sensors

MDPI

Article

A Lightweight Three-Factor Authentication Scheme for WHSN Architecture

Abdullah M. Almuhaideb and Kawther S. Alqudaihi *

Department of Computer Science, College of Computer Science and Information Technology, Imam, Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia; amalmuhaideb@iau.edu.sa

* Correspondence: 2190500127@iau.edu.sa

Received: 29 October 2020; Accepted: 24 November 2020; Published: 30 November 2020

Abstract: Wireless Healthcare Sensor Network (WHSN) is a benchmarking technology deployed to levitate the quality of lives for the patients and doctors. WHSN systems must fit IEEE 802.15.6 standard for specific application criteria, unlike some standard criteria that are difficult to meet. Therefore, many security models were suggested to enhance the security of the WHSN and promote system performance. Yu and Park proposed a three-factor authentication scheme based on the smart card, biometric, and password, and their scheme can be easily employed in three-tier WHSN architecture. Furthermore, they claimed that their scheme can withstand guessing attack and provide anonymity, although, after cryptanalysis, we found that their scheme lacks both. Accordingly, we suggested a three-factor authentication scheme with better system confusion due to multiplex parametric features, hash function, and higher key size to increase the security and achieve anonymity for the connected nodes. Moreover, the scheme included initialization, authentication, re-authentication, secure node addition, user revocation, and secure data transmission via blockchain technology. The formal analysis of the scheme was conducted by BAN logic (Burrows Abadi Nadeem) and the simulation was carried out by Tamarin prover to validate that the proposed scheme is resistant to replay, session hijacking, and guessing attacks, plus it provides anonymity, perfect forward secrecy, and authentication along with the key agreement.

Keywords: WHSN; BAN logic; tamarin prover; authentication protocol; stolen smart card; anonymity

1. Introduction

Wireless Sensor Network (WSN) is widely spread through various firms such as shrewd homes, shrewd manufactory, shrewd businesses, and smart health systems such as in WHSN [1–7]. This technology aims to reduce the patient's need to go to the hospital for checkups and allow the doctors to monitor the patients' health status from a remotely far location at any time. In the latest years, the adaptability of WHSN consists of small sizes, lower power, cheap sensors, and enables the communication among them to occur in a short-range [8]. Those sensors can be micro-controller, transceiver, memory, and battery. WHSN architecture supports sensors cooperation with each other's to build the connected sensor network architecture and inspect the user's health [9], as depicted in Figure 1.

The data collected by the sensor are saved for long time to increase its quality and to make better processing and analysis for better treatment choices [10]. Also, WHSN architecture consists of weak sensors that infringe the privacy of the patient data. Many authentication schemes were proposed to solve this issue along with many others such as anonymity, eavesdropping, DoS (Denial of Service Attack), and nodes impersonation attack [11]. After thorough analysis for the proposed schemes, we found that each has its strengths and weaknesses.

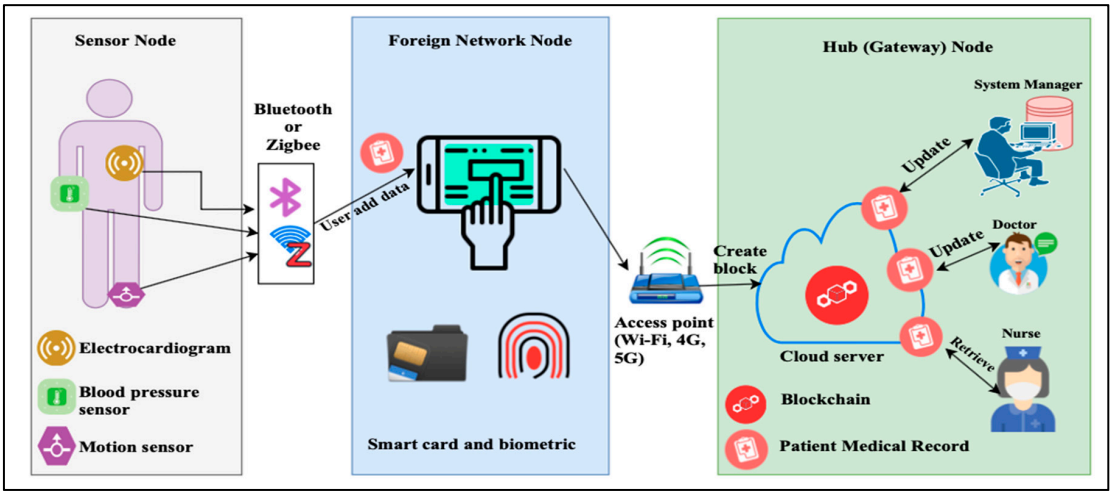

Figure 1. Network model for the Wireless Healthcare Sensor Network (WHSN) system.

Recently, Yu and Park [12] proposed three-factor authentication scheme (SLUA-WSN) for WSN network smart homes to enable the user of authenticating themselves in a secure manner. They claimed that their scheme is protected against impersonation, stolen integrated circuit card, and guessing attacks, and provides user-anonymity with un-traceability. However, we identified a lack of smart card data protection that leads to node impersonation and guessing in cases where stolen smart card attack occurred. Also, issues in anonymity and un-traceability arise, when all the previously mentioned acts are committed by the intruder i. Their scheme can be improved regarding computation and communication costs on both the foreign network side and gateway side too. Therefore, we propose a robust authentication scheme based on three-factor for WHSN higher performance and capacity efficiency besides advanced security to overcome the weaknesses in [12] scheme.

1.1. Contribution and Motivation

In continuation to the development of the WHSN authentication scheme that is proposed in our previous research [13]. We considered that the sensor node data is secure, and we proposed a secure authentication scheme between the foreign network node and the hub node. The main contributions of this article are as follows:

- Performing cryptanalysis of Yu and Park [12] scheme and show its vulnerability regarding anonymity protection, un-traceability protection, impersonation, guessing, and stolen smart card attacks.
- Proposing a lightweight three-factor authentication and re-authentication schemes consist of the biometric, smart card, and password with better key management, and less operations to increase the scheme efficiency. Also, introducing additional mechanisms such as secure node addition, secure user revocation, and data transmission via blockchain.
- Validate the scheme BAN logic, and Tamarin simulation tool to prove its authentication, key agreement, and security. The results validated the scheme security versus replay, and session hijacking attacks, plus it achieved perfect forward secrecy along with authentication and key agreement.
- Calculate the efficiency of the new scheme with computation in line with communication costs and storage. It showed an advantage of our scheme over [12] structure regarding computation cost, communication cost, and storage capacity.

The motivations of our work are described below:

- The noticeable drawbacks in most of WHSN structures, and their weaknesses towards most well-known attacks such as impersonation, session hijacking, and stolen smartcard attacks.

- Designing authentication scheme needs to achieve system scalability along with security.
- WHSN authentication schemes must provide appropriate complexity algorithm in conjunction the system capabilities along with capacity.

Accordingly, we proposed a lightweight authentication scheme to enhance the security and solve the performance deficiencies in [12]. The newly proposed scheme will provide more security with less hash functions and high parameters confusion. It is secure against offline/online shared secret guessing, brute force, replay, impersonation, eavesdropping, collision, and jamming attacks. Also, it provides numerous security features such as anonymity, integrity, un-traceability, key agreement, and mutual authentication. Likewise, the scheme is appropriate for WHSN constraint system due to its efficiency in comparison to other authentication schemes.

1.2. Organization

The remaining of this article is structured as follows. We present the state-of-art for WSN architecture in Section 2 and explain the preliminaries in Section 3. Section 4 analyzes Yu and Park's structure, and Section 5 illustrates a protected and efficient authentication schemes for WHSN architecture to improve the downfalls of Yu and Park's scheme. Section 6 assesses the security evaluation of the new scheme by executing informal and formal analysis containing BAN logic along with Tamarin simulation. Section 7 shows the outcomes of the efficiency analysis of the new scheme in comparison with the associated schemes. Finally, the conclusion is discussed in Section 8.

2. Related Works

In recent years, numerous access control and authentication systems were suggested to secure the data in WHSN technology. Some schemes are non-cryptographic based schemes that rely on the physiological signal, channel-based schemes that rely on special software or sensor, and cryptographic based schemes which are more popular [14].

Chang et al. and Park et al. [15,16] had offered an authentication structure between the user node and the gateway node and utilized a honeyword checker for the password security. Also, their scheme used random number generator from the Elliptic curve along with a hash function right before sending the authentication request. Consequently, C. Wang et al. [17] had cryptanalyzed both schemes and exposed their lack of anonymity along with their vulnerability to known session-specific temporary information (KSSTI), and privileged-user attacks. Therefore, ref. [17] suggested an improved anonymity three-factor authentication scheme utilizing an Elliptic curve cryptosystem (ECC). The structure relied on the biometric fuzzy extractor method to enhance scheme security against password guessing and identity spoofing. Unfortunately, their scheme suffered from issues in anonymity as well as backward secrecy attack when the user loses his/her smartcard, and due to some parameters lack protection.

Similarly, Challa et al. [18] recommended an authentication system with three factors in wireless body area network (WBAN) architecture based on the pubic key and Elliptic curve structure to create a secure system. They declared that their system is strong versus several types of attacks such as insider attack, password cracking, stolen smart-card, denial-of-service, known session key, masquerading, session hijacking, and replay attacks. However, their scheme lacked anonymity of the user and sensor identities. Also, the weak protection to the public key by the user phone and temporary identity made the scheme weak toward anonymity and guessing attack due to the exposure of random parameters in the open channel.

Mo and Chen [19] had analyzed the security flaws in the proposed three-factor scheme in WSN by Lu et al. [20] and found that their structure is suspectable to offline password cracking, known session-specific temporary information attacks, and lack of session key backward secrecy. Therefore, ref. [19] had offered a three-factor authentication structure utilized user biometric, smart card and key where they used hash function and Elliptic curve (ECC) to protect the passwords and security parameters. But the issue is the user anonymity might be compromised because the user identity is

only protected by random number and biometric which both might be easily guessed and spoofed by the intruder i.

To deal with the sensitivity of data issues, Garg et al. [21] proposed a system using the Elliptic curve, signatures, and blockchain for WHSN to protect the transmitted data in an insecure channel and provide anonymity. Their scheme included the identity of the trusted authority as an additional secure parameter to authenticate between the communicated nodes. Although that their scheme deployed a great combination of cryptographic and emerging technologies to protect the data, it might face DoS attack and communication delay between the nodes, because of the heavy computation along with high storage cost. Ali et al. [22] had cryptanalyzed Liu and Chung [23] scheme and found out that it is unguarded to lost smart-card, offline key cracking, insider, and masquerading attacks. Moreover, ref. [22] had analyzed [18]'s approach, and found that it has correctness issues, broadcasting problems, lack of authentication between the trusted authority and sensor nodes, replay attack, Denial of Service (DoS), and forgery attacks.

