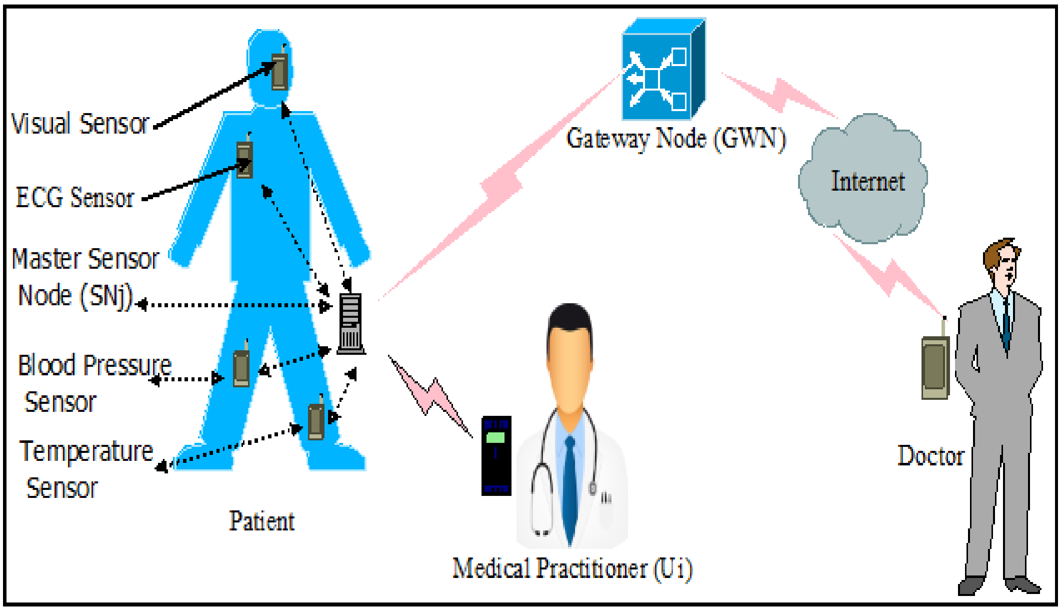


Figure 1. Wireless body area network (WBAN).

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

In 2002, Akyildiz et al. [1] explored many significant aspects of WSNs and discussed critical open research issues of WSNs. Afterwards, several user authentications and session key agreement mechanism for WSNs have been proposed. Unfortunately, many of them still suffer from various security vulnerability. In 2004, Benenson et al. [3] proposed a user authentication and access control mechanism for WSNs. Consequently, Watro et al. [4] (in 2004) developed a public-key (RSA) based user authentication scheme TinyPK using Diffie-Hellman key exchange mechanism which provides mutual authentication and withstand sensor node impersonation attack. Subsequently (in 2005), Benenson et al. [5] designed an elliptic curve cryptography based user authentication system. In 2006, Wong et al. [6] declared that Benenson et al.'s [5] scheme is resistless to denial of service and impersonation attacks. Then, Wong et al. [6] designed a secure hash function based authentication scheme to enhance the security features but it does not support mutual authenticity and session key establishment between the user and sensor node. However, in 2007, Tseng et al. [7] specified that Watro et al.'s [4] and Wong et al.'s [6] schemes exhibit replay and forgery attack. Further, Tseng et al. improved Wong et al.'s scheme and recommended password update mechanism. In 2008, Lee [8] revealed that Wong et al. [6] scheme exhibit more computational overhead on sensor node compared to gateway node and proposed an improved authentication scheme by fixing the security drawbacks of Wong et al. scheme with less computation overhead of sensor node. Later, L.C. Ko [9] indicated that Tseng et al.'s scheme does not provide mutual authentication. Then, L.C. Ko [9] proposed mutual authenticity and time-stamp based user authentication scheme in 2008. In 2009, Vaidya et al. [10] elaborated mutual authentication scheme with formal verification. In 2009, Das [11] developed a secure mechanism to provide authenticity using smart card and user's password (two-factor) but it does not offer session key between the user and sensor node. In 2010, Khan-Alghathbar (2010) [12] identified the gateway node bypass attack, insider attack and lack of password update mechanism in Das's [11] scheme and improved Das's scheme by including password update and mutual authentication technique.

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.

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

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 SN_j 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 $h : In \rightarrow Out$, with a binary string $s \in In \{0,1\}^*$ of arbitrary length as input and a binary string $d \in Out \{0,1\}^m$ of fixed length m as an output, is a secure hash function if the following conditions holds:*

- *$\mathcal{A}$'s advantage to find the collision $Adv^h_{\mathcal{A}}(t_1) = Pr[(s,s') \leftarrow_R \mathcal{A} : s \neq s', h(s) = h(s')]$ and*
- *$Adv^h_{\mathcal{A}}(t_1) \leq \tau$, for any sufficiently small $\tau > 0$.*

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

Definition 2. *Secure Encryption Scheme [25]: For any probabilistic, polynomial time adversary $\mathcal{A}$, an encryption algorithm Enc is said to be IND-CPA (indistinguishability of encryption and chosen plaintext attack) secure if $Adv^{IND-CPA}_{Enc,\mathcal{A}}$ is negligible. Where $Adv^{IND-CPA}_{Enc,\mathcal{A}}(t_2) = 2Pr[\mathcal{A} \leftarrow O_k; (b_0, b_1 \leftarrow \mathcal{A}); \tau \leftarrow_R 0,1; \gamma \leftarrow_R O_k(b_\tau) : \mathcal{A}(\gamma) = \tau] - 1$ denotes the advantage function of $\mathcal{A}$ and $\tau \leftarrow_R \{0,1\}$ denotes that the bit τ 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 $\mathcal{O}$ of infinity, where $4a^3 + 27b^2 \neq 0 \ mod \ p$. 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., $G = (P)$, 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 $\mathcal{A}$ who knows X_{U_i} and Y_{SN_j}. i.e.,

The advantage in finding r_{U_i} is defined by $Adv^{ECDH}_{\mathcal{A}}(t_3) = Pr[(r_{U_i}, P) \leftarrow_R \mathcal{A} : X_{U_i} = r_{U_i} \times P]$. Where $Adv^{ECDH}_{\mathcal{A}}(t_3) \leq \tau$, for any sufficient small $\tau > 0$ and $(r_{U_i}, P) \leftarrow_R \mathcal{A}$ means the pair (r_{U_i}, P) is randomly selected by $\mathcal{A}$ 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 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:*

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

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

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 follwing 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 ek_i. Then, U_i transmits $\langle ID_{U_i}, RPW_i, ek_i \rangle$ to GWN using secure communication channel.

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

Step 1: For User (U_i)	**Step 2: For Gateway (GWN)**
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 ek_i. Then, U_i transmits $\langle ID_{U_i}, RPW_i, ek_i \rangle$ to GWN $\xrightarrow[ViaSecureChannel]{}$	After receiving the message $\langle ID_{U_i}, RPW_i, ek_i \rangle$, the gateway node GWN generates 1024 bit key X_s, evaluates $f_i = h(ID_{U_i} \oplus h(X_s))$, and stores $(h(\dot{}), Gen(\dot{}), Rep(\dot{}), f_i, \mathcal{T})$ into SC_i Finally, GWN sends $\langle SC_i \rangle$ to U_i $\xleftarrow[ViaSecureChannel]{}$ Then, GWN stores ek_i related to ID_{U_i}
Step 3: For User (U_i)	
U_i evaluates $Gen(B_i) = (\sigma_i, \tau_i)$, $f_i^* = f_i \oplus h(ID_{U_i}\|\|\sigma_i\|\|K)$, $r_i = h(ID_{U_i}\|\|\sigma_i) \oplus K$, $e_i = h(ID_{U_i}\|\|RPW_i\|\|\sigma_i)$, and $BE_i = h(ID_{U_i}\|\|\sigma_i) \oplus ek_i$. Then, U_i replaces f_i with f_i^* in SC_i. Finally, U_i stores e_i, τ_i, BE_i, r_i into SC_i.	

Table 3. Login, authentication and key sharing phase of A.K.Das's scheme.

Step 1: For User (U_i)	**Step 2: For Gateway (GWN)**
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 $\sigma'_i = Rep(B_i, \tau_i), K' = r_i \oplus h(ID_{U_i} \| \| \sigma'_i)$, $RPW'_i = h(ID_{U_i} \| \| PW_{U_i} \| \| K')$, $e'_i = h(ID_{U_i} \| \| RPW'_i \| \| \sigma'_i)$ **if** $e'_i = e_i$ **then** U_i transmits $\langle ID_{U_i}, req \rangle$ to GWN $\xrightarrow[\textit{ViaPublicChannel}]{}$ **else** The user U_i terminates this phase	After receiving the message $\langle ID_{U_i}, req \rangle$, **if** ID_{U_i} *is valid* **then** GWN sends a Random challenge R to U_i $\xleftarrow[\textit{ViaPublicChannel}]{}$ **else** GWN aborts this phase
Step 3: For User (U_i)	**Step 4: For Gateway (GWN)**

After receiving the Random challenge R, U_i evaluates $ek_i = BE_i \oplus h(ID_{U_i} \| \| \sigma'_i)$. Then,
U_i transmits $\left\langle Enc_{ek_i}(R, T_1, ID_{SN_j}) \right\rangle$ to GWN $\xrightarrow[\textit{ViaPublicChannel}]{}$
	GWN evaluates R, T_1, ID_{SN_j} using ek_i.
	if T_1 *and R are valid* **then**
	  GWN computes
	  $f^*_i = h(ID_{U_i} \oplus h(X_s)), f^{**}_i = h(ID_{SN_j} \| \| f^*_i)$
	  and $Y_j = Enc_{K_j}[ID_{U_i}, ID_{SN_j}, T_1, T_2, f^{**}_i]$.
	  Then, GWN transmits $\langle ID_{U_i}, Y_j \rangle$ to SN_j $\xrightarrow[\textit{ViaPublicChannel}]{}$
	else
	  Reject U_i

Step 5: For Sensor Node (SN_j)	**Step 6: For User (U_i)**

SN_j Retrieves $(ID_{U_i}, ID_{SN_j}, T_1, T_2, f^{**}_i)$ as $(ID''_{U_i}, ID''_{SN_j}, T''_1, T''_2, f''_i)$.
if T_2 *and* ID_{U_i} *are valid* **then**
  SN_j Evaluate the session key
  $SK_{ij} = h(f''_i \| \| ID_{U_i} \| \| ID_{SN_j} \| \| T''_1, T_3)$
  SN_j sends $h(SK_{ij}), T_3$ to U_i $\xrightarrow[\textit{ViaPublicChannel}]{}$
else
  Reject U_i
Store SK_{ij}
	if T_3 *is valid* **then**
	  Computes $f'_i = f^*_i \oplus h(\sigma'_i \| \| ID_{U_i} \| \| K')$,
	  $f''_i = h(ID_{SN_j} \| \| f'_i)$,
	  $SK'_{ij} = h(f''_i \| \| ID_{U_i} \| \| ID_{SN_j} \| \| T_1 \| \| T_3)$
	  if $h(SK'_{ij}) = h(SK_{ij})$ **then**
	    U_i Stores SK'_{ij}
	  else
	    Reject U_i
	else
	  Reject U_i

Step DR2: After receiving the message $\langle ID_{U_i}, RPW_i, ek_i \rangle$, the gateway node GWN generates 1024 bit key X_s, evaluates $f_i = h(ID_{U_i} \oplus h(X_s))$ and stores $(h(\dot{)}, Gen(\dot{)}, Rep(\dot{)}, f_i, \mathcal{T})$ 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 $Gen(B_i) = (\sigma_i, \tau_i)$, $f^*_i = f_i \oplus h(ID_{U_i}||\sigma_i||K)$, $r_i = h(ID_{U_i}||\sigma_i) \oplus K$, $e_i = h(ID_{U_i}||RPW_i||\sigma_i)$, and $BE_i = h(ID_{U_i}||\sigma_i) \oplus ek_i$. Finally, U_i replaces f_i with f^*_i in SC_i and stores e_i, τ_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 $\sigma_i' = Rep(B_i, \tau_i)$, $K' = r_i \oplus h(ID_{U_i}||\sigma_i')$, $RPW_i' = h(ID_{U_i}||PW_{U_i}||K')$, $e_i' = h(ID_{U_i}||RPW_i'||\sigma_i')$. If $e_i' = e_i$, U_i transmits $\langle ID_{U_i}, req\rangle$ to GWN via public communication channel. Otherwise, U_i aborts this phase.

Step DA2: After receiving the message $\langle ID_{U_i}, req\rangle$, GWN verifies the message. If 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 $ek_i = BE_i \oplus h(ID_{U_i}||\sigma_i')$. Finds the current time-stamp T_1. Then, U_i transmits $\left\langle Enc_{ek_i}(R, T_1, ID_{SN_j})\right\rangle$ to GWN via public communication channel.

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

Step DA5: SN_j retrieves $(ID_{U_i}, ID_{SN_j}, T_1, T_2, f_i^{**})$ as $(ID_{U_i}'', ID_{SN_j}'', T_1'', T_2'', f_i'')$ using decryption operation on $\langle ID_{U_i}, Y_j\rangle$ based on key K_j. If T_2'' is fresh and ID_{U_i} is valid, SN_j finds the current time-stamp T_3 and evaluates the session key $SK_{ij} = h(f_i''||ID_{U_i}||ID_{SN_j}||T_1'', T_3)$. 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 $f_i' = f_i^* \oplus h(\sigma_i'||ID_{U_i}||K')$, $f_i'' = h(ID_{SN_j}||f_i')$, $SK_{ij}' = h(f_i''||ID_{U_i}||ID_{SN_j}||T_1||T_3)$. 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 $\mathcal{A}$ ascertains the value of $\{\tau_i, e_i, r_i, BE_i, f^*, h(.), Gen(.), Rep(.), \mathcal{T}\}$ from stolen SC_i by measuring the power consumption of smart card [24]. Then, $\mathcal{A}$ computes: $BE_i \oplus r_i = [h(ID_{U_i} \parallel \sigma_i) \oplus K] \oplus [h(ID_{U_i} \parallel \sigma_i) \oplus ek_i] = K \oplus ek_i$.

Afterwards, the adversary $\mathcal{A}$ find out the value of K and ek_i by implementing one of the following three mechanism:

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 $E_{ek_i}(R, T, ID_{SN_j})$ and implements the known plain text attack to find out the value of ek_i. Thereafter, $\mathcal{A}$ 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 $d(B_i, B_i^{'}) \leq \mathcal{T}$) and find out the value of $\sigma_i = Rep(B_i^{'}, \tau_i)$. Eavesdrops the value of ID_{U_i} from public communication channel and then evaluates the value of $ek_i = BE_i \oplus h(ID_{U_i} \parallel \sigma_i)$, $K = r_i \oplus h(ID_{U_i} \parallel \sigma_i)$. It is possible, because ek_i is not password PW_{U_i} protected.

Subsequently, $\mathcal{A}$ chooses its own identity ID_A, password PW_A, biometric information B_A and computes:

$RPW_A = h(ID_A \parallel K \parallel PW_A)$, $Gen(B_A) = (\sigma_A, \tau_A)$, $e_A = h(ID_A \parallel RPW_A \parallel \sigma_A)$, $r_A = h(ID_A \parallel \sigma_A) \oplus K$ and $BE_A = h(ID_A \parallel \sigma_A) \oplus ek_i$.

Finally, $\mathcal{A}$ replaces the information $\{\tau_i, e_i, r_i, BE_i, f^*, h(), Gen(.), Rep(.), \mathcal{T}\}$ of SC_i with $\{\tau_A, e_A, r_A, BE_A, f^*, h(), Gen(.), Rep(.), \mathcal{T}\}$ respectively.

The login phase of the adversary $\mathcal{A}$ is as follows:

- $\mathcal{A}$ insert 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 $e_A^{'} = h(ID_A \parallel RPW_A^{'} \parallel \sigma_A^{'})$. Then, it verifies if $e_A^{'} = e_A$. It would be true i.e., both the password and bio-metric verification would be correct.
- 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 $\mathcal{A}$.
- $\mathcal{A}$ select the login sensor node SN_j and sends $\langle E_{ek_i}(R, T_1, ID_{SN_j}) \rangle$ to GWN.
- After receiving $\langle E_{ek_i}(R, T_1, ID_{SN_j}) \rangle$, GWN decrypt it using ek_i and verifies the validity of T_1 and R. Subsequently, GWN computes $f_i^* = h(ID_{U_i} \oplus h(X_s)), f_i^{**} = h(ID_{SN_j} \parallel f_i^*), Y_j = E_{K_j}[ID_{U_i}, ID_{SN_j}, T_1, T_2, f_i^{**}]$ 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 $f_i' = f_i^* \oplus h(\sigma_i' || ID_{U_i} || K')$ using ID_{U_i}, stolen bio-metric and evaluated K. It is possible because f_i' has no password protection.
- Finally, $\mathcal{A}$ computes $f_i^{''} = h(ID_{SN_j} || f_i')$ 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 $h(ID_{SN_j} || y)$ 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 $M_{U_i} = h(ID_{U_i} || x) \oplus A_i, N_{U_i} = h(ID_{U_i} \oplus x) \oplus A_i, V_{U_i} = h(ID_{U_i} || A_i)$. Then, GWN stores $\langle ID_{U_i}, M_{U_i}, N_{U_i}, V_{U_i}, h(.) \rangle$ 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 τ_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.

Step 1: For User (U_i)

```
The registered user $U_i$ inputs $ID_{U_i}$, $B_i^{'}$
and computes $\sigma_i^{'} = Rep(B_i^{'}, \tau_i)$, $A_i^{'} = h(\sigma_i^{'})$,
$V_i' = h(ID_{U_i}||A_i')$,
if $V_i = V_i'$ then
    $U_i$ generates random number $r_i$
    $X_i = r_i \times P$, $D_i = M_i \oplus A_i'$,
    $h(x||y) = N_i \oplus A_i'$
    Finds out current time-stamp $T_i$ and
    computes $k_i = h(D_i||T_i)$,
    $C_i = E_{k_i}(ID_{U_i}||X_i)$,
    $AID_i = ID_{U_i} \oplus h(h(x||y)||T_i)$,
    $W_i = h(h(x||y)||AID_i||X_i||C_i||T_i)$

    Then, $U_i$ constructs a message
    $M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle$
    $\xrightarrow[ViaPublicChannel]{\text{Finally, } U_i \text{ transmits } M_1 \text{ to } GWN}$
else
    Abort this phase.
```

Step 2: For Gateway (GWN)

```
if $(T' - T_i) \leq \Delta T$ then
    if $W_i = h(h(x||y)||AID_i||X||C_i||T_i)$ then
        $GWN$ computes
        $ID'_{U_i} = AID_i \oplus h(h(x||y)||T_i)$,
        $D_i' = h(ID'_{U_i}||x), k_i' = h(D_i'||T_i)$,
        $ID''_{U_i}||X_i' = D_{k_i'}(C_i)$
        if $ID'_{U_i} = ID''_{U_i}$ then
            $GWN$ finds its current time-stamp
            $T_g$ and computes
            $k_g = h(h(SID_j||y)||T_g)$,
            $C_g = Enc_{k_g}(AID_i||X_i')$,
            $W_g = h(h(SID_j||y)||AID_i||C_g||T_g)$.

            Then, $GWN$ construct the message
            $M_2 = \langle AID_i, C_g, T_g, W_g \rangle$. Finally,
            $\xrightarrow[ViaPublicChannel]{GWN \text{ transmits } M_2 \text{ to } SN_j}$
        else
            $GWN$ aborts this phase.
    else
        $GWN$ aborts this phase.
else
    $GWN$ abort this phase.
```