Therefore, Ali et al. [22] had suggested a secure and lightweight three-factor authentication process for WHSN which employed both ECC, and bilinear pairing to resolve the issues in [18,23] schemes. Although their scheme is guarded against impersonation, privileged-insider, offline password cracking, stolen smart-card, and replay attacks, but it still has high computation cost and delays in communication due to extensive cryptographic operations.

One of the significant issues that faces the IoT authentication structures is jamming attack, when the intruder i sends jamming signal during the update of authentication values, and parameters [24]. In this context, two authentication schemes proposed by Shen et al. and Tewari et al. [25,26] that employed simple operations such as hash, XOR, and random number generators. Their schemes focused on time duration of the session, mutual random number generation, and keeping the latest identities of the communicating entities to increase the protection against jamming attacks.

Recently, Yu and Park [12] proposed SLUA-WSN which is a lightweight three-factor authentication scheme with secure user authentication system. Their scheme has the best in efficiency of all the previous schemes in the state-of-art, and the best robustness against sensor node capture, replay attack, insider attack, and impersonation attack, also it guarantees un-traceability and mutual authentication. Thus, SLUA-WSN is appropriate for applied WHSN environments because it is the strongest and efficient than related schemes. Their scheme suffers from stolen smart cards and shared secret key guessing because of the number of stored parameters in the smartcard. Secondly, there is no mechanism to check the validity of the generated random number at the first communication session between FN and GW. There must be a validation method to check whether the user generated the accepted random number that is generated before or not in case of mobile lost/smart card lost attacks.

3. Preliminary

This section deliberates the preliminaries used in both of our proposed protocols.

3.1. Fuzzy Extractor

In this section, we discuss fuzzy extractor function which is a cryptographic authentication mechanism that employs biometric and consists of two operations:

- Gen: After the biometric input Bio is imprinted by users, Gen produces a consistent random string σ {0, 1}, a random auxiliary string σ {0, 1}, and a probabilistic function.
- Rep: It reproduces σ with value σ when a disruptive biometric *BIO* new is inscribed, where σ is a public replication value connected with Bio.

3.2. Intruder Model

To analyze our model security, we discussed a very well-known Dolev–Yao (DY) threat model [27]. In the DY design, the intruder i capabilities are as presented below.

- Referring to the DY model [27] an intruder i can inject, delete, intercept, and eavesdrop the data exchanged over wireless networks.
- Using the DY model [27], the data transmitted over wireless networks can be implanted, modified, recorded, and snooped by an intruder i.
- An intruder i can capture legal users' smart cards and can use power-analysis to retrieve confidential keys stored in memory [28].
- An intruder i can undertake numerous attacks after extracting the smart card's secret credentials, such masquerade, offline key guessing, trusted insider, and forward secrecy attacks [29,30].

4. Review on Yu and Park Scheme

In this section, we reviewed [12] to discover its weaknesses and points of enhancements, also we conducted cryptanalysis of the scheme, and we found that it lacks anonymity and the protection against secret shared key guessing. We discuss the scheme symbols in Table 1 as well as the registration and authentication phases.

Table 1. Symbols used in Yu and Park protocol.

Notation	Description
U_i	User
GWN	Gateway node
S_j	Sensor node
ID_i	Ui's identity
PW_i	Ui's password
SID_j	Sj's identity
K_{GWN}	Master key of GWN
X_{pub}	Public key of GWN
x_j	Secret key of Sj
E, F_p	Elliptic curve E defined on the finite field Fp with order p
G	A group for an elliptic curve
P	The generator of G
E_k, D_k	Symmetric key encryption/decryption
SK	Session key
T_i	Timestamp
BIO	Biometric of Ui
$h(.)$	Hash function
$\oplus$	XOR operation
$\|\|$	Concatenation operation

4.1. Registration Phase of Yu and Park Scheme

In the registration phase of [12], the user and GWN communicate with one another to produce username, password, biometric, and smartcard values:

1. The user U_i, inputs his/her password, username, and biometric, extract the biometric features using reproduction function and send those value over a secure channel to GWN.
2. GWN produces random value r_g, calculates identification values MID_i, X_i, Q_i, and W_i to store $\{Qi, Wi, MIDi\}$ in the SC, and save r_g in secure database. The number of saved parameters in the smartcard causes a weakness, that the attacker can seize to exploit the system parameters, by performing a smartcard impersonation attack along with database hijacking to retrieve the random value and user biometric.

4.2. Authentication Phase of Yu and Park Scheme

In the authentication phase of [12], the user and GWN authenticate each other along with the sensor to agree on the next session key as follows:

1. Client U_i inputs his/her username, password, and biometric in the smartphone, and checks the user identity before generating a current random value R_u along with a timestamp.
2. The mobile masks the following parameters such as original user identity ID_i, the hidden user identity SID_i, user masked identity in the smartcard MID_i, and the masked X_i.
3. The GWN node receives the parameters, checks the timestamp to avoid replay attack, and retrieves the random values from the masked X_i without checking if the random value had been used before. This step might rise a vulnerability in the scheme in case of a user node impersonation attack.
4. The GWN shares the hidden values with the sensor S_j for further authentication to the user node, and to support the next session key generation.
5. S_j authenticates both GWN and U_i, and generates new random nonce to produce the next pre-shared key.
6. Both GWN and U_i receive the new parameters to recover the generated values and save the new session key.

4.3. Cryptanalysis of Yu and Park Scheme

From the above, we deliberated the flaws for [12] structure in both registration and authentication phases which are: The number of authentication parameters in the smartcard along with the absence of random number checking in the GWN. Those two weaknesses allow the intruder to impersonate the user in the lost smartcard attack and weaken the anonymity.

Stolen Smart Card Attack

In this malicious act, an intruder i may attempt to masquerade the legal client and discover all the security parameters of the user by stealing the user smartcard and performing a guessing attack. According to the intruder model, we assumed that intruder i had extracted all the secret credentials that are adequately enough to impersonate the user from the smart card as follows $\{Q_i, W_i, MID_i\}$ and had obtained the random number $\{r_g, K_{GWN}\}$, and spoofed the *BIO* of the user by performing database hijacking attack on the GWN node and smartphone node. Then, the attacker can perform the following:

1. Intruder i computes the $ID_i = h(MID_i \| h(K_{GWN} \| r_g))$, and discovers the identity of the real user that brings lack of anonymity issue.
2. Calculate $X_i = h(MID_i \| r_g \| K_{GWN})$, $MPW_i = h(MID_i \| X_i) \oplus Q_i$.
3. Start authentication operation use the spoofed *BIO* and evade the threshold value of the biometric checking.
4. U_i computes the security parameters X_i, W_i^*, MPW_i, then checks the $W_i^* =? W_i$ provided by the attacker. If they are the same then the attacker can generate any random value instead of R_u which is R_{fake} and current time T_1 to deduce the following: $CID_{fake} = (ID_i \| SID_j) \oplus h(MID_i, R_{fake})$, $M_{UG} = h(ID_i, R_{fake}, X_i, T_1)$, then send the following parameters to GWN $\{M_1, MID_i, CID_{fake}, M_{UG}, T_1\}$.
5. GWN first checks the timestamp, if it is valid, then it will check other security parameters without validating the random number R_{fake} whether similar to the random R_u in the database or whether it is used before or not. Moreover, if the GWN does not have a mechanism to check the validity of the random number generated by the U_i node then the Intruder i can generate fake random values and bypass the system security.
6. Then GWN chooses any random number R_g the same random number known by the intruder i, and the current time T_2 to calculate M_2, M_{GS} by performing the following equations $M_2 = (R_{fake} \| R_g) \oplus h(SID_i \| X_j \| T_2)$ and $M_{GS} = h(MID_i \| SID_j \| R_{fake} \| R_g \| T_2)$. Then, GWN sends the parameters $\{M_2, MID_i, M_{GS}, T_2\}$ to S_j.
7. Intruder i can intercept the parameters $\{M_2, MID_i, M_{GS}, T_2\}$ between the channels to get the sensor node identity by applying this formula $SID_j = h(MID_i \| M_{GS} \| R_{fake} \| R_g \| T_2)$.

After repeating these steps from 1 to 7 by the intruder i, the intruder can discover all the security parameters alongside predict, intercept, and impersonate all the next upcoming parameters between the channels. This cryptanalysis showed that a smart card attack with little effort from the attacker can jeopardize the nodes anonymity and evade the system security.

In the next section, we propose the enhanced protocol to improve the security of [12] scheme that covers different phases in the authentication process.

5. Proposed Protocol

We explain our suggested authentication scheme assuming that our sensor node is trusted node and we want to secure the communication between the hub node and foreign network node. As the foreign network node works as a data collector for the sensor node and ensures the security of the transmitted parameters to the hub node. The scheme ensures strong authentication between FN-HN due to its three-factor authentication nature, and complex parametric system. In the below explanation, we showed the system five phases such as FN pre-deployment and registration, FN-HN authentication, re-authentication, safe node addition, user revocation, and secure data transmission via blockchain. Refer to Table 2 below for the notations.

Table 2. Notations of the scheme.

Symbol	Description
SM	System controller (manager)
SN	Second-level node (sensor)
$FN/(user)$	Foreign network node/user
$HN\ (gw)$	Gateway node (hub)
SC	Smart card
ID_u, PW_U	User identity and password picked by the user
SID_U, SPW_U, SID_U^{**}	User shadow identity and shadow password/updated shadow identity
ID_{SN}, ID_{SN}^{+}, ID_{SN}^{++}	Second-level node identity generated by SM/hidden ID_{SN} updated/changed ID_{SN} constantly.
ID_{FN}, ID_{FN}^{+}	Foreign network node identity (user identity) created by SM/masked ID_{FN}
ID_{FN}^{++}	Updated Foreign network in every period or in every updated user identity
ID_{HN}, ID_{HN}^{+}, ID_{HN}^{++}	Gateway identity created by the system controller/hidden HN identity/updated HN identity
t_i, t_j, t_S	Recent time of the Foreign network and GW nodes
K_{MS}	Master secret key created by the controller pre-shared between FN and HN
SK	Session key computed by SM
SK^{+}, SK_n^{+}	Renewed session key/updated symmetric key
$rand$, $rand^{*}$, $rand^{+}$	Random nonce/renewed random nonce
BIO	Biometric of the user
DB_{HN}	Database of the hub node
$E, F_{i1}, F_{(i+1)}, A, B, J, K$	Verification parameters
$\oplus$	XOR operator
$h(.)$	Cryptographic hash
$\|\|$	Concatenation operation

5.1. Initialization Stage

In this stage, the parameter generation and registration of this protocol engaged the SM to choose secret identities, parameters and keys and allow all the entities to share securely the generated arguments over an offline and secure channel to *SN*, *FN*, and *HN*:

1. The SM stores the ID_{SN}, ID_{FN}, and ID_{HN} in the SM memory.
2. SM chooses secret key K_{MS}, and $l*$ as a secret parameter to be added to the node's keys for the confusion.
3. SM computes the secret key between the parameters

$$SK = h(ID_{SN} \ || \ h(l* \ || \ K_{MS}) \ || \ ID_{FN} \ || \ ID_{HN}) \quad (1)$$

4. SM calculates a new shadow identity for all the communicated nodes to ensure their anonymity during the communication and transmits those identities over a secure channel.

$$ID_{SN}^{+} = ID_{SN} \parallel l* \parallel ID_{HN} \tag{2}$$

$$ID_{FN}^{+} = ID_{FN} \parallel l* \parallel ID_{HN} \tag{3}$$

$$ID_{HN}^{+} = ID_{SN} \parallel l* \parallel ID_{HN} \parallel ID_{FN} \tag{4}$$

5. *HN* communicates with *SN* to generate secret parameters in a secure channel to authenticate each other during the communication.

$$E = h\Big(ID_{SN} \parallel h\Big(l* \parallel ID_{HN}^{+}\Big)\Big) \tag{5}$$

6. *SN* saves the newly generated secret parameter in a secure location.
7. *HN* communicates with *FN* to generate secret parameters in a secure channel to authenticate each other during the communication.

$$F_{i1} = h\Big(ID_{FN} \parallel h\Big(l* \parallel ID_{HN}^{+}\Big)\Big) \tag{6}$$

8. *FN* keeps the new produced secret parameter in a secure location to enable the user from registering securely in the WHSN authentication system.