Step 3: For Sensor Node (SN_j)

```
if $(T'' - T_g) \leq \Delta T$ then
    if $W_g = h(h(SID_j||y)||AID_i||C_g||T_g)$ then
        $SN_j$ computes $k_g' = h(h(SID_i||y)||T_g)$,
        $AID_i'||X' = Dec_{k_g'}(C_g)$,
        if $AID_i = AID_i'$ then
            Generates random number $r_s$
            $K_{SU} = r_s \times X_i', Y_i = r_s \times P$
            Computes $sk = h(AID_i||K_{SU}||T_s)$
            Find the current time-stamp $T_s$
            and computes $RM =$ Query
            response,
            $V_s = h(AID_i'||X_i'||Y_i||RM||T_s)$,
            $M_3 = \langle RM, Y_i, V_s, T_s \rangle$. Finally,
            $\xrightarrow[ViaPublicChannel]{SN_j \text{ transmits } M_3 \text{ to } U_i}$
        else
            Abort this phase
    else
        Abort this phase
else
    Abort this phase
```

Step 4: For User (U_i)

```
if $(T''' - T_s) \leq \Delta T$ then
    if $V_s = h(AID_i||X_i||Y_i||RM||T_s)$ then
        Then $U_i$ computes $K_{US} = r_u \times Y_i$,
        $sk = h(AID_i||K_{US}||T_s)$,
        Accept $PM$. Where
        $sk = h(AID_i||r_i \times r_s \times P||T_s)$
        {based on ECDH}
    else
        Abort this phase
else
    Abort this phase
```

4.4.1. Attack Based on Legitimate User

In this scheme, a legitimate user U_L can be an adversary $U_{\mathcal{A}}$, 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:

- $U_{\mathcal{A}}$ inputs $ID_{U_{\mathcal{A}}}$, imprints $B^{'}_{\mathcal{A}}$, computes $\sigma^{'}_{\mathcal{A}} = Rep(B^{'}_{\mathcal{A}}, \tau_{\mathcal{A}})$, $A'_{\mathcal{A}} = h(\sigma^{'}_{\mathcal{A}})$, $V'_{\mathcal{A}} = h(ID_{\mathcal{A}}||A'_{\mathcal{A}})$ and finally verifies $V_{\mathcal{A}} = V'_{\mathcal{A}}$,
- If verification succeeds, $U_{\mathcal{A}}$ generate random number $r_{\mathcal{A}}$, and computes $X_{\mathcal{A}} = r_{\mathcal{A}} \times P$, $D_{\mathcal{A}} = M_{\mathcal{A}} \oplus A'_{\mathcal{A}}, h(x||y) = N_{\mathcal{A}} \oplus A'_{\mathcal{A}}$
- $\mathcal{A}$ intercepts the message $M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle$ of U_i and find out: $ID_{U_i} = AID_i \oplus h(h(x||y)||T_i)$.
- Therefore, we find that Choi et al. scheme does not provide user anonymity i.e., an adversary $\mathcal{A}$ can compute user U_i's identification ID_{U_i}. However, Choi et al. claimed that their protocol provides user anonymity.
- Furthermore $\mathcal{A}$ intercepts the cipher text $C_i = E_{k_i}(ID_{U_i}||X_i)$ and derives the plain-text $(ID_{U_i}||X_i)$, therefore Choi et al. scheme is vulnerable to known plain-text attack.

4.4.2. User Impersonation Attack

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

- $\mathcal{A}$ extracts $\langle ID_{U_i}, M_{U_i}, N_{U_i}, V_{U_i}, h(.), \tau_i \rangle$ from the smart card SC_i of the user U_i and computes $A_i^* = N_i \oplus h(x||y), V_i^* = h(ID_{U_i}||A_i^*)$ and verify the computed V_i^* with the stored V_i.
- $\mathcal{A}$ generates a random number $r_{\mathcal{A}}$, calculates $X_{\mathcal{A}} = r_{\mathcal{A}} \times P$,$D^*_{\mathcal{A}} = M_i \oplus A_i^*$. Find out the current timestamps $T_{\mathcal{A}}$, computes $k_{\mathcal{A}} = h(D_i^*||T_{\mathcal{A}})$, $C_{\mathcal{A}} = E_{k_{\mathcal{A}}}(ID_{U_i}||X_{\mathcal{A}})$, $AID_{\mathcal{A}} = ID_{U_i} \oplus h(h(x||y)||T_{\mathcal{A}})$, $W_{\mathcal{A}} = h(h(x||y)||AID_{\mathcal{A}}||X_{\mathcal{A}}||C_{\mathcal{A}}, T_{\mathcal{A}})$.
- $\mathcal{A}$ sends the message $M_1^{\mathcal{A}} = \langle AID_{\mathcal{A}}, X_{\mathcal{A}}, C_{\mathcal{A}}, T_{\mathcal{A}}, W_{\mathcal{A}} \rangle$ to GWN. Subsequently, $\mathcal{A}$ establishes the session key $sk = h(AID_{\mathcal{A}}||r_{\mathcal{A}} \times r_s \times P)$ 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 $h(ID_{SN_j}||y)$ 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 $\mathcal{A}$ need to have the key $k_{GWN} = h(h(ID_{SN_j}||y)||T_{GWN})$. Although, an adversary $\mathcal{A}$ can impersonate the sensor node SN_j without having k_{GWN} with the help of following steps:

- The adversary $\mathcal{A}$ intercepts the message $M_1 = \langle AID_{U_i}, X_{U_i}, C_{U_i}, T_{U_i}, W_{U_i} \rangle, M$ and $M_2 = \langle AID_{GWN}, C_{GWN}, T_{GWN}, W_{GWN} \rangle$.
- Then, $\mathcal{A}$ generates a random number $r_{\mathcal{A}}$, finds current times-stamp $T_{\mathcal{A}}$ and computes: $K_{\mathcal{A}U} = r_{\mathcal{A}} \times X_{U_i}$, $Y_{\mathcal{A}} = r_{\mathcal{A}} \times P$, $sk = h(AID_{U_i}||K_{\mathcal{A}U}||T_{\mathcal{A}})$, RM =Query response and $V_{\mathcal{A}} = h(AID_{U_i}||X_{U_i}||Y_{\mathcal{A}}||RM||T_{\mathcal{A}})$.

- Afterwards, $\mathcal{A}$ sends $M_3 = \langle RM, Y_{\mathcal{A}}, V_{\mathcal{A}}, T_{\mathcal{A}} \rangle$ to U_i.
- After receiving M_3, U_i computes: $V_{\mathcal{A}} = h(AID_{U_i}||X_{U_i}||Y_{\mathcal{A}}||RM||T_{\mathcal{A}})$. If $V_{\mathcal{A}}* = V_{\mathcal{A}}$, $\mathcal{A}$ computes $K_{U\mathcal{A}} = r_{U_i} \times Y_{\mathcal{A}}, sk = h(AID_{U_i}||K_{U\mathcal{A}}||T_{\mathcal{A}})$.

Therefore, the adversary $\mathcal{A}$ succeeds in impersonating the sensor node SN_j and establishing the session key sk with the user U_i.

Table 5. User registration phase of Park et al.'s protocol.

Step 1: For User (U_i)	**Step 2: For Gateway (GWN)**
U_i selects the identity ID_{U_i}, imprints bio-metric information B_i and computes: $(\sigma_{U_i}, \tau_{U_i}) = Gen(B_i), A_{U_i} = h(\sigma_{U_i})$ $\xrightarrow[ViaSecureChannel]{U_i \text{ transmits } \langle ID_{U_i}, A_{U_i} \rangle \text{ to } GWN}$	GWN computes 1024 bit secret key x and Computes: $M_{U_i} = h(x\|\|y\|\|A_i)$, $N_{U_i} = M_{U_i} \oplus A_{U_i}$, $V_{U_i} = h(ID_{U_i}\|\|A_i)$, $C_{U_i} = Enc_x(A_{U_i}\|\|up_{U_i})$ Store $\langle V_{U_i}, C_{U_i}, N_{U_i}, h(.) \rangle$ into smart card SC_i. $\xleftarrow{GWN \text{ sends smart card } SC_i \text{ to } U_i}$
Step 3: For User U_i	
U_i Inputs τ_{U_i} into the smart card SC_i	

4.6.2. User Impersonation Attack

In Park et al.'s scheme, a legitimate user U_k can be an adversary $U_{\mathcal{A}}$ 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 $\mathcal{A}$ extract the information $\langle V_{U_k}, N_{U_k}, C_{U_k}, h(.), P_{U_k} \rangle$ from the smart card.
- Then, $\mathcal{A}$ imprints its biometric information B'_k and computes $\sigma'_k = Rep(B'_k, P_{U_k})$ and $A'_{U_k} = h(\sigma'_k)$, $M_{U_k} = N_{U_k} \oplus A'_{\mathcal{A}}$.
- Afterwards, $\mathcal{A}$ generates random number $r_{\mathcal{A}}$, selects an identity ID_{U_i} and computes: $X_{\mathcal{A}} = X_{U_k} = r_{U_k} \times P$, $AID_{U_i} = ID_{U_i} \oplus h(M_{\mathcal{A}}||T_{\mathcal{A}})$ and $W_{\mathcal{A}} = W_{U_k} = h(M_{\mathcal{A}}||ID_{U_i}||X_{\mathcal{A}}||T_{\mathcal{A}})$. Finally, $\mathcal{A}$ sends $M_1 = \langle AID_{U_i}, X_{\mathcal{A}}, C_{U_k}, T_{U_k}, W_{U_k} \rangle$ to GWN.
- After receiving M_1, if $(T' - T_{U_k}) \leq \Delta T$, GWN computes $A'_{U_i}||up_{U_k} = Dec_x(C_{U_k})$, $M'_{U_k} = h(x||y||A'_{U_k})$, $ID'_{U_i} = AID_{U_i} \oplus h(M'_{U_k}||T_{U_k})$, $W'_{U_i} = h(M_{U_k}||ID'_{U_i}||X'_{U_k}||T_{U_k})$.
- If $(W_{U_i} = W'_{U_i})$, the GWN finds the current time stamp T_{GWN} and computes: $k_{GWN} = h(h(ID_{SN_j}||y)||T_{GWN})$, $C_{GWN} = Enc_{k_{GWN}}(AID_{U_k}||X_{U_k})$, $W_{GWN} = h(h(ID_{SN_j}||y)||AID_{U_i}||C_{GWN}||T_{GWN})$.
- Finally, GWN sends $M_2 = \langle AID_{U_k}, W_{GWN}, C_{GWN}, T_{GWN} \rangle$ to SN_j.
- After receiving M_2, if $(T''' - T_{GWN}) \leq \Delta T$ and $W_{GWN} = h(h(ID_{SN_j}||y)||AID_{U_k}||C_{GWN}||T_{GWN})$, SN_j computes: $k'_{GWN} = h(h(ID_{SN_j}||y)||T_{GWN})$ and $(AID'_{U_k}||X'_{U_k}) = Dec'_{GWN}(C_{GWN})$.
- If $(AID_{U_k} = AID'_{U_k})$, SN_j generates a random number r_{SN_j} and computes: $K_{SU} = r_{SN_j} \times X'_{U_k}$, $Y_{U_i} = r_{SN_j} \times P$, $sk = h(AID_{U_i}||K_{SU}||T_{SN_j})$, $RM =$ Query Response, $V_{SN_j} = h(AID_{U_k}||X_{U_k}||Y_{U_i}||RM||T_{SWN_j})$.
- Then, SN_j sends $M_3 = \left\langle RM, Y_{U_i}, V_{SN_j}, T_{SN_j} \right\rangle$ to the adversary $\mathcal{A}$.

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

Step 1: For User (U_i)

```
$U_i$ inserts the smart card $SC_i$, inputs $ID_{U_i}$
and imprints $B_i'$. Then, computes
$\sigma_{U_i}' = Rep(B_i', \tau_{U_i})$, $A_{U_i}' = h(\sigma_{U_i}')$,
$V_{U_i}' = h(ID_{U_i}||A_{U_i}')$,
if $V_{U_i} = V_{U_i}'$ then
    Generate random number $r_{U_i}$,
    and computes $X_{U_i} = r_{U_i} \times P$,
    $M_{U_i} = N_{U_i} \oplus A_{U_i}'$,
    Find out current time-stamp $T_{U_i}$
    and computes
    $AID_{U_i} = ID_{U_i} \oplus h(M_{U_i}||T_{U_i})$,
    $W_{U_i} = h(M_{U_i}||ID_{U_i}||X_{U_i}||T_{U_i})$.
    Then, $U_i$ constructs a message
    $M_1 = \langle AID_{U_i}, X_{U_i}, C_{U_i}, T_{U_i}, W_{U_i} \rangle$
    Finally, $U_i$ transmits $M_1$ to $GWN$
    $\xrightarrow[ViaPublicChannel]{}$
else
    Abort this phase.
```

Step 2: For Gateway (GWN)

```
if $(T' - T_{U_i}) \leq \Delta T$ then
    $A_i^* \leftarrow Dec_x(C_{U_i})$, $M_{U_i}^* = h(x||y||A_{U_i}^*)$,
    $ID_{U_i}' = AID_{U_i} \oplus h(M_{U_i}^*||T_{U_i})$,
    $W_{U_i}' = h(M_{U_i}'||ID_{U_i}'||X_{U_i}'||T_{U_i})$ if
    $W_{U_i} = W_{U_i}'$ then
        Find the current time-stamp $T_{GWN}$
        and computes
        $k_{GWN} = h(h(ID_{SN_j}||y)||T_{GWN})$,
        $C_{GWN} = Enc_{k_{GWN}}(AID_i||X_{U_i}')$,
        $W_{GWN} =$
        $h(h(ID_{SN_j}||y)||AID_{U_i}||C_{GWN}||T_{GWN})$.
        Then, $GWN$ constructs the message
        $M_2 = \langle AID_{GWN}, C_{GWN}, T_{GWN}, W_{GWN} \rangle$
        Finally, $GWN$ transmits $M_2$ to $SN_j$
        $\xrightarrow[PublicChannel]{}$
    else
        Abort this phase.
else
    Abort this phase.
```

Step 3: For Sensor Node (SN_j)

```
if $(T'' - T_{GWN} \leq \Delta T$ then
    $k_{GWN}' = h(h(ID_{SN_j}||y)||T_{GWN})$
    $AID_{U_i}'||X_{U_i}' = Dec_{k_{GWN}'}(C_{GWN})$
    if $W_{GWN} =$
    $h(h(ID_{SN_j}||y)||AID_{U_i}||C_{GWN}||T_{GWN})$
    and $(AID_{U_i} = AID_{U_i}')$ then
        $SN_j$ generates random number $r_{SN_j}$,
        computes
        $K_{SU} = r_{SN_j} \times X_{U_i}', Y_{U_i} = r_{SN_j} \times P$
        and $sk = h(AID_{U_i}||K_{SU}||T_{SN_j})$. Then,
        $SN_j$ finds the current time-stamp $T_{SN_j}$,
        computes $RM$ = Query response,
        $V_{SN_j} = h(AID_{U_i}'||X_{U_i}'||Y_{U_i}||RM||T_{SN_j})$,
        $M_3 = \left\langle RM, Y_{U_i}, V_{SN_j}, T_{SN_j} \right\rangle$.
        Finally, $SN_j$ transmits $M_3$ to $U_i$
        $\xrightarrow[PublicChannel]{}$
    else
        Abort this phase.
else
    Abort this phase.
```

Step 4: For User (U_i)

```
if $(T''' - T_{SN_j}) \leq \Delta T$ then
    if $V_{SN_j} = h(AID_{U_i}||X_{U_i}||Y_{U_i}||RM||T_{SN_j})$
    then
        $U_i$ computes $K_{US} = r_{U_i} \times Y_{U_i}$,
        $sk = h(AID_{U_i}||K_{US}||T_{SN_j})$, accepts $RM$
        and establishes the session key
        $sk = h(AID_{U_i}||r_{U_i} \times r_{SN_j} \times P||T_{SN_j})$
        with $SN_j$.
    else
        Abort this phase.
else
    Abort this phase.
```

- After receiving M_3, if $(T'' - T_{SN_j}) \leq \Delta T$, the adversary $\mathcal{A}$ computes: $V'_{SN_j} = h(AID_{U_k}||X_{U_k}||Y_{U_i}||RM||T_{SN_j})$. If $(V_{SN_j} = V'_{SN_j})$, SN_j computes $K_{US} = r_{U_k} \times Y_{U_i}$ and establishes the session key $sk = h(AID_{U_k}||K_{US}||T_{SN_j})$ with sensor node SN_j. Therefore, Park et al.'s scheme is vulnerable to user impersonation attack. Similar attack is possible in Moon et al.'s scheme [23] also, since the value of C_{U_i} in Moon et al.'s scheme can be evaluated using x, y and N_{U_i}.

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 $\mathcal{A}$), hundreds of sensor nodes (these nodes can be captured by $\mathcal{A}$) 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.

Step 1: For User (U_i)	**Step 2: For Gateway (GWN)**
U_i inputs ID_{U_i}, PW_{U_i} and B_i Computes: $Gen(B_i) = (\sigma_{U_i}, \tau_{U_i}), PB_{U_i} = h(PW_{U_i}\|\|\sigma_{U_i})$ $\xrightarrow[ViaSecureChannel]{U_i \text{ transmits } \langle ID_{U_i}, PB_i\rangle \text{ to } GWN}$	GWN computes 1024 bit secret key x and Computes: $K_{U_i} = h(ID_{U_i}\|\|x) \times P$, $A_{U_i} = PB_i \oplus h(ID_{U_i} \oplus x)$, $B_{U_i} = h(ID_{U_i}\|\|PB_i\|\|h(ID_{U_i} \oplus x))$, $W_{U_i} = h(ID_{U_i}\|\|PB_i) \oplus K_{U_i}$ GWN stores the value of P, A_{U_i}, B_{U_i}, W_{U_i} into SC_i. $\xleftarrow{GWN \text{ transmits } \langle SC_i\rangle \text{ to } U_i}$
Step 3: For User (U_i)	
U_i stores $\mathcal{T}, h(), Gen(), Rep()$ and the value of τ_i into SC_i.	

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 $PB_i = h(PW_{U_i}||\sigma_i)$ 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:
$K_{U_i} = h(ID_{U_i}||x) \times P$ (where "×" is the scalar multiplication operator of elliptic curve),
$A_{U_i} = PB_i \oplus h(ID_{U_i} \oplus x)$,
$B_{U_i} = h(ID_{U_i}||PB_i||h(ID_{U_i} \oplus x))$,

$W_{U_i} = h(ID_{U_i}||PB_i) \oplus K_{U_i}$ Finally, the gateway node GWN stores the value of P, A_{U_i}, B_{U_i}, W_{U_i} into the smart card SC_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 $\mathcal{T}, \tau_i$ 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)	**Step 2: For Gateway (GWN)**
U_i inputs ID_{U_i}, PW_{U_i} and $B_i^{'}$. Computes $\sigma_i^{'} = Rep(B_i^{'}, \tau_i)$, $PB_i^{'} = h(PW_{U_i} \Vert \sigma_i^{'}), h^{'}(ID_{U_i} \oplus x) = A_{U_i} \oplus PB_i^{'}$, $B_{U_i}^{'} = h(ID_{U_i} \Vert PB_i^{'} \Vert h^{'}(ID_{U_i} \oplus x))$. **if** $B_{U_i}^{'} = B_{U_i}$ **then** Evaluate $K_{U_i} = W_{U_i} \oplus h^{'}(ID_{U_i} \Vert PB_i^{'})$. Generate $r_{U_i} \in \mathbb{Z}_q^*$. Find current time stamp T_{U_i}, $X_{U_i} = r_{U_i} \times P, X_{U_i}^{'} = r_{U_i} \times K_{U_i}$, $\alpha = Enc_{X_{U_i^{'}}}[ID_{SN_j} \Vert T_{U_i}]$. Construct a message $M_1 = \langle ID_{U_i}, X_{U_i}, \alpha \rangle$ **else** U_i is unauthenticated, abort this phase.	GWN compute $X_{U_i}^{'} = h(ID_{U_i} \Vert x) \times X_{U_i}$, $[ID_{SN_j} \Vert T_{U_i}] = Dec_{X_{U_i^{'}}}[\alpha]$, **if** $T^{\prime} - T_{U_i} \leq \Delta T$ **then** Generates $r_{SN_j} \in \mathbb{Z}_q^*$, 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}, $\beta = Enc_{X_{U_i}^{\prime}}[ID_{SN_j} \Vert Y_{SN_j} \Vert T_{GWN}]$, $\gamma = Enc_{K_{GSN_j}}[ID_{U_i} \Vert sk \Vert \beta \Vert T_{GWN}]$, Construct the message $M_2 = \langle \gamma \rangle$ $\xrightarrow[ViaPublicChannel]{GWN \text{ transmits } M_2 \text{ to } SN_j}$ **else** Replay and energy exhausting attack possible.Abort this phase.