5.2. Registration Phase

In this phase, the client uses his/her smartphone to enter the password, identity, and biometric to allow the *HN* from generating an authentication smart card securely:

1. The user inputs his/her ID_u and PW_U and imprints the biometric *BIO* to compute the user identity and password for SC registration:

$$Gen(BIO) = \langle \delta_i | \tau_i \rangle \tag{7}$$

$$SPW = h\Big(BIO \parallel ID_{FN}^{+} \parallel \delta_i\Big), \tag{8}$$

where δ_i: is the user biometric feature and τ_i: Is the threshold. Then, *FN* sends these values $\{ID_U, SPW\}$ to *HN* in a secure channel.

2. *HN* receives the parameters, generates random value *rand*, and computes the following:

$$SID_U = h\Big(ID_U \parallel ID_{FN}^{+} \parallel ID_{HN}^{+} \parallel rand\Big) \tag{9}$$

$$F_{(i+1)} = h\Big(SID_U \parallel ID_{FN}^{+} \parallel rand\Big) \tag{10}$$

$$G = h(SID_U \parallel SPW \parallel l* \parallel rand) \tag{11}$$

3. *HN* hides the value of *rand* with this equation:

$$H = h\Big(rand \parallel ID_{FN}^{+} \parallel ID_{HN}^{+} \parallel ID_{SN}^{+}\Big) \tag{12}$$

4. Store the parameters $\Big\{F_{(i+1)}, G, H\Big\}$ in the SC and send it to the user.
5. *FN* receives the parameters and retrieves the random number from Formula (12) to store it securely in the memory and deletes H from the smart card to avoid stolen smart card attacks. The set of new parameters will be $\Big\{F_{(i+1)}, G\}$, as depicted in Figure 2.

FN (USER)		HN(Gateway)	
Enter ID_u and PW_u			
Input biometric ***BIO***			
Calculate			
$Gen(BIO) = \langle \delta_i \vert \tau_i \rangle$	(7)		
$SPW = h(BIO \parallel ID_{FN}^{+} \parallel \delta_i)$	(8)		
$\{ID_U, SPW\}$ →			
		Produce random value rand	
		Calculate	
		$SID_U = h(ID_U \parallel ID_{FN}^{+} \parallel ID_{NN}^{+} \parallel rand)$	(9)
		$F_{(i+1)} = h(SID_U \parallel ID_{FN}^{+} \parallel rand$	(10)
		$G = h(SID_U \parallel SPW \parallel l* \parallel rand)$	(11)
		Hide rand in a masking formula	
		$H = h(rand \parallel ID_{FN}^{+} \parallel ID_{NN}^{+} \parallel ID_{SN}^{+})$	(12)
		Store $\{F_{(i+1)}, G, H\}$ in SC	
		$\{SC\}$ ←	
Delete H from SC			
Store $\{F_{(i+1)}, G\}$ in SC.			

Figure 2. The proposed scheme registration phase.

5.3. P-I: Authentication Phase

In this section, we assumed that the *SN* is a trusted node and *FN* authenticates itself and *SN* to the *HN*. Furthermore, it encompasses four phases of communications including *FN*, *HN*, and *SN* depicted in the Figure 3 below and denoted as follows:

FN→	←HN
User inserts SC	
Choose ID_u, PW_U,	
Imprint the biometric BIO	
$Gen(BIO) = \langle \delta_i \vert \tau_i \rangle$ from Formula (7)	
$SPW = h(BIO \parallel ID_{FN}^{+} \parallel \delta_i)$ from Formula (8).	
$G^* = h(SID_U \parallel SPW \parallel l* \parallel rand)$ from Formula (11).	
Checks $G^* =? G$	
Generate new timestamp t_{i1} random value $rand^*$	
$A = rand^* \oplus F_{i1} \oplus ID_{SN}^{+}$ Formula (13)	
$B = rand^* \oplus SK$ Formula (14)	
$(F_{(i+1)}, A, B, t_{i1})$ →	
	Check the time validity $\Delta t = t_{i2} - t_{j1}$
	Compute $SID_U^* = h(F_{(i+1)} \parallel ID_{FN}^{+} \parallel rand)$ from Formula (10)
	Verify $SID_u^* =? SID_U$
	Calculate $rand^* = A \oplus F_{i1} \oplus ID_{SN}^{+}$ from Formula (13)
	Check if the $rand^*$ had been used before, if not, continue.
	Extract $ID_{SN}^{+} = A \oplus F_{i1} \oplus rand^*$, and check if the $ID_{SN}^{+} ? = ID_{SN}^{+}$
	Calculate $F_{i1} = h(ID_{FN} \parallel h(l* \parallel ID_{NN}^{+}))$ from Formula (6)
	$SK = rand^* \oplus B$
	Generate new random nonce $rand^+$, new time stamp t_{j2}
	Deduce $ID_{SN}^{++} = ID_{SN}^{+} \parallel l* \parallel ID_{NN}^{+}$ from Formula (2)
	$ID_{FN}^{++} = ID_{FN}^{+} \parallel l* \parallel ID_{NN}^{+}$ from Formula (3)
	$ID_{NN}^{++} = ID_{SN}^{++} \parallel l* \parallel ID_{NN}^{+} \parallel ID_{FN}^{++}$ from Formula (4).
	Calculate $SK^+ = h(ID_{SN}^{++} \parallel h(l* \parallel K_{MS}) \parallel ID_{FN}^{++} \parallel ID_{NN}^{++})$ Formula (1)
	$SID_u^{+*} = h(ID_U \parallel ID_{FN}^{++} \parallel ID_{NN}^{++} \parallel rand^+)$ Formula (9)
	Update $F_{(i+1)}^{new} = h(SID_u^{+*} \parallel ID_{FN}^{++} \parallel rand^+)$from Formula (10)
	$G_{new} = h(SID_u^{+*} \parallel SPW \parallel l* \parallel rand^+)$ from Formula (11)
	Create $J = rand \oplus SID_u^{+*}$, $K = ID_{NN}^{++} \oplus rand^+$ Formula (15)
	(J, K, t_{j2}) ←
Verify the time $\Delta t = t_{i3} - t_{j2}$	
Upon receiving the parameters FN calculates:	
Deduce $SID_u^{+*} = rand \oplus J$ from Formula (15)	
$ID_{SN}^{++} = ID_{SN}^{+} \parallel l* \parallel ID_{NN}^{+}$ from Formula (2)	
$ID_{FN}^{++} = ID_{FN}^{+} \parallel l* \parallel ID_{NN}^{+}$ from Formula (3)	
$ID_{NN}^{++} = ID_{SN}^{++} \parallel l* \parallel ID_{NN}^{+} \parallel ID_{FN}^{++}$ from Formula (4).	
$rand^+ = ID_{NN}^{++} \oplus K$.	
Replace $F_{i2} = h(SID_u^{+*} \parallel ID_{FN}^{++} \parallel rand^+)$from Formula (10)	
$G = h(SID_u^{+*} \parallel SPW \parallel l* \parallel rand^+)$ from Formula (11),	
Add the new parameters to the SC $\{F_{(i+1)}^{new}, G_{new}\}$.	

Figure 3. P-I: Three-factor authentication and key agreement protocol.

Step 1: $FN \rightarrow HN\left(M1 = \left\{F_{(i+1)}, A, B, t_{i1}\right\}\right)$

- The user inserts the smart card, enters his/her ID_u, PW_U, imprints the biometric BIO, and calculates $Gen(BIO) = \langle \delta_i \vert \tau_i \rangle$ from Formula (7) and $SPW = h\left(BIO \parallel ID_{FN}^{+} \parallel \delta_i\right)$ from Formula (8). Also, computes $G^* = h(SID_U \parallel SPW \parallel l* \parallel rand)$ from Formula (11).

- *FN* checks whether $G^* = ?G$, then continue, else abort the session. *FN* generates a new timestamp t_{i1} random value $rand^*$ and calculates the following:

$$A = rand^* \oplus F_{i1} \oplus ID_{SN}^+ \tag{13}$$

$$B = Rand^* \oplus SK \tag{14}$$

The previous step is very important to prevent jamming attack in any communication session. It allows both *FN* and *HN* to be part of generating random numbers that supports the session key formation.

FN sends the parameters to the *HN* in the open channel $\{F_{(i+1)}, A, B, t_{i1}\}$.

Step 2: $HN \rightarrow FN\left(M2 = \{J, K, t_{j2}\}\right)$

HN performs the following:

- Verify the time $\Delta t = t_{j2} - t_{i1}$ to prevent a replay attack.
- If $\Delta t = t_{j2} - t_{i1} > 0$ then continue, else terminate the process.
- Calculate $SID_U^* = h\left(F_{(i+1)} \parallel ID_{FN}^+ \parallel rand\right)$ from Formula (10).
- Check if $SID_u^* = ?\ SID_U$ to validate the user identity and remain, else terminate the process.
- *HN* should keep track of each used random number in the scheme to avoid replay attack or impersonation attacks.
- Calculate $rand^* = A \oplus F_{i1} \oplus ID_{SN}^+$, from Formula (13) and check if the $rand^*$ has been used before, if it is not continue to extract $ID_{SN}^+ = A \oplus F_{i1} \oplus rand^*$, and check if the $ID_{SN}^+ = ?\ ID_{SN}^+$ to authenticate the sensor node.
- Calculate $F_{i1} = h\left(ID_{FN} \parallel h\left(l* \parallel ID_{HN}^+\right)\right)$ from Formula (6) to authenticate the *FN*.