Step 3: For Sensor Node (SN_j)	**Step 4: For User (U_i)**
SN_j computes $[ID_{U_i} \Vert sk \Vert \beta \Vert T_{GWN}] = Dec_{K_{GSN_j}}[\gamma]$, **if** $T^{''} - T_{GWN} \leq \Delta T$ **then** Store the session key sk Construct the message $M_3 = \langle \beta \rangle$ $\xrightarrow[ViaPublicChannel]{SN_j \text{ transmits } M_3 \text{ to } U_i}$ **else** Replay and energy exhausting attack possible. Abort this phase.	U_i computes $[ID_{SN_j} \Vert Y_{SN_j} \Vert T_{GWN}] = D_{X_{U_i}^{\prime}}[\beta]$ **if** $T^{\prime\prime\prime} - T_{GWN} \leq 2\Delta T$ **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}, imprints her noisy biometric information $B_i^{'}$ and computes $\sigma_i^{'} = Rep(B_i^{'}, \tau_i)$ using reproduction function of fuzzy extractor as described in Definition 4. Then, U_i calculates $PB_i^{'} = h(PW_{U_i}||\sigma_i^{'})$, $h^{'}(ID_{U_i} \oplus x) = A_{U_i} \oplus PB_i^{'}$, $B_{U_i}^{'} = h(ID_{U_i}||PB_i^{'}||h^{'}(ID_{U_i} \oplus x))$.

If the equivalent condition $B_{U_i}^{'} = B_{U_i}$ 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 "×" 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 $M_1 = \langle ID_{U_i}, X_{U_i}, \alpha \rangle$ and sends M_1 to the gateway node GWN.

Step A2: After receiving the message M_1, the gateway node GWN compute $X_{U_i}^{'} = h(ID_{U_i}||x) \times X_{U_i}$ and decrypts the cipher text α considering $X_{U_i'}$ as a symmetric key to find: $[ID_{SN_j}||T_{U_i}] = Dec_{X_{U_i'}}[\alpha]$. 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 "×" is the scalar multiplication operator of elliptic curve). Then, the gateway node GWN finds its current time-stamp T_{GWN} and calculates: $\beta = Enc_{X_{U_i}'}[ID_{SN_j}||Y_{SN_j}||T_{GWN}]$, $\gamma = Enc_{K_{GSN_j}}[ID_{U_i}||sk||\beta||T_{GWN}]$. Finally, GWN construct the message $M_2 = \langle \gamma \rangle$ and sends M_2 to the sensor node SN_j.

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

Step A4: After receiving the message M_3, the user U_i decrypts the message β considering X_{U_i}' as a symmetric key and find out: $[ID_{SN_j}||Y_{SN_j}||T_{GWN}] = D_{X_{U_i}'}[\beta]$. 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.

```
$U_i$ inserts $SC_i$ into the card reader and
Inputs $ID_{U_i}, PW_{U_i}, B_i'$.
Then, $U_i$ computes $\sigma_i^{'} = Rep(B_i^{'}, \tau_i)$, {Using fuzzy extractor }
$PB_i^{'} = h(PW_{U_i}||\sigma_i)$, $h'(ID_{U_i} \oplus x) = A_{U_i} \oplus PB_i^{'}$,
$B_i' = h(ID_{U_i}||PB_i^{'}||h'(ID_{U_i} \oplus x))$.
if $B_i' = B_i$. then
    $U_i$ calculates $h(ID_{U_i} \oplus x) = A_{U_i} \oplus PB_i^{'}$, $K_{U_i} = W_{U_i} \oplus h(ID_{U_i}||PB_i^{'})$,
    $U_i$ inputs new $PW_{new_i}, B_{new_i}$,
    Then, $U_i$ computes $Gen(B_{new_i}) = (\sigma_{new_i}, \tau_{new_i})$ {Using fuzzy extractor },
    $PB_i^{new} = h(PW_{new_i}||\sigma_{new_i})$,
    $A_{U_i}^{new} = PB_i^{new} \oplus h(ID_{U_i} \oplus x)$,
    $B_{U_i}^{new} = h(ID_{U_i}||PB_i^{new}||h(ID_{U_i} \oplus x))$,
    $W_{U_i}^{new} = h(ID_{U_i}||PB_i^{new}) \oplus K_{U_i}$,
    Finally, replaces the value of $A_{U_i}, B_{U_i}, W_{U_i}$ with $A_{U_i}^{new}, B_{U_i}^{new}, W_{U_i}^{new}$. into $SC_i$
else
    User $U_i$ is unauthenticated. Abort protocol to avoid stolen smart card attack.
```

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$\mathcal{A}$ who have stolen the smart card SC_i can extract the intimate data such as $A_{U_i}, B_{U_i}, W_{U_i}, h(.), Rep(.), Gen(.), \tau_i$ from the SC_i using side channel attacks such as differential and simple power analysis. However, in our protocol the most important private information such as σ_{U_i}, x and K_{U_i} are stored in well-protected form. If $\mathcal{A}$ succeed to find out A_{U_i}, it can not find out PB_i or $h(ID_{U_i} \oplus x)$ using frequency analysis attack. The private information σ_{U_i} also can not be extracted by $\mathcal{A}$ because it is hashed after concatenated with PW_{U_i}. □

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 $\mathcal{A}$ can capture the sensor node SN_j and find out the value of the key K_{GSN_j} and session key sk. However, $\mathcal{A}$ can not use the same session key at next session because we made the session key unique using the random number r_{U_i} and r_{SN_j}. If $\mathcal{A}$ 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 $\mathcal{A}$ eavesdrops the message M_1 during user authentication and session key establishment phase, generates a random number $r_{\mathcal{A}}$ and the current time-stamp $T_{\mathcal{A}}$. However, $\mathcal{A}$ can not evaluate the value of X'_{U_i} without knowing the bio-metric information and smart card credentials of U_i in order to decrypt and modify the value of α. Likewise, it is computationally infeasible for an adversary $\mathcal{A}$ to modify the value of γ and β without knowing the key K_{GSN_j} and X_{U_i} 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 $\mathcal{A}$ intercepts the message $M_1 = \left\langle ID_{U_i}, X_{U_i}, \alpha = Enc_{X'_{U_i}}[ID_{SN_j}||T_{U_i}] \right\rangle$ 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, $\mathcal{A}$ 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_{U_i} will not be fresh and the condition $T' - T_{U_i} \leq \Delta T$ will not be satisfied. Similarly, if the adversary $\mathcal{A}$ 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.*

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, σ_i is generated using Fuzzy extractor and the function $h(.)$ is a secure one-way hash function. Therefore, for an adversary $\mathcal{A}$, 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, $\mathcal{A}$ 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 Scyther 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 $\mathcal{A}$ 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 contradiction, 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 $\mathcal{A}$ for deriving the values of user U_i's secret parameters $PW_{U_i}, \sigma_i, K_{U_i}$ 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 $\mathcal{A}$ can design an procedure $EXP_{\mathcal{A}}^{h-Enc-ECDH}$ as shown in Algorithm 1 such that probability of success of $EXP_{\mathcal{A}}^{h-Enc-ECDH}$ is $Success_{\mathcal{A}}^{h-Enc-ECDH} = |P_r[EXP_{\mathcal{A}}^{h-Enc-ECDH} = 1] - 1|$. The advantage function for $EXP_{\mathcal{A}}^{h-Enc-ECDH}$ can be represented as:

$$Adv_{\mathcal{A}}^{h-Enc-ECDH}((t_1+t_2+t_3),(q_{R_1}+q_{R_2}+q_{R_3})) = [Adv_{\mathcal{A}}^{h}(t_1) \cdot Adv_{Enc,\mathcal{A}}^{IND-CPA}(t_2) \cdot Adv_{\mathcal{A}}^{ECDH}(t_3)].$$