$SK = rand^* \oplus B$ from Formula (14) to validate the shared secret key.

After authenticating the *SN* and *FN*, *HN* generates new random nonce $rand^+$, new timestamp t_{j2} and calculates the following

- Deduce $ID_{SN}^{++} = ID_{SN}^+ \parallel l* \parallel ID_{HN}^+$ from Formula (2), $ID_{FN}^{++} = ID_{FN}^+ \parallel l* \parallel ID_{HN}^+$ from Formula (3), $ID_{HN}^{++} = ID_{SN}^{++} \parallel l* \parallel ID_{HN}^+ \parallel ID_{FN}^{++}$ from Formula (4).
- Calculate $SK^+ = h\left(ID_{SN}^{++} \parallel h(l* \parallel K_{MS}) \parallel ID_{FN}^{++} \parallel ID_{HN}^{++}\right)$ Formula (1)
- Compute $SID_u^{**} = h\left(ID_U \parallel ID_{FN}^{++} \parallel ID_{HN}^{++} \parallel rand^+\right)$ Formula (9) and update $F_{(i+1)}^{new} = h\left(SID_u^{**} \parallel ID_{FN}^{++} \parallel rand^+\right)$ from Formula (10), $G_{new} = h(SID_u^{**} \parallel SPW \parallel l* \parallel rand^+)$ from Formula (11). The above formulas ensure our scheme robustness towards jamming attack, due to the usage of old identities and keys in the generation of the new system parameters.
- Create security parameters to hide the new values:

$$J = rand \oplus SID_u^{**},\ K = ID_{HN}^{++} \oplus rand^+. \tag{15}$$

HN sends the parameters to the *FN* in the open channel $\{J, K, t_{j2}\}$.

Step 3: $FN \rightarrow SN$ ($M_3 = \{J, K, t_{i3}\}$)

FN performs the following:

- Verify the time $\Delta t = t_{i3} - t_{j2}$ to prevent a replay attack.
- If $\Delta t = t_{i3} - t_{j2} > 0$ then proceed, else halt the connection.

Upon receiving the parameters *FN* calculates the following:

- Deduce $SID_u^{**} = rand \oplus J$ from Formula (15), $ID_{SN}^{++} = ID_{SN}^+ \parallel l* \parallel ID_{HN}^+$ from Formula (2), $ID_{FN}^{++} = ID_{FN}^+ \parallel l* \parallel ID_{HN}^+$ from Formula (3), $ID_{HN}^{++} = ID_{SN}^{++} \parallel l* \parallel ID_{HN}^+ \parallel ID_{FN}^{++}$ from Formula (4) $rand^+ = ID_{HN}^{++} \oplus K$.

- Replace $F^{new}_{(i+1)} = h\Big(SID^{**}_u \parallel ID^{++}_{FN} \parallel rand^{+}\Big)$ from Formula (10), $G_{new} = h(SID^{**}_u \parallel SPW \parallel l* \parallel rand^{+})$ from Formula (11), and add the new parameters to the SC $\{F^{new}_{(i+1)},\ G_{new}\Big\}$.

  FN sends the parameters to the *SN* in the open channel $\{K,\ t_{i3}\}$.
  Step 4: $\rightarrow$ *SN* ($M_4 = \{K,\ t_{i3}\}$)
  Upon receiving the parameters *SN* calculates the following:

- Verify the time $\Delta t = t_{S4} - t_{i3}$ to prevent a replay attack.
- If $\Delta t = t_{s4} - t_{i3} > 0$ then proceed, else terminate the session.
- Deduce $ID^{++}_{SN} = ID^{+}_{SN} \parallel l* \parallel ID^{+}_{HN}$ from Formula (2), $ID^{++}_{FN} = ID^{+}_{FN} \parallel l* \parallel ID^{+}_{HN}$ from Formula (3), $ID^{++}_{HN} = ID^{++}_{SN} \parallel l* \parallel ID^{+}_{HN} \parallel ID^{++}_{FN}$ from Formula (4).
- $rand^{+} = ID^{++}_{HN} \oplus K$, then replace $SK^{+} = h\Big(ID^{++}_{SN} \parallel h(l* \parallel K_{MS}) \parallel ID^{++}_{FN} \parallel ID^{++}_{HN}\Big)$ Formula (1). Save the new parameters in the *SN* memory and establish the new session key.

5.4. P-II: Re-Authentication Phase

After an effective authentication session, the user is qualified to approach the system services. The authentic client might want to reach to some facilities throughout the day before night-time. Furthermore, it is timewasting, and un-efficient to compute all the values of the updated authentication session for the verified client. Hence, it necessitates the concept of re-authentication to improve the scheme efficiency, as shown in Figure 4. The stages of the re-authentication are as follows:

1. The user enters to his/her account to approach some data from the smartphone. **Step 1:** *FN* $\rightarrow$ *HN* ($M_1 = (F_{(i_{+1})},\ A,\ t_{i1})$) The *FN* calls the last SID_U, PW_U before the night-time to confirm the *FN* to the GW.

   - Imprint the biometric *BIO*.
   - Calculate $Gen(BIO) = \langle \delta_i | \tau_i \rangle$ from Formula (7) and $SPW = h\Big(BIO \parallel ID^{+}_{FN} \parallel \delta_i\Big)$ in Formula (8).
   - Compute $G^{*} = h(SID_U \parallel SPW \parallel l* \parallel rand)$ from Formula (11).
   - Check $G^{*} =?G$ if no abort, and if yes continue.
   - Generate new time stamp t_{i1} and calculate

$$A_i = rand^{*} \oplus SK \oplus ID^{+}_{SN} \oplus ID^{+}_{FN}. \tag{16}$$

2. Send the following parameters $\{F_{(i+1)}, A_i,\ t_{i1}\}$ to the *HN* for authentication. **Step 2:** *HN* $\rightarrow$ *FN* ($M_3 = \{L,\ O, t_{j2}\}$)

   - Verify the time $\Delta t = t_{j2} - t_{i1}$, to prevent replay attack.
   - If $\Delta t = t_{j2} - t_{i1} > 0$ then remain, else cancel the session.
   - *HN* checks if $F_{(i+1)}$ was generated throughout the past 12 h.
   - If yes, then *HN* gets the newest random nonce and computes the fresh key $A^{*}_i = rand^{*} \oplus SK \oplus ID^{+}_{SN} \oplus ID^{+}_{FN}$ from Formula (16).
   - Check if $A^{*}_i =? A_i$ if equal, then proceed.
   - Produce a new random nonce for the new connection *rand* (i), where $i = \{i+1,\ i+2,\ i+3,\ i\ +\ n \,|n :$ *the number of new sessions*$\}$, then compute the following to replace the SC old parameters with the new ones: $F^{new}_{(i+1)} = h\Big(SID_U \parallel ID^{++}_{FN} \parallel rand_{(i+1)} \parallel rand^{*}\Big)$ from Formula (10), $G_{new} = h\Big(SID_U \parallel SPW \parallel l* \parallel rand_{(i+1)}\Big)$ from Formula (11).
   - Produce recent time t_{j2} and confirm the *HN* response.

$$L = ID^{+}_{HN} \oplus ID^{++}_{SN} \oplus G_{new} \tag{17}$$

$$O = F_{(i+1)}^{new} \oplus rand^* \tag{18}$$

- Change the tuple values with the new ones $\left\{F_{(i+1)}^{new}, G_{new}\right\}$, and save them to the SC.

3. Send the following parameters $\{L, O, t_{j2}\}$ to *FN*. **Step 3:** → *FN* ($M_4 = \{L, O, t_{j2}\}$)

- Check the time validity $\Delta t = t_{i3} - t_{j2}$. If $\Delta t = t_{i3} - t_{j2} > 0$ then proceed, else halt the connection.
- Compute the following $G_{new} = ID_{HN}^{+} \oplus ID_{SN}^{++} \oplus L$ from Formula (17), $F_{(i+1)}^{new} = O \oplus rand^*$ from Formula (18).
- Change the parameters with the fresh ones $\left\{F_{(i+1)}^{new}, G_{new}\right\}$, and save them to the SC.

FN→	←HN
Select the last SID_U, PW_U before the midnight Imprint the biometric BIO Calculate $Gen(BIO) = (\delta_i \| \tau_i)$ from Formula (7) $SPW = h(BIO \| ID_{HN}^{+} \| \delta_i)$ from Formula (8). Compute $G^* = h(SID_U \| SPW \| l* \| rand)$ from Formula (11). Checks $G^* =? G$ Generate new time stamp t_{i1} and calculate: $A_i = rand^* \oplus SK \oplus ID_{SN}^{+} \oplus ID_{FN}^{+}$ Formula (16) $(F_{(i+1)}, A_i, t_{i1})$ →	
	Validate the time $\Delta t = t_{j2} - t_{i1}$ Verify the F_{i2} was used throughout the past 12 hours. If it is, then HN fetches the latest random number and computes $A_i^* = rand^* \oplus SK \oplus ID_{SN}^{+} \oplus ID_{FN}^{+}$ from Formula (16) Verify if $A_i^* =? A_i$ Generate a new random number for the new session rand(i), where $i = \{i+1, i+2, i+3, i+n \mid n: the\ number\ of\ new\ sessions\}$ $F_{(i+1)}^{new} = h(SID_U \| ID_{FN}^{++} \| rand_{(i+1)} \| rand^*)$ from Formula (10), $G_{new} = h(SID_U \| SPW \| l* \| rand_{(i+1)})$ from Formula (11) Produce recent time t_{j2} and validate the HN response $L = ID_{HN}^{+} \oplus ID_{SN}^{+} \oplus G_{new}$ Formula (17) $O = F_{(i+1)}^{new} \oplus rand^*$ Formula (18) Replace the SC parameters with the new ones.
	(L, O, t_{j2}) ←
Check the time validity $\Delta t = t_{i3} - t_{j2}$ Compute $G_{new} = ID_{HN}^{+} \oplus ID_{SN}^{++} \oplus L$ from Formula (17) $F_{(i+1)}^{new} = O \oplus rand^*$ from Formula (18) Replace the SC parameters with the new ones SC $\{F_{(i+1)}^{new}, G_{new}\}$.	

Figure 4. P-II: Three-factor re-authentication and key agreement protocol.

A protected connection can be initiated between *FN* and GW.

5.5. Secure Node Addition

SM adds new nodes to the system and performs the following:

Step 1: User sends a request to add new SN_i^{new}.

The client needs to normally log into the session with his/her credentials, inserts SC, enters ID_u and PW_U, imprints the biometric *BIO*, and generates time stamp t_{i1}.