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: $EXP_{\mathcal{A}}^{h-Enc-ECDH}$

```
Extract $\{P, A_{U_i}, B_{U_i}, W_{U_i}, \tau_i, \mathcal{T}, h(), Gen(), Rep()\}$ from $SC_i$ using simple and differential power analysis attacks. Where
$A_{U_i} = PB_i \oplus h(ID_{U_i} \oplus x)$
$B_{U_i} = h(ID_{U_i}||PB_i||h(ID_{U_i} \oplus x))$,
$W_{U_i} = h(ID_{U_i}||PB_i) \oplus K_{U_i}$,
Call Reveal1 oracle on input $B_{U_i}$ to retrieve the information of $ID_{U_i}, PB_i, h(ID_{U_i}||x)$ as $(ID'_{U_i}||PB'_i||h(ID_{U_i}||x)') \leftarrow Reveal1(B_{U_i})$
Call Reveal1 oracle on input $PB'_i$ to retrieve the information of $PW_{U_i}, \sigma_i$ as $(PW'_{U_i}, \sigma'_i) \leftarrow Reveal1(PB'_i)$
Compute $h(ID'_{U_i}||PB'_i)$
Compute $K'_{U_i} = W_{U_i} \oplus h(ID'_{U_i}||PB'_i)$
Intercept the message $M_1 = \langle ID_{U_i}, X_{U_i}, \alpha \rangle$
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 $ID_{SN_j}, T_{U_i}$ as $(ID'_{SN_j}||T'_{U_i}) \leftarrow Reveal2$
    if $(E_{X'_{U_i}}[ID'_{SN_j}||T'_{U_i}] = \alpha)$ then
        Accept the derived $ID'_{U_i}, PW'_i, \sigma'_i and X'_{U_i}$ 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 $\mathcal{A}$ 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

$$Adv_{\mathcal{A}}^{h}(t_1) = Pr[(s, s') \leftarrow_R \mathcal{A} : s \neq s', h(s) = h(s')],$$

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

$$Adv_{\mathcal{A}}^{ECDH}(t_3) = Pr[(r_{U_i}, P) \leftarrow_R \mathcal{A} : X_{U_i} = r_{U_i} \times P]$$

Where $Adv_{\mathcal{A}}^{ECDH}(t_1) \leq \tau, Adv_{Enc,\mathcal{A}}^{IND-CPA}(t_2) \leq \tau, Adv_{\mathcal{A}}^{ECDH}(t_3) \leq \tau$.

$$\text{Therefore, } Adv_{\mathcal{A}}^{h-Enc-ECDH}((t_1+t_2+t_3),(q_{R_1}+q_{R_2}+q_{R_3})) \leq \tau.$$

which indicates that $Adv_{\mathcal{A}}^{h-Enc-ECDH}((t_1+t_2+t_3),(q_{R_1}+q_{R_2}+q_{R_3}))$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$. Now, we find that the secure hash function $h()$, encryption mechanism $Enc_k[s]$ and elliptic curve Diffie-Hellman problem $ECDH$ defined in Section 3 contradicts the oracle Reveal1, Reveal2 and Reveal3 respectively considered in Algorithm 1. This indicates that our scheme resist the adversary $\mathcal{A}$ for deriving the values of the secret parameters $PW_{U_i}, \sigma_i, K_{U_i}$, and X'_{U_i}. Hence, the theorem is proved. □

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, GWN and 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, GWN and 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 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.

```
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.

```
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’);
}
```

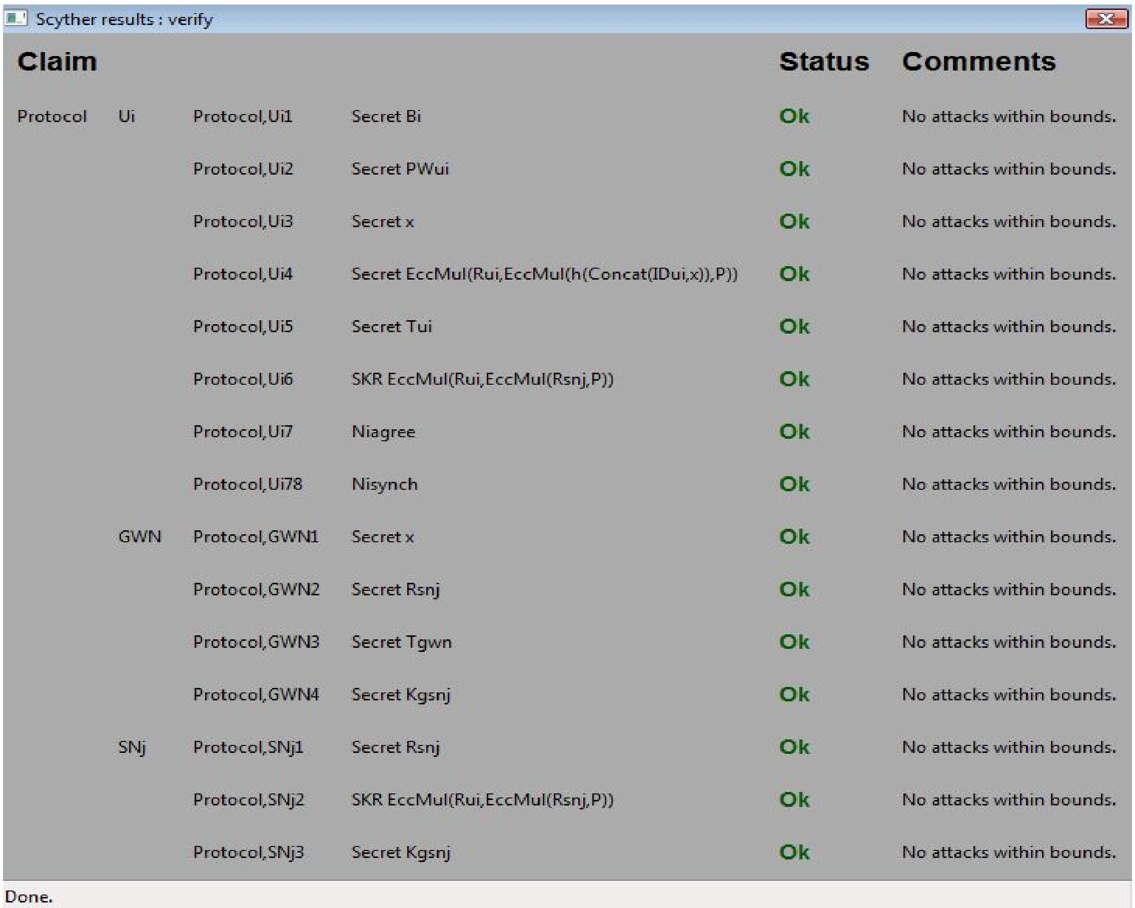
Table 13. Specification of the sensor's role in SPDL.

```
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, TUgwn);
match(Beta, Beta');
send_5(SNj, Ui, Delta, Gamma, Ysnj, Tsnj, TUgwn);
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)));
}
}
```



Scyther results : verify

Claim				Status	Comments
Protocol	Ui	Protocol,Ui1	Secret Bi	Ok	No attacks within bounds.
		Protocol,Ui2	Secret PWui	Ok	No attacks within bounds.
		Protocol,Ui3	Secret x	Ok	No attacks within bounds.
		Protocol,Ui4	Secret EccMul(Rui,EccMul(h(Concat(IDui,x)),P))	Ok	No attacks within bounds.
		Protocol,Ui5	Secret Tui	Ok	No attacks within bounds.
		Protocol,Ui6	SKR EccMul(Rui,EccMul(Rsnj,P))	Ok	No attacks within bounds.
		Protocol,Ui7	Niagree	Ok	No attacks within bounds.
		Protocol,Ui78	Nisynch	Ok	No attacks within bounds.
	GWN	Protocol,GWN1	Secret x	Ok	No attacks within bounds.
		Protocol,GWN2	Secret Rsnj	Ok	No attacks within bounds.
		Protocol,GWN3	Secret Tgwn	Ok	No attacks within bounds.
		Protocol,GWN4	Secret Kgsnj	Ok	No attacks within bounds.
	SNj	Protocol,SNj1	Secret Rsnj	Ok	No attacks within bounds.
		Protocol,SNj2	SKR EccMul(Rui,EccMul(Rsnj,P))	Ok	No attacks within bounds.
		Protocol,SNj3	Secret Kgsnj	Ok	No attacks within bounds.

Done.

Figure 2. Security verification result obtained using Scyther tool.

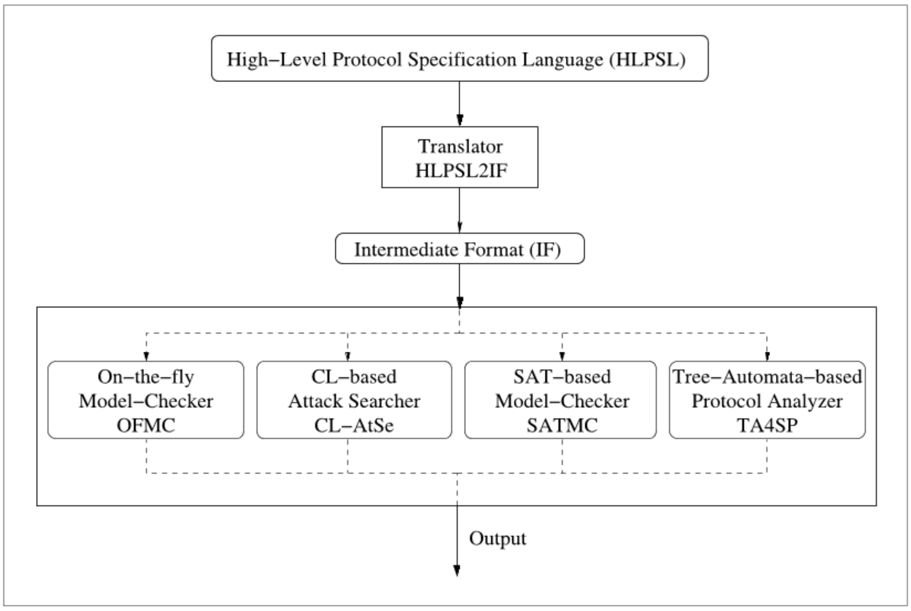

High–Level Protocol Specification Language (HLPSL)
Translator HLPSL2IF
Intermediate Format (IF)
On–the–fly Model–Checker OFMC
CL–based Attack Searcher CL–AtSe
SAT–based Model–Checker SATMC
Tree–Automata–based Protocol Analyzer TA4SP
Output

Figure 3. AVISPA Architecture.

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 Ui, Gateway GWN and Sensor node SNj) involve in the protocol.

    - Each role may have some parameter like U_i, GWN, SN_j of type agent and Kui1, Kgsnj 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 Ui" denotes that the role User is played by Ui.

    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 $\wedge$ 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:

- secret({PWi,Bi,SIGi'}, sub1, Ui): It indicates that the information {PWi,Bi,SIGi'} is secretly shared to Ui and it can be recognize 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(PWi,Bi,SIGi', sub1, Ui)).

Table 14. Specification of U_i'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, 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: = 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.

```
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 SN_j'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.

Using OFMC BACKEND

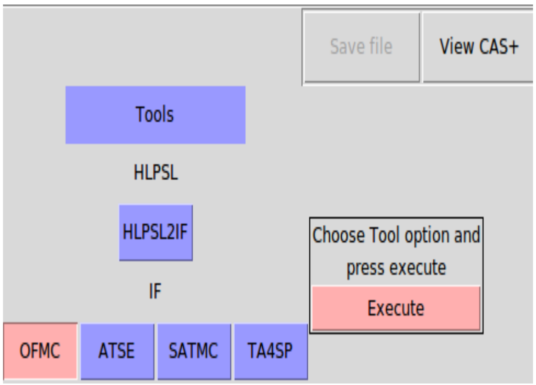
```
SUMMARY
        SAFE

DETAILS
        BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
        /home/cmb-lab-22/Desktop/Proto.if

GOAL
        as_specified
BACKEND
        OFMC

STATISTICS
        Time: 984 ms
        parseTime: 0 ms
        visitedNodes: 456 nodes
        depth: 9 piles
```



Using CL-AtSe BACKEND

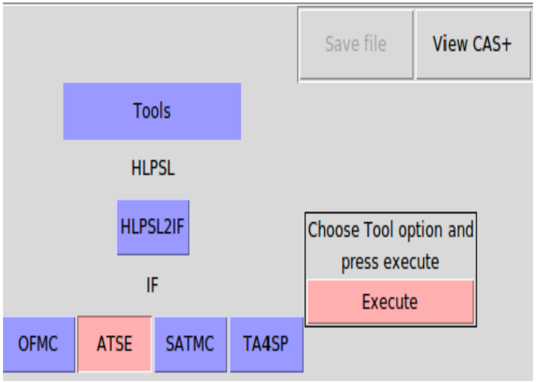
```
SUMMARY
        SAFE