- After a successful log in the user generates secret value for request validation.

$$M_1 = h('add\ node', E, SID_U) \tag{19}$$

- Send this message to SM for node addition.

Step 2: SM receives the request of the user to create new SN_i^{new}.

- SM checks the time validity $\Delta t = t_{j2} - t_{i1}$.
- If $\Delta t = t_{j2} - t_{i1} > 0$ then proceed, else halt the session.
- SM opens the message to generate the new identity for the sensor ID_{SN}^{new} and calculates $ID_{SN}^{new+} = ID_{SN}^{new} \| l* \| ID_{HN}$ from Formula (2) and make a new secret parameter $E = h\left(ID_{SN}^{new} \| h\left(l* \| ID_{HN}^{+}\right)\right)$ from Formula (5).

- The newly generated values are shared securely with the user and saved into the device's memories.
- SM broadcast the new identity to all the communicating nodes for future access.

5.6. Secure User Revocation

In the below steps, SM revokes the user card from the system to add new one and performs the following:

Step 1: The user sends a request to remove his/her previous card and adds a new SC_{NEW} to the system.

The user needs to normally log in to the session with his/her credentials, inserts SC, enters SID_u, PW_U, imprints the biometric BIO, and generates time stamp t_{i1}.

- After a successful log in the user generates secret value for the request validation.

$$M_1 = h('revocation \ \& \ replace',\ E, SID_U, PW_U\ ) \tag{20}$$

- Send this message to SM for card/mobile revocation.

Step 2: SM receives the user request to revoke from the system.

- SM checks the time validity $\Delta t = t_{j2} - t_{i1}$.
- If $\Delta t = t_{j2} - t_{i1} > 0$ then continue, else abort the session.
- SM checks the secret parameters and the request of the user via $E = h\left(ID_{SN} \ \| \ h\left(l* \ \| \ ID^{+}_{HN}\right)\right)$ from Formula (5).
- Send a secure link to the user to open, add his/her new ID^{new}_{U}, SPW^{new}_{U}, and BIO_{new}.
- Compute the following:

$$Gen(BIO_{new}) = \langle \delta_i | \tau_i \rangle \tag{21}$$

$$SPW^{new}_{U} = h\left(BIO_{new} \ \| \ ID^{+}_{FN} \| \ \delta_i\right) \tag{22}$$

- The user sends the new parameter securely over the secure one-time link to the SM.
- SM receives the request, generates new random value $rand_{new}$, and computes the following:

$$SID^{new}_{U} = h\left(ID^{new}_{U} \ \| \ ID^{+}_{FN} \| \ ID^{+}_{HN} \| \ rand_{new}\right) \tag{23}$$

$$F_{(i+1)} = h\left(SID^{new}_{U} \ \| \ ID^{+}_{FN} \| \ rand_{new}\right) \tag{24}$$

$$G_{new} = h\left(SID^{new}_{U} \ \| \ SPW \| \ l* \ \| \ rand_{new}\right) \tag{25}$$

$$H_{new} = h\left(rand_{new} \ \| \ ID^{+}_{FN} \| \ ID^{+}_{HN} \ \| \ ID^{+}_{SN}\right) \tag{26}$$

- Add the new values to the smart card SC_{NEW}.
- User receives the new smart card SC_{NEW}, replaces the new parameters $\left\{ F_{(i+1)},\ G_{new},\ H_{new}\right\}$, and retrieves the random number from H_{new}, then deletes it from the new smart card in a secure channel. The new set of parameters will be $\left\{ F_{(i+1)},\ G_{new} \right\}$.

5.7. Secure Data Transmission via Blockchain

In [12] scheme, there is no defined strategy to protect the stored data for retrieval or other usages after successful authentication. Since most of the WHSN structures are based on main centralized data storage that is accessible by the assigned doctor. So, this could put patient information in danger due to this source of error. Whereas the blockchain adds-up the data to blocks and splits them. Therefore, the integrity of the data is kept, each transaction is encrypted. Access control policies guarantee privacy [31]. Several methods were proposed to aid the purpose smart contract establishing

along with patient identity tracking. In the case of government authorities who want to evaluate a medical facility service or measure the spread of a disease, the authorities need to have access to all the citizens' information.

We adopted [32] method who proposed a Hyperledger blockchain which supports consensus algorithms that only permit the authenticated patients, and communications, and only accept reserved as well as confidential transactions. The Hyperledger blockchain consists of the transaction log that tackles all the changes made to the connections and changes the value of the world state. The blocks are built by a collection of transactions sent to the evaluator peer to simulate it, vote on it, and approve it. The structure of communication, electronic contracts, access policies are stored in the business network that the user can interact with from a mobile application connected to a server, where all the communication are encrypted by hashing to be able to access the blockchain for data storage and retrieval.

Another method is discussed by [21] that utilized the blockchain technology to store the individual data safely in the cloud. The sensor nodes contain some data that needs to be stored in the gateway safely for another retrieval or processing. The sensor sends encrypted data with the shared key to the foreign network along with the current timestamp. The foreign network node checks the timeliness and decrypts the data to get the information, then encrypts the data again with its pre-shared key to be sent to the hub node. The hub node decodes the info and checks the timestamp for validity to start building a data block. The block is added to the blockchain when all the communicated entities agreed upon the block contents in peer to peer cloud server network. After successfully gathering a group of valid data, the hub node starts to build transaction values and adds them together in one block to enable the system manager to create a blockchain of data for storage, deletion, update, and retrieve. The proposed method suggested the usage of cryptographic hash to encode the transmitted blocks and compute the "Merkle tree root" (MR) for the tree building. MR is a technique used in cryptocurrency to assure the data integrity in a peer-to-peer network structure. All the block information such as block owner and block payload are computed with the current block hash (CBHash). The hub node embeds the hashed identity of the user and sends the block of data to the system manager which uses "Ripple Protocol Consensus Algorithm (RPCA)" [33] for node verification and addition. Suppose that a user wants to access some data from a specific block, the user has to log in successfully to the connected hub node. So, as the hub node uses the user key that matches the user identity from the block, performs a hash function on data, decrypts the encrypted data to extract the hashes values and compare them with the computed hash for integrity check. Then, the hub node transmits the data to the user and the user decrypts the data with his/her key to retrieve the information from the block, as depicted in Figure 5.

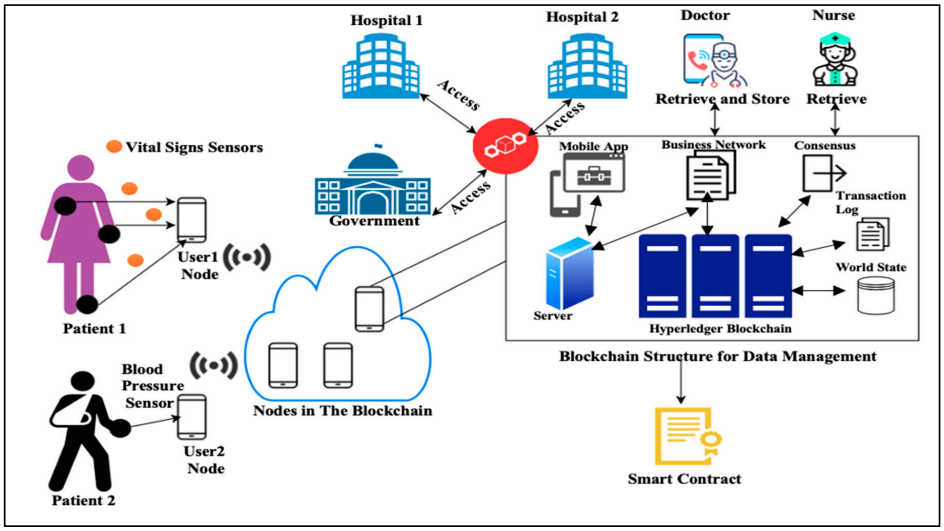

Figure 5. Medical record formation to the blockchain in the cloud.

6. Security Analysis

In this section, we discussed informal security analysis to analyze our scheme robustness against attacks in Section 6.2. In Section 6.3, we conducted a mathematical proof with BAN logic to confirm our structure mutual authentication and key agreement. Also, we simulate our model with Tamarin interactive tool to prove that our scheme is secure against session hijacking, replay attack, and attains perfect forward secrecy in Section 6.3.

6.1. Informal Security Evaluation

The below list indicated our system qualities.

6.1.1. Mutual Authentication

Mutual authentication ensures that all communicating objects authenticate each other at the same time. In our protocol, we conducted the mutual authentication phase and all interactions between *FN* and *HN* in the authentication phase, and we conducted BAN logic formal proof along with simulation in Tamarin protocol to prove the mutual authentication. Thence, our scheme accomplishes mutual authentication because *HN* checks both user identity in the formula $SID_u^* = SID_U$ along with the sensor node identity in $ID_{SN}^+ ? = ID_{SN}^+$ before calculating the secret parameter. So, the mutual authentication is achieved in our scheme.

6.1.2. Offline/Online Secret Shared Key Guessing

Regarding DY model, the intruder i can obtain all the parameters saved in the smart card, phone, *FN*, *SN*, and *HN*. Also, i can guess the perfect combination between username and password without the need to have SC or user mobile phone. Many elements protect our scheme from the attacks such as secret parameter $l*$, the fresh biometric of the user *BIO*, the random values $rand^*$, and $rand^+$ that are checked observing their freshness, the secret parameters between nodes F_{i1} and *E*, and the one-way hash function. Therefore, our scheme is robust against i shared secret key guessing in the online mode or offline mode.

6.1.3. Nodes Anonymity

In the initialization phase, we masked all the important communicating objects identities with random values, and secret parameter. We concealed the SN, FN, HN, ID_U, nodes identities, and biometric *BIO* in Formulas (2)–(4), (8), and (9), respectively. Therefore, the intruder i cannot trace where the data came from and where it goes because of the anonymity and dynamicity of the connected objects' identities. Moreover, the intruder i cannot guess the real identity of the user from SID_U because it is protected by the power of hash function and a random number. Also, the biometric of the user is a unique value and it is hashed with the threshold value which stop any kind of guessing to this parameter. As a result, both protocols are holding the anonymity feature.

6.1.4. Brute Force Attack

Intruder i can run a brute force attack on any identity, key or security parameters and can successfully know the correct parameters. Although our scheme makes it hard for the intruder i to guess the secure value correctly in polynomial time because we are implementing SHA-2 group of keys with size 224 bit, so by calculating the run-time of our key with one-way hash which is 2^{224}. The intruder i cannot perform a brute force attack on our scheme due to the hash key size. The system's key size fits our authentication procedure. However, it can be raised when necessary to attain security preliminaries in the future.