DETAILS
        BOUNDED_NUMBER_OF_SESSIONS
        TYPED_MODEL

PROTOCOL
        /home/cmb-lab-22/Desktop/Proto.if

GOAL
        as_specified
BACKEND
        CL-AtSe

STATISTICS
        Analysed: 1956 states
        Reachable: 1956 states
        Translation: 0.06 s
        Computation: 0.01 s
```



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.

Notations	**Description**
P_{BAN}, Q_{BAN}	Principals like U_i, GWN, and SN_j
S	Statements like $T_{U_i}, T_{GWN}, \alpha, \beta$ etc.
K	Secret key or secret informations like $K_{GSN_j}, X'_{U_i} etc.$
$P_{BAN} \mid \equiv S$	P_{BAN} believes S, or P_{BAN} believes S is true.
$P_{BAN} \lhd S$	P_{BAN} has received a information containing S and it can read or repeat S
$P_{BAN} \mid \sim S$	P_{BAN} once said S. P_{BAN} sent a data containing S and it could be a fresh or old data.
$P_{BAN} \Rightarrow S$	P_{BAN} has jurisdiction over S. That is P_{BAN}'s beliefs about S should be trusted
$\#(S)$	The information S is fresh and it has not been sent before.
$P_{BAN} \overset{S}{\rightleftharpoons} Q_{BAN}$	S is a secret data and it is only known to P_{BAN} or Q_{BAN} and perhaps to the trusted principals
$< S >_{S1}$	$S1$ is a secret and its presence gives the identity of whoever generates $< S >_{S1}$

Rule 1 Message meaning rule: $\frac{P_{BAN}|\equiv P_{BAN} \overset{K}{\longleftrightarrow} Q_{BAN}, P_{BAN} \lhd \{S\}_k}{P_{BAN}|\equiv Q_{BAN}|\sim S}$. 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: $\frac{P_{BAN}|\equiv \#(S), P_{BAN}|\equiv Q_{BAN}|\sim S}{P_{BAN}|\equiv Q_{BAN} \equiv S}$. 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: $\frac{P_{BAN}|\equiv Q_{BAN} \Rightarrow S, P_{BAN}|\equiv Q_{BAN} \equiv S}{P_{BAN}|\equiv S}$. 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 $U_i \rightarrow GWN : ID_{U_i}, X_{U_i}, \left\langle ID_{SN_j} || T_{U_i} \right\rangle_{X'_{U_i}}$

Message 2 $GWN \rightarrow SN_j : \left\langle ID_{U_i} || sk || \left\langle ID_{SN_j} || Y_{SN_j} || T_{GWN} \right\rangle_{X'_{U_i}} || T_{GWN} \right\rangle_{K_{GSN_j}},$

Message 3 $SN_j \rightarrow U_i : \left\langle ID_{SN_j} || Y_{SN_j} || T_{GWN} \right\rangle_{X'_{U_i}}$

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.

H 1: $U_i \mid \equiv \#T_{U_i}$
H 2: $GWN \mid \equiv \#T_{GWN}$
H 3: $SN_j \mid \equiv \#T_{SN_j}$
H 4: $U_i| \equiv GWN \Rightarrow K_{U_i}$
H 5: $U_i| \equiv GWN| \equiv K_{U_i}$
H 6: $SN_j| \equiv GWN \Rightarrow K_{GSN_i},$
H 7: $SN_j| \equiv GWN| \equiv K_{GSN_j}$
H 8: $GWN| \equiv U_i \overset{X'_{U_i}}{\leftrightharpoons} GWN,$
H 9: $GWN \lhd \left\langle T_{U_i} \right\rangle_{X'_{U_i}}$
H 10: $GWN| \equiv \#(T_{U_i})$
H 11: $SN_j| \equiv GWN \overset{K_{GSN_j}}{\leftrightharpoons} SN_j,$
H 12: $SN_j \lhd \langle T_{GWN} \rangle_{K_{GSN_j}}$
H 13: $SN_j| \equiv \#(T_{GWN})$
H 14: $U_i| \equiv GWN \overset{X'_{U_i}}{\leftrightharpoons} GWN,$
H 15: $U_i \lhd \langle T_{GWN} \rangle_{X'_{U_i}}$
H 16: $U_i| \equiv \#(T_{GWN})$
H 17: $GWN| \equiv U_i \overset{X'_{U_i}}{\leftrightharpoons} GWN,$
H 18: $GWN \lhd \left\langle ID_{SN_j} \right\rangle_{X'_{U_i}}$
H 19: $SN_j| \equiv GWN \overset{K_{GSN_j}}{\leftrightharpoons} SN_j,$
H 20: $SN_j \lhd \left\langle ID_{U_i} \right\rangle_{K_{GSN_j}}$
H 21: $U_i| \equiv GWN \overset{X'_{U_i}}{\leftrightharpoons} U_i$
H 22: $U_i \lhd < Y_{SN_j} >_{X'_{U_i}}$

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

   - $\dfrac{U_i| \equiv GWN \Rightarrow K_{U_i}, U_i| \equiv GWN| \equiv K_{U_i}}{U_i| \equiv K_{U_i}}$

   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}.

   - $\dfrac{SN_j| \equiv GWN \Rightarrow K_{GSN_i}, SN_j| \equiv GWN| \equiv K_{GSN_j}}{SN_j| \equiv K_{GSN_j}}$

   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):

   - $\dfrac{GWN| \equiv U_i \overset{X'_{U_i}}{\leftrightharpoons} GWN, GWN \lhd \left\langle T_{U_i} \right\rangle_{X'_{U_i}}}{GWN| \equiv U_i| \sim T_{U_i}}$ (Based on message-meaning rule)

   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}

   - $\dfrac{GWN| \equiv \#(T_{U_i}), GWN| \equiv U_i| \sim T_{U_i}}{GWN| \equiv U_i| \equiv T_{U_i}}$ (Based on nonce verification rule)

   That is, if GWN believes that the time-stamp T_{U_i} is fresh and U_i once said T_{U_i}, then GWN believes U_i believes T_{U_i}. Therefore, we achieve Goal 3.
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):

- $$\frac{SN_j| \equiv GWN \overset{K_{GSN_j}}{\rightleftharpoons} SN_j, SN_j \triangleleft \langle T_{GWN} \rangle_{K_{GSN_j}}}{SN_j| \equiv GWN| \sim T_{GWN}}$$ (Based on message-meaning rule)

  That is, if SN_j believes the secret K_{GSN_j} is shared with GWN and sees $< T_{GWN} >_{K_{GSN_j}}$, then SN_j believes GWN once said T_{GWN}.

- $$\frac{SN_j| \equiv \#(T_{GWN}), SN_j| \equiv GWN| \sim T_{GWN}}{SN_j| \equiv GWN| \equiv T_{GWN}}$$ (Based on nonce-verification rule)

  That is, if SN_j believes that the time-stamp T_{GWN} is fresh and GWN once said T_{GWN}, then SN_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):

- $$\frac{U_i| \equiv GWN \overset{X'_{U_i}}{\rightleftharpoons} GWN, U_i \triangleleft \langle T_{GWN} \rangle_{X'_{U_i}}}{U_i| \equiv GWN| \sim T_{GWN}}$$ (Based on message-meaning rule)

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

- $$\frac{U_i| \equiv \#(T_{GWN}), U_i| \equiv GWN| \sim T_{GWN}}{U_i| \equiv GWN| \equiv T_{GWN}}$$ (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:

- $$\frac{GWN| \equiv U_i \overset{X'_{U_i}}{\rightleftharpoons} GWN, GWN \triangleleft \left\langle ID_{SN_j} \right\rangle_{X'_{U_i}}}{GWN| \equiv U_i| \sim ID_{SN_j}}$$ (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 SN_j:

- $$\frac{SN_j| \equiv GWN \overset{K_{GSN_j}}{\rightleftharpoons} SN_j, SN_j \triangleleft \langle ID_{U_i} \rangle_{K_{GSN_j}}}{SN_j| \equiv GWN| \sim ID_{U_i}}$$ (Based on message-meaning rule)

  That is, if SN_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:

- $$\frac{U_i| \equiv GWN \overset{X'_{U_i}}{\rightleftharpoons} U_i, U_i \triangleleft < Y_{SN_j} >_{X'_{U_i}}}{U_i| \equiv GWN| \sim Y_{SN_j}}$$ (Based on message-meaning rule)

That is, if U_i believes the secret X'_{U_i} is shared with GWN and sees $< Y_{SN_j} >_{X_{U_i}}$, 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 the popular features which need to be considered and the resistant against well-known attacks. Table 24 shows that our scheme overcomes the major attacks and provides more security.

Table 24. Comparison of protocols based on security features.

Security Feature	A.K.Das [17]	Choi et al. [21]	Park et al. [22]	Moon et al. [23]	Proposed Protocol
Resist stolen smart card attack	No	No	No	No	Yes
Resists Replay attack	Yes	Yes	No	Yes	Yes
Resists Man-in-the-middle attack	No	No	No	Yes	Yes
Resists user impersonation attack	No	No	No	Yes	Yes
Resists sensor impersonation attack	No	No	No	Yes	Yes
Resists insider attack	Yes	Yes	Yes	Yes	Yes
Offers mutual authentication	Yes	Yes	Yes	Yes	Yes
Offers biometric data updating	Yes	Yes	Yes	Yes	Yes
Offers secure password updating	No	No	No	Yes	Yes
Offers formal security analysis	Yes	Yes	Yes	Yes	Yes

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 25. Execution time on computer system for cryptographic operation.

Notation	Operation	Time Taken (in *Millisecond*)
T_h	One-way cryptographic hash function	0.5
T_e	Elliptic curve point multiplication	50.3
T_f	Fuzzy extractor used in biometric verification	0.5
T_E	Symmetric key encryption/decryption	8.7

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_j 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.

Function	Time (in $millisecond$)	Energy (in μ $Joule$)
Symmetric Encryption and Decryption (AES-128) [39]	$TS_E \approx 5.05$	121.2
Hashing (SHA-1) [40]	$TS_h \approx 3.63$	87.12
Elliptic curve Fixed Point Multiplication (MoTE ECC-160) [41]	$TS_e \approx 370$	8880

Table 27. Comparison of protocols based on computational cost.

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

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:

$$Energy = Voltage \times Current \times Times$$

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_{U_i}, message request req, message response R/RM, encrypted message $Enc_k[s]$, time-stamp $T_{U_i}/T_{GWN}/T_{SN_j}$, 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 $ID_{U_i}, X_{U_i}, \alpha$ requires $(160 + 160 + 128 = 448)$ bits, whereas the messages (β) and α require $(128 + 128 = 265)$ 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_{U_i}, req) requires $(160 + 32) = 192$ bits, whereas in the authentication and key agreement phase, the messages $R, Enc_{ek_i}(R, T_1, ID_{SN_j}), (ID_{U_i}, Y_j)$ 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 28. Comparison of protocols based on energy consumption on sensor node SN_j.

	A.K.Das [17]	**Choi [21]**	**Park [22]**	**Moon [23]**	**Proposed Protocol**
Energy (in μ *Joule*)	295.44	18,403.92	18,108.48	18,229.68	121.2

Table 29. Comparison of protocols based on communication overhead.

	A.K.Das [17]	**Choi [21]**	**Park [22]**	**Moon [23]**	**Proposed Protocol**
Communication Overhead (in *bits*)	832	1504	1696	1920	713
Number of Messages Communicated	5	3	3	3	3

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

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A Survey. *Comput. Netw.* **2002**, *38*, 393–422.
2. Ashton, K. That 'Internet of Things' Thing. In the real world, things matter more than ideas. *RFID J.* **2009**. Available online: http://www.rfidjournal.com/articles/view?4986 (accessed on 27 October 2017).
3. Benenson, Z.; Gartner, F.; Kesdogan, D. User authentication in sensor networks. In Proceedings of the Workshop Sensor Networks, Lecture Notes Informatics Proceedings Informatik, Ulm, Germany, 2004; pp. 385–389.
4. Watro, R.; Kong, D.; Cuti, S.F.; Gardiner, C.; Lynn, C.; Kruus, P. TinyPK: Securing Sensor Networks with Public Key Technology. In *ACM Workshop on Security of Ad Hoc and Sensor Networks*; ACM Press: Washington, DC, USA, 2004; pp. 59–64.
5. Benenson, Z.; Gedicke, N.; Raivio, O. Realizing robust user authentication in sensor networks. In Proceedings of the Workshop on Real-World Wireless Sensor Network (REALWSN'05), Stockholm, Sweden, 20–21 June 2005.
6. Wong, K.H.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the 2006 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Taichung, Taiwan, 5–7 June 2006; pp. 1–9.
7. Tseng, H.R.; Jan, R.H.; Yang, W. An improved dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'07), Washington, DC, USA, 26–30 November 2007; pp. 9869–9890.
8. Lee, T.H. Simple Dynamic User Authentication Protocols for Wireless Sensor Networks. In Proceedings of the Second International Conference on Sensor Technologies and Applications, Cap Esterel, France, 25–31 August 2008; pp. 657–660.
9. Ko, L.C. A Novel Dynamic User Authentication Scheme for Wireless Sensor Networks. In Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08), Reykjavik, Iceland, 21–24 October 2008; pp. T608–T612.
10. Vaidya, B.; Silva, J.S.; Rodrigues, J.J. Robust Dynamic User Authentication Scheme for Wireless Sensor Networks. In Proceedings of the 5th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2009), Tenerife, Spain, 26–30 October 2009; pp. 88–91.
11. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1086–1090.
12. Khan, M.K.; Alghathbar, K. Cryptanalysis and Security Improvements of "Two-factor User Authentication in Wireless Sensor Networks". *Sensors* **2010**, *10*, 2450–2459.
13. Yuan, J.; Jiang, C.; Jiang, Z. A biometric-based user authentication for wireless sensor networks. *Wuhan Univ. J. Nat. Sci.* **2010**, *15*, 272–276.
14. Yoo, S.G.; Park, K.Y.; Kim, J. A Security-performance-balanced User Authentication Scheme for Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2012**, *8*, 1–11.
15. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323.
16. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2014**, doi:10.1007/s12083-014-0285-z.

17. Das, A.K. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *Int. J. Commun. Syst.* **2015**, doi:10.1002/dac.2933.
18. Althobaiti, O.; Al-Rodhaan, M.; Al-Dhelaan, A. An efficient biometric authentication protocol for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2013**, *8*, 1–13.
19. Sharaf-Dabbagh, Y.; Saad, W. On the Authentication of Devices in the Internet of Things. In Proceedings of the 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal, 21–24 June 2016.
20. Alizadeh, M.; Abolfazli, S.; Zamani, M.; Baharun, S.; Sakurai, K. Authentication in mobile cloud computing: A survey. *J. Netw. Comput. Appl.* **2016**, *61*, 59–80.
21. Choi, Y.; Lee, Y.; Won, D. Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1–16.
22. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123.
23. Moon, J.; Lee, D.; Lee, Y.; Won, D. Improving Biometric-Based Authentication Schemes with Smart Card Revocation/Reissue for Wireless Sensor Networks. *Sensors* **2017**, *17*, 940.
24. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology-CRYPTO 99, LNCS, Santa Barbara, California, USA, 15–19 August 1999*; Springer: Berlin, Germany, 1999; Volume 1666, pp. 388–397.
25. Stinson, D.R. Some observations on the theory of cryptographic hash functions. *Des. Codes Cryptogr.* **2006**, *38*, 259–277.
26. Miller, V.S. Use of elliptic curves in cryptography. In *Advances in Cryptology-CRYPTO 85*; Lecture Notes in Computer Sciences; Springer-Verlag New York, Inc.: New York, NY, USA, 1986; pp. 417–426.
27. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology (Eurocrypt 04), Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004*; Springer: Berlin, Germany, 2004; pp. 523–540.
28. Yoon, E.J.; Kim, C. Advanced biometric-based user authentication scheme for wireless sensor networks. *Sens. Lett.* **2013**, *11*, 1836–1843.
29. Chang, I.P.; Lee, T.F.; Lin, T.H.; Liu, C.M. Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. *Sensors* **2015**, *15*, 29841–29854.
30. Cremers, C. Scyther-Semantics and Verification of Security Protocols. Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.
31. AVISPA Tool. Available online: http://www.avispa-project.org/ (accessed on 11 July 2017).
32. Bellare, M.; Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In Proceedings of the First Annual Conference Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; pp. 62–73.
33. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208.
34. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. Lond.* **1989**, *426*, 233–271.
35. Mohit, P.; Amin, R.; Karati, A.; Biswas, G.; Khan, M.K. A standard mutual authentication protocol for cloud computing based health care system. *J. Med. Syst.* **2017**, *41*, 50.
36. Chiou, S.Y.; Ying, Z.; Liu, J. Improvement of a privacy authentication scheme based on cloud for medical environment. *J. Med. Syst.* **2016**, *40*, 1–15.
37. TinyOS: An Open-Source OS for the Networked Sensor Regime. Available online: http://www.tinyos.net/ (accessed on 13 July 2017).
38. Gay, D.; Levis, P.; Von Behren, R.; Welsh, M.; Brewer, E.; Culler, D. The nesc language: A holistic approach to networked embedded systems. *ACM SIGPLAN Not.* **2003**, *38*, 1–11, doi:10.1145/781131.781133, ISSN 0362-1340,
39. Lee, J.; Kapitanova, K.; Son, S. The price of security in wireless sensor networks. *Comput. Netw.* **2010**, *54*, 2967–2978.
40. Eastlake, D., 3rd.; Jones, P. US Secure Hash Algorithm 1 (SHA 1). Available online: https://www.rfc-editor.org/rfc/rfc3174.txt (accessed on 25 October 2017).
41. Liu, Z.; Wenger, E.; Johann, G. MoTE-ECC: Energy-scalable elliptic curve cryptography for wireless sensor networks. In *Applied Cryptography and Network Security—ACNS 2014, LNCS, Proceedings of the 12th*

International Conference, ACNS 2014, Lausanne, Switzerland, 10–13 June 2014; Springer: Cham, Switzerland, 2014; Volume 8479, pp. 361–379.