6.1.5. Stolen Smartcard Attack

In [34] scheme, they did not specify a method to prevent brute force attacks in case of the lost smart card. Their paper did not mention the concept of encrypting and locking smartcard data information with user biometric or password. Therefore, we suggest in the cryptanalysis to reduce the number of parameters, random number checking, as well as smart card blocking policy after three times error in entering the authentication biometric, and password. Moreover, encrypting and locking the card information with a password, and user biometric at each time to authenticate the user to the smartcard will guarantee the tamper-resistant feature when the card is lost. Thus, our scheme prevents stolen smartcard attack.

6.1.6. Replay Attack

All the communications between nodes during the authentication phase are protected by time stamp such that the communicating nodes generate new timestamp in each new parameter set creation. In the first communication between FN and HN, FN generates t_{i1} and computes $A = rand^* \oplus F_{i1} \oplus ID^+_{SN}$ and $B = rand^* \oplus SK$ for secret key as well as secret parameter confusion. In the second communication between HN and FN, HN generates t_{j2} before updating the security parameters along with masking values. Therefore, Intruder i will not be able to crack the real session key or the hidden arguments in a valid time during successful communication.

6.1.7. Integrity

Our scheme achieves integrity because all the security parameters, smart card parameters, biometric identity, and keys are protected with the one-way hash function. Moreover, the shadow identity of the user smart card is guarded with the formula $SID^*_U = h\left(F_{(i+1)} \parallel ID^+_{FN} \parallel rand\right)$, and the secret session key is protected by the formula $SK = h(ID_{SN} \parallel h(l* \parallel K_{MS}) \parallel ID_{FN} \parallel ID_{HN})$. So, integrity is held in our both proposed protocols withal anonymity and un-traceability.

6.1.8. Node Impersonation

Intruder i can compromise one communicating node and get its correct identity such as ID_{FN} the real identity of the user stored in the smartphone memory. Although that the SC password SPW along with session key SK is protected by one-way hash function $h(.)$ along with the biometric, valid random number and secret parameter $l*$. Besides, the intruder i is not able to compromise any other secret value or credential of the same communicating node or other nodes such as HN or SN. Subsequently, the proposed protocols are robust against any impersonation attack.

6.1.9. Session Hijacking Attack

Intruder i can freely hijack any passing message in the public insecure medium. Also, the intruder i can hijack all the parameters sent among the communicated entities, collect them, and process them differently to elude the system security. Our security parameters are transmitted in the public medium are as few as possible, so the attacker will not get any useful information from collecting and intercepting the transmitted parameters between channels. The identity of the user is shadowed and protected by a one-way hash function ($F_{(i+1)}$, A, B, t_{i1}) and (J, K, t_{j2}) where the attacker cannot guess the hidden parameters from the transmitted tuples. So, our proposed protocols are secure towards the session hijacking attack.

6.1.10. Collision Attack

Intruder i goes for many permutations to crack the cryptographic hash and recovers arguments. This malicious act is useless in the proposed techniques since it is difficult to obtain two distinct messages that encompass the equal value in hash function $h(m1) = h(m2)$. Thence, the robust hash

function should stop collision [35]. So, in line with [17] the SHA- 2 cryptographic hash operation with size of the keys: 224 bit, 256 bit, and 384 bit, respectively, is protected versus collision attack.

6.1.11. Scalability

Scalability is maintained when the growing of the system does not quite affect the performance of the system by increasing or decreasing a sensor or unit. In the case of fresh component adding or illegitimate component detection, our scheme is scalable by registering each user valid card, a sensor with unique security parameters, and IDs. Consequently, GW only permits the reliable nodes to make the connection and removes illegal nodes or cards in any future connection. Also, as per [36], to achieve scalability in the scheme, we should reduce the computation complexity for WHSN participating parties. Therefore, the core objective of this work is to boost the performance of [12] so we had accomplished our objective by decreasing cost of telecommunications.

6.1.12. Forward/Backward Secrecy

Forward secrecy evading is the capability of the intruder i to anticipate the potential key pair. Whereas backward secrecy happens when the attacker gathers as many previous keys as necessary to infer the former keys. The session keys are dynamic and secured in our schemes by several parameters such as random nonce, new foreign network identification, hidden value, and the timestamp. Thus, even though the intruder i correctly identified the keys, due to the complicated parametric method, he/she is unable to predict the future keys or breach the prior keys. In addition, the intruder i needs to correctly predict the following: $ID_{FN}^{++} = ID_{FN}^{+} \parallel l* \parallel ID_{HN}^{+}$, $ID_{HN}^{++} = ID_{SN}^{++} \parallel l* \parallel ID_{HN}^{+} \parallel ID_{FN}^{++}$, $SK^{+} = h\left(ID_{SN}^{++} \parallel h(l* \parallel K_{MS}) \parallel ID_{FN}^{++} \parallel ID_{HN}^{++}\right)$, $SID_{u}^{**} = h(ID_{U} \parallel ID_{FN}^{++} \parallel ID_{HN}^{++} \parallel rand^{+}$ to be able to disclose all the hidden data in the session. Consequently, optimal forward and backward secrecy is accomplished by our protocols.

6.1.13. Jamming Attack

Intruder i tries to disrupt the authentication process by generating a jamming signal to prevent the exchanging of some parameters during the communication. In our scheme, we enabled *FN* and *HN* to generate two random numbers that aided the key establishment. The last generated key and identities are used in the creation of the new parameters. So, the Intruder i needs to be aware of the formed session keys, identities, and random values to generate a successful jamming attack. Also, our scheme is protected by a timestamp that prevents the attacker from using old parameters after a long-time passage because the scheme will halt the expired session.

6.2. Ban Logic Proof

In this section, a formal proof with BAN logic method is conducted to prove our scheme mutual authentication and key agreement for P-I:

6.2.1. Essential Symbolization

The following covers the over-all fundamental representation for BAN logic to be employed in protocols P-I and P-II, see Table 3:

6.2.2. P-I: Goals

The goals to be achieved by P-I are stated below:

Goal 1 $\rightarrow FN \left| \equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$

Goal 2 $\rightarrow HN \left| \equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$

Goal 3 $\rightarrow FN \mid \equiv HN \mid \equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)$

$$\text{Goal 4} \rightarrow HN \mid\equiv FN \mid \equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)$$

P-I: Idealized Form

Below, we mentioned the ideal messages forms on the P-I:

$$Msg1: FN \rightarrow HN\ (ID_U,\ SID_U,\ rand^*,\ t_{i1})_{rand^*}$$
$$Msg2: HN \rightarrow FN\left(SID_U,\ ID_{SN}^{++},\ rand^*,\ rand^+,\ t_{j2}\right)_{rand^+}$$
$$Msg3: FN \rightarrow SN\left(SID_U,\ ID_{SN}^{++},\ rand^*,\ rand_{(i+1)},\ t_{I3}\right)_{rand^+}$$
$$Msg4: \rightarrow SN\left(SID_U,\ rand^*, rand_{(i+1)},\ t_{S4}\right)_{rand^+}$$

Table 3. Symbols used in the BAN (Burrows Abadi Nadeem) logic.

Symbols	Description
$M \mid\equiv N$	M trusts N
$M \Delta N$	M sees N
$M \mid\sim N$	M once responded N
$M \mid=\rangle N$	M governs N
$\#(N)$	N is new
$\langle N \rangle B$	N is merged with B
$\langle N \rangle B$	N is encrypted by B
$\mathrm{N} \overset{K}{\Leftrightarrow} Q$	K is shared secret between N and Q
SK	Pre-shared key used in connection

P-I: Assumptions

In the following, we explained the assumption of P_I:

$$\text{A1}: HN \left|\equiv \#(t_{i1})\right.$$
$$\text{A2}: FN \left|\equiv \#\left(t_{j2}\right)\right.$$
$$\text{A3}: SN \left|\equiv \#(t_{i3})\right.$$
$$\text{A4}: FN \left|\equiv \#(t_{S4})\right.$$
$$\text{A5}: HN \left|\equiv \#\left(FN \overset{rand^*}{\leftrightarrow} HN\right)\right.$$
$$\text{A6}: FN \left|\equiv \#\left(FN \overset{rand^*}{\leftrightarrow} HN\right)\right.$$
$$\text{A7}: SN \left|\equiv \#\left(FN \overset{rand^+}{\leftrightarrow} SN\right)\right.$$
$$\text{A8}: FN \left|\equiv \#\left(FN \overset{rand^+}{\leftrightarrow} SN\right)\right.$$
$$\text{A9}: FN \left|\equiv HN \Longrightarrow \#\left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$$
$$\text{A10}: HN \left|\equiv FN \Longrightarrow \#\left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$$
$$\text{A11}: FN \left|\equiv SN \Longrightarrow \#\left(FN \overset{SK^+}{\leftrightarrow} SN\right)\right.$$
$$\text{A12}: SN \left|\equiv FN \Longrightarrow \#\left(FN \overset{SK^+}{\leftrightarrow} SN\right)\right.$$

P-I: BAN Logic Proof

The BAN logic proof is processed as follows.

Step 1: according to $Msg1$ we get:

$$HN \leftarrow (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}$$

Step 2: using Step 1 with "message meaning rule":

$$HN|\equiv FN|\sim (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}$$

Step 3: using Step 2 and A1 with "freshness rule":

$$HN\left|\equiv \#(ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}\right.$$

Step 4: using Step 2 and Step 3 with "random nonce verification rule":

$$HN|\equiv FN|\equiv (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}$$

Step 5: according to $Msg2$ we get:

$$FN\leftarrow (SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j2})_{rand^+}$$

Step 6: using Step 5 and A6 "message meaning rule":

$$FN\leftarrow (SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j2})_{rand^+}$$

Step 7: using Step 6 and A3 "freshness rule":

$$FN\left|\equiv \#(SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j2})_{rand^+}\right.$$

Step 8: using Step 6 and Step 7 "random nonce verification rule":

$$FN|\equiv HN|\equiv (SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j2})_{rand^+}$$

Step 9: according to $Msg3$ we get:

$$SN\leftarrow (SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j3})_{rand^+}$$

Step 10: using Step 9 and A5 with "message meaning rule":

$$FN|\equiv SN|\sim (SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j3})_{rand^+}$$

Step 11: using Step 10 and A2 with "freshness rule":

$$FN\left|\equiv \#(SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j3})_{rand^+}\right.$$

Step 12: using Step 10 and Step 11 with "random nonce verification rule":

$$FN|\equiv SN|\equiv (SID_U, ID_{SN}^{++},\ rand^*, rand^+, t_{j3})_{rand^+}$$

Step 13: from $Msg4$ we get:

$$SN\leftarrow (SID_U,\ rand^*, rand^+, t_{S4})_{rand^*}$$

Step 14: using Step 13 and A7 with "message meaning rule":

$$SN|\equiv FN|\sim (SID_U,\ rand^*, rand^+, t_{S4})_{rand^*}$$

Step 15: using Step 14 and A4 with "freshness rule":

$$SN|\equiv FN \mid \#(SID_U,\ rand^*, rand^+, t_{S4})_{rand^*}$$

Step 16: using Step 14 and Step 15 with "random nonce verification rule":

$$SN|\equiv FN \mid \#(SID_U,\ rand^*, rand^+, t_{S4})_{rand^*}$$

Step 17: because $SPW = h(rand \ \| \ rand^+)$ we get Step 12 and Step 6 (Goal 3)

$$FN|\equiv HN \mid \equiv \left(FN \overset{SK^+}{\leftrightarrow} HN\right)$$

Step 18: because $SPW = h(rand \ \| \ rand^+)$, according to Step 4 and Step 8 (Goal 4)

$$HN|\equiv FN \mid \equiv \left(FN \overset{SK^+}{\leftrightarrow} SN\right)$$

Step 19: from A9 and Step 17 (Goal 1)

$$FN \left|\equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$$

Step 20: from A10 and Step 18 (Goal 2)

$$HN \left|\equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$$

In this section, a formal proof with BAN logic method is conducted to prove our scheme mutual authentication and key agreement for P-II:

6.2.3. P-II: Goals

The ideal goals to be achieved by P-II are stated as follows:

Goal 1 $\rightarrow FN \left|\equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$

Goal 2 $\rightarrow HN \left|\equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)\right.$

Goal 3 $\rightarrow FN \mid\equiv HN \mid \equiv \left(FN \overset{SK^+}{\leftrightarrow} HN\right)$

Goal 4 $\rightarrow HN \mid\equiv FN \mid \equiv \left(FN \overset{SK^+}{\leftrightarrow} HN\right)$

P-II: Idealized Form

Below, we illustrated the idealized form of the message to be transmit between nodes in P-II:

$Msg1: FN \rightarrow HN\ (ID_U,\ SID_U,\ rand^*,\ t_{i1})_{rand^*}$

$Msg2: HN \rightarrow FN\left(SID_U,\ ID_{SN}^{++},\ rand^*,\ rand^+,\ t_{j2}\right)_{rand_{(i+1)}}$

$Msg3: FN \rightarrow SN\left(SID_U,\ ID_{SN}^{++},\ rand^*,\ rand^+,\ t_{I3}\right)_{rand_{(i+1)}}$

$Msg4: \rightarrow SN\ (SID_U,\ rand^*,\ rand^+,\ t_{S4})_{rand_{(i+1)}}$

P-II: Assumptions

Same assumption as before just change A5–A8:

$A5: HN \left|\equiv \#\left(FN \overset{rand_{(i+1)}}{\leftrightarrow} HN\right)\right.$

$$\text{A6}: FN \left| \equiv \#\left(FN \overset{rand_{(i+1)}}{\leftrightarrow} HN\right)\right.$$
$$\text{A7}: SN \left| \equiv \#\left(FN \overset{rand_{(i+1)}}{\leftrightarrow} SN\right)\right.$$
$$\text{A8}: FN \left| \equiv \#\left(FN \overset{rand_{(i+1)}}{\leftrightarrow} SN\right)\right.$$

P-II: BAN Logic Proof

The BAN logic proof then proceeds as below.

Step 21: According to *Msg*1 we get:

$$HN \leftarrow (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}$$

Step 22: Using Step 21 with "message meaning rule":

$$HN| \equiv FN| \sim (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}$$

Step 23: Using Step 22 and A1 with "freshness rule":

$$HN\left| \equiv \#\ (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}\right.$$

Step 24: Using Step 22 and Step 23 with "random nonce verification rule":

$$HN| \equiv FN| \equiv (ID_U,\ SID_U,\ rand^*, t_{i1})_{rand^*}$$

Step 25: According to *Msg*2 we get:

$$FN \leftarrow\ (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j2})_{rand_{(i+1)}}$$

Step 26: Using Step 25 and A6 "message meaning rule":

$$FN \leftarrow\ (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j2})_{rand_{(i+1)}}$$

Step 27: Using Step 26 and A3 "freshness rule":

$$FN\left| \equiv \#\ (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j2})_{rand_{(i+1)}}\right.$$

Step 28: Using Step 26 and Step 27 "random nonce verification rule":

$$FN| \equiv HN| \equiv (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j2})_{rand_{(i+1)}}$$

Step 29: According to *Msg*3 we get:

$$SN \leftarrow (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j3})_{rand_{(i+1)}}$$

Step 30: Using Step 29 and A5 with "message meaning rule":

$$FN| \equiv SN| \sim (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j3})_{rand_{(i+1)}}$$

Step 31: Using Step 30 and A2 with "freshness rule":

$$FN \Big| \equiv \#(SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j3})_{rand_{(i+1)}}$$

Step 32: Using Step 30 and Step 31 with "random nonce verification rule":

$$FN| \equiv SN \mid \equiv (SID_U, ID_{SN}^{++},\ rand^*, rand_{(i+1)}, t_{j3})_{rand_{(i+1)}}$$

Step 33: From $Msg4$ we get:

$$SN \ \leftarrow (SID_U,\ rand^*, rand_{(i+1)}, t_{S4})_{rand^*}$$

Step 34: According Step 33 and A7 with "message meaning rule":

$$SN| \equiv FN \mid \sim (SID_U,\ rand^*, rand_{(i+1)}, t_{S4})_{rand^*}$$

Step 35: From Step 34 and A4 with "freshness rule":

$$SN| \equiv FN \mid \#(SID_U,\ rand^*, rand_{(i+1)}, t_{S4})_{rand^*}$$

Step 36: Using Step 34 and Step 35 with "random nonce verification rule":

$$SN| \equiv FN \mid \#(SID_U,\ rand^*, rand_{(i+1)}, t_{S4})_{rand^*}$$

Step 37: Because $SPW = h\left(rand_{(i+1)} \ \| \ rand^*\right)$ we get Step 32 and Step 26 (Goal 3)

$$FN| \equiv HN \mid \equiv \left(FN \overset{SK^+}{\leftrightarrow} HN\right)$$

Step 38: Due $SPW = h\left(rand_{(i+1)} \ \| \ rand^*\right)$, according to Step 24 and Step 28 (Goal 4)

$$HN| \equiv FN \mid \equiv \left(FN \overset{SK^+}{\leftrightarrow} SN\right)$$

Step 39: From A9 and Step 37 (Goal 1)

$$FN \Big| \equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)$$

Step 40: From A10 and Step 38 (Goal 2)

$$HN \Big| \equiv \left(FN \overset{SK}{\leftrightarrow} HN\right)$$

6.3. Simulation with Tamarin Prover

We simulated our scheme with Tamarin prover [37] to prove our scheme robustness against session hijacking, replay attacks, perfect forward secrecy, and mutual authentication. It is a tool used for formal protocols validation and written in the Haskell language. The simulation was operated on MacBook Air, it ran on MacOS Catalina, with processor 1.8 GHz Dual-Core Intel Core i5, Memory 8 GB 1600 MHz DDR3, and Intel HD Graphics 6000 1536 MB. Also, we uploaded some tools to help our system to simulate the protocol which are graphviz version 2.44.1, maude tool, SAPIC tool, and sublime text to show colorful coding for the protocol syntax.

Haskell Specification

We specified our nodes in the communication model as *FN* (user), *HN* (gateway), and *SN* (sensor node) to be represented in the Tamarin environment with their specified interaction along with attack simulation to ensure the scheme validity and robustness against the simulated attacks.

In Figure 6, we showed how nodes, secret key, biometric, and smart card are registered by the SM over a secure channel. Then when the user received the smart card, the user registers the biometric *BIO*, the identity SID_U, and the password to form the following parameters $\{ F_{(i+1)},\ G\ Gen(\ ),\ Rep(\ ),\ h\ (\ ) \}$. The registered client inserts the smart card *SC*, and starts the authentication between the *FN* and *HN*.

```
rule Register_USER:
let
pskn = KDF(<'f0', ~idsn, h(<v, Kidsn>), idhn, idfn>)
idfnnew = idfn XOR idhn XOR v XOR ~idsn
idsnnew = ~idsn XOR v XOR idhn
idhnnew = ~idsn XOR v XOR idhn
sc = h(idfnhn, bio)
fi = h(idfn,h(<v,idhnnew>))
sidu = h(<idfnhn, idhnnew, ~rand>)
fii = h(<sidu, idfnnew, ~rand>)
g = h(<sidu, sc, v, ~rand>)
in [ Fr(~rand), !Sharevaluessn(~idsn, idhn, pskn), !Sharevaluesfn(sidu, idhn, pskn), In(bio)  ]
--[!Primery_session_key(sidu, idhn, pskn)]->
[Out(sidu), !Primery_session_keyt(sidu, idhn, pskn, idsnnew)]
```

Figure 6. Initialization and registration phase for the user and other nodes.

We showed how the *FN* gets all the parameters from the user and calculates the masking parameters ($F_{(i+1)}$, A, B, t_{i1}) to be transmitted to the *HN*, as depicted in Figure 7.

```
rule FN_3send:

let A = ~randd XOR fi XOR idsnnew
B = ~randd XOR pskn

        in
[ Fr(~randd)
, St_FN_2(fii, id, A, $idhn, B), In(pskn),In(pskn)
]
--[ FN_SendEncrypted(fii,~randd) ]->
[ St_FN_3(fii, id, A, $idhn, B, ~randd) ]
```

Figure 7. *FN* send authentication request to the *HN*.

Next, *HN* receives the authentication parameters and authenticates the user to start calculating the new set of parameters (J, K, t_{j2}) for the card secret data renewal as well as the nodes' identities and secret keys, as depicted in Figures 8 and 9.

```
rule HN_1send:
let
IDFN_NEW = <idhnnew ,S,idfnnew>
IDSN_NEW = <idsnnew, idhnnew,S>
IDHN_NEW = <idsnnew, idhnnew,S, idfnnew>
sidunew = h(<idu, IDFN_NEW, IDHN_NEW, ~randdd>)
fiinew = h(<sidunew, IDFN_NEW, ~randdd>)
gnew = h(<sidunew, sc, S, ~randd>)
J = IDSN_NEW XOR ~rand
K = IDFN_NEW XOR ~randdd
NSK = KDF(<'NSK',pskn,S, IDFN_NEW,IDSN_NEW, S, IDHN_NEW >)
MAC = KDF(<'f1', pskn, 'AMF', ~rand>)
XRES1 = KDF(<'f2', pskn, ~rand, '1'>) XRES2 = KDF(<'f2', pskn, ~rand, '2'>)
XRES = <XRES1,XRES2>
 AUTN = < 'AMF', MAC>
AV_STAR = <~rand, XRES1, NSK, AUTN>
AV_STARenc = senc{<S,AV_STAR>}SK
in
[ St_HN_10($idhn, id, sidunew, pskn, idsn, SK, S),
  // Modeling trick to get rid of partial deconstructions

 Fr(~randdd), In (idfnnew), In (idhnnew),  In (~rand),  In (idu), In (sc)
]
--[ HN_4SendEncrypted($idhn, <S,AV_STAR>)
, Running($idhn, sidunew, <'SNN','HN',NSK>)
, Honest(<'SN',sidunew>) ]->
[ St_HN_2($idhn, id, sidunew, pskn, idsn, SK, S, ~randdd)
, Out(AV_STARenc), Out(J), Out(K)
]
```

Figure 8. *HN–FN* authentication and new parameters creation.

```
rule USER_5recv_SEQGOOD:
let IDFN_NEW = <idhnnew ,S,idfnnew>
IDSN_NEW = <idsnnew, idhnnew,S>
IDHN_NEW = <idsnnew, idhnnew,S, idfnnew>
sidunew = h(<idu, IDFN_NEW, IDHN_NEW, ~randdd>)
fiinew = h(<sidunew, IDFN_NEW, ~randdd>)
gnew = h(<sidunew, sc, S, ~randdd>)
J = IDSN_NEW XOR ~rand
K = IDFN_NEW XOR ~randdd
NSK = KDF(<'NSK',pskn,S, IDFN_NEW, IDSN_NEW, S, IDHN_NEW >)
MAC = KDF(<'f1', pskn, 'AMF', ~rand>)
XRES1 = KDF(<'f2', pskn, ~rand, '1'>) XRES2 = KDF(<'f2', pskn, ~rand, '2'>)
XRES = <XRES1,XRES2>
 AUTN = <'AMF', MAC>
AV_STAR = <~rand, XRES1, NSK, AUTN>
AV_STARenc = senc{<S,AV_STAR>}SK
m = <~randdd,AUTN>
in
[ In(m), In(sidunew)
, St_SNN_211(idsnnew, Kidsn, ~id, $idhn, $SNid),
// Modeling trick to get rid of partial deconstructions
 St_FN_50(fiinew, ~id, IDSN_NEW, $idhn, SK, S, AV_STAR)
]
--[
 Secret(<'SN',fiinew>, NSK)
, Running(idsnnew, $SNid, <'FN','SNN',NSK>) , Honest(<'FNHN',$SNid,$idhn>)
, Honest(<'SN',sidunew>)
, Running(idsnnew,$idhn,<'HN','SNN',<'RAND',~randdd>>)
, USER_5recv_SEQGOOD(~id)
, Fr(~randdd) ]->
[ St_SNN_3(idsnnew, Kidsn, ~id, $idhn, $SNid, ~randdd, NSK)]
```

Figure 9. *FN* new parameters calculation and honesty verification.

We specified some lemmas to ensure our parameters and nodes secrecy in a matter of session hijacking and guarantee that our scheme holds perfect forward secrecy, as depicted in Figure 10.

```
lemma secrecy_HN:
"All A x #i. Secret(<'HN',A>,x) @i ==> not (Ex #j. K(x)@j)
| (Ex X #r. Reveal(X)@r & Honest(X) @i)"

lemma secrecy_FN:
"All A x #i. Secret(<'FN',A>,x) @i ==> not (Ex #j. K(x)@j)
| (Ex X #r. Reveal(X)@r & Honest(X) @i)"
// FINDS _NO_ ATTACK AUTOMATICALLY (see following lemma)
```

Figure 10. Session hijacking and perfect forward secrecy resistance lemmas in *HN* and *FN*.

Also, we specified other lemmas to prove our scheme parameters secrecy against replay attack, as depicted in Figure 11.

```
lemma secrecy_PFS_HN:
"All A x #i. Secret(<'HN',A>,x) @i ==> not (Ex #j. K(x)@j)
| (Ex X #r. Reveal(<'SN',X>)@r
& Honest(<'SN',X>) @i & r < i)"

lemma secrecy_PFS_HN_special:

"(not (Ex idfn idhn #j. Reveal(<'FNHN',idfn,idhn>)@j))
==>
All A x #i.
Secret(<'HN',A>,x) @i ==> not (Ex #j. K(x)@j)
| (Ex X #r. Reveal(<'SN',X>)@r
& Honest(<'SN',X>) @i & r < i)"
// FINDS _NO_ ATTACK AUTOMATICALLY (see following lemma)
lemma secrecy_PFS_FN:
"All A x #i. Secret(<'FN',A>,x) @i ==> not (Ex #j. K(x)@j)
| (Ex X #r. Reveal(<'SN',X>)@r
& Honest(<'SN',X>) @i & r < i)"
// (this enforces an attack without
// reuses the HELPERsecrecy lemmas
lemma secrecy_PFS_FN_special:

"(not (Ex idfn idhn #j. Reveal(<'FNHN',idfn,idhn>)@j))
==>
All A x #i.
Secret(<'FN',A>,x) @i ==> not (Ex #j. K(x)@j)
| (Ex X #r. Reveal(<'SN',X>)@r
& Honest(<'SN',X>) @i & r < i)"
```

Figure 11. Replay attack resistance lemmas in *HN* and *FN*.

The below results describe that the scheme holds the property by highlighting the codes in green color. Our scheme fulfils the perfect forward secrecy, resistance against replay attack, and session hijacking attack in both *HN* and *FN*, as depicted in Figures 12 and 13, respectively.

```
lemma secrecy_PFS_FN:
  all-traces
  "∀ A x #i.
          (Secret( <'FN', A>, x ) @ #i) ⇒
          ((¬(∃ #j. K( x ) @ #j)) ∨
           (∃ X #r.
             ((Reveal( <'SN', X> ) @ #r) ∧
              (Honest( <'SN', X> ) @ #i)) ∧
             (#r < #i)))"
simplify
solve( Secret( <'FN', A>, x ) @ #i )
  case SN_6recv_NOSYNCFAILURE
  by solve( St_FN_5( A, id, IDSN_NEW, $idhn, SK, ~S,
                     <x, XRES1, x.1, 'AMF', MAC>
            ) ►₀ #i )
qed

lemma secrecy_PFS_FN_special:
  all-traces
  "(¬(∃ idfn idhn #j.
            Reveal( <'FNHN', idfn, idhn> ) @ #j)) ⇒
        (∀ A x #i.
          (Secret( <'FN', A>, x ) @ #i) ⇒
          ((¬(∃ #j. K( x ) @ #j)) ∨
           (∃ X #r.
             ((Reveal( <'SN', X> ) @ #r) ∧
              (Honest( <'SN', X> ) @ #i)) ∧
             (#r < #i))))"
simplify
solve( Secret( <'FN', A>, x ) @ #i )
  case SN_6recv_NOSYNCFAILURE
  by solve( St_FN_5( A, id, IDSN_NEW, $idhn, SK, ~S,
                     <x, XRES1, x.1, 'AMF', MAC>
            ) ►₀ #i )
qed
```

Figure 12. Replay attack resistance in *FN*.

```
lemma secrecy_PFS_HN_special:
  all-traces
  "(¬(∃ idfn idhn #j.
            Reveal( <'FNHN', idfn, idhn> ) @ #j)) ⇒
        (∀ A x #i.
          (Secret( <'HN', A>, x ) @ #i) ⇒
          ((¬(∃ #j. K( x ) @ #j)) ∨
           (∃ X #r.
             ((Reveal( <'SN', X> ) @ #r) ∧
              (Honest( <'SN', X> ) @ #i)) ∧
             (#r < #i))))"
simplify
solve( Secret( <'HN', A>, x ) @ #i )
  case HN_7recv_NOSYNCFAILURE
  by solve( St_HN_21( $idhn, id, IDSN_NEW, pskn,
idfn, SK, S, ~randdd
            ) ►₀ #i )
qed

lemma secrecy_PFS_FN:
  all-traces
  "∀ A x #i.
          (Secret( <'FN', A>, x ) @ #i) ⇒
          ((¬(∃ #j. K( x ) @ #j)) ∨
           (∃ X #r.
             ((Reveal( <'SN', X> ) @ #r) ∧
              (Honest( <'SN', X> ) @ #i)) ∧
             (#r < #i)))"
simplify
solve( Secret( <'FN', A>, x ) @ #i )
  case SN_6recv_NOSYNCFAILURE
  by solve( St_FN_5( A, id, IDSN_NEW, $idhn, SK, ~S,
                     <x, XRES1, x.1, 'AMF', MAC>
            ) ►₀ #i )
qed
```

Figure 13. Replay attack resistance in *HN*.

The below graphs illustrate, that our scheme fulfils the perfect forward secrecy, resistance against replay attack, and session hijacking attack in both *HN* and *FN* with absence of the red arrows in the figures, as depicted in both Figures 14 and 15.

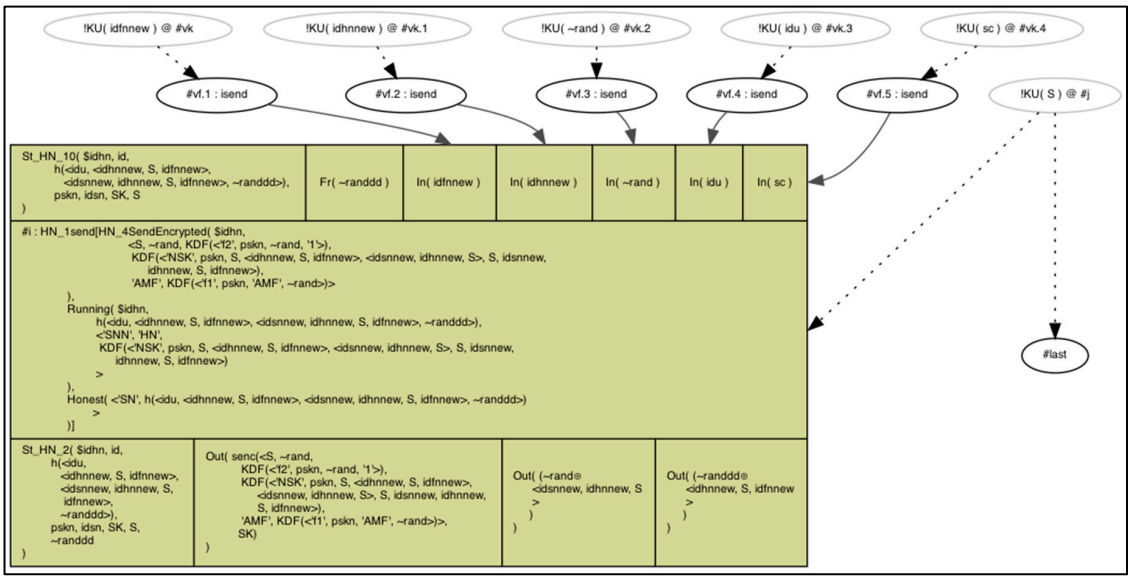

Figure 14. Secrecy of the scheme parameters during transmission on the public channel by *HN*.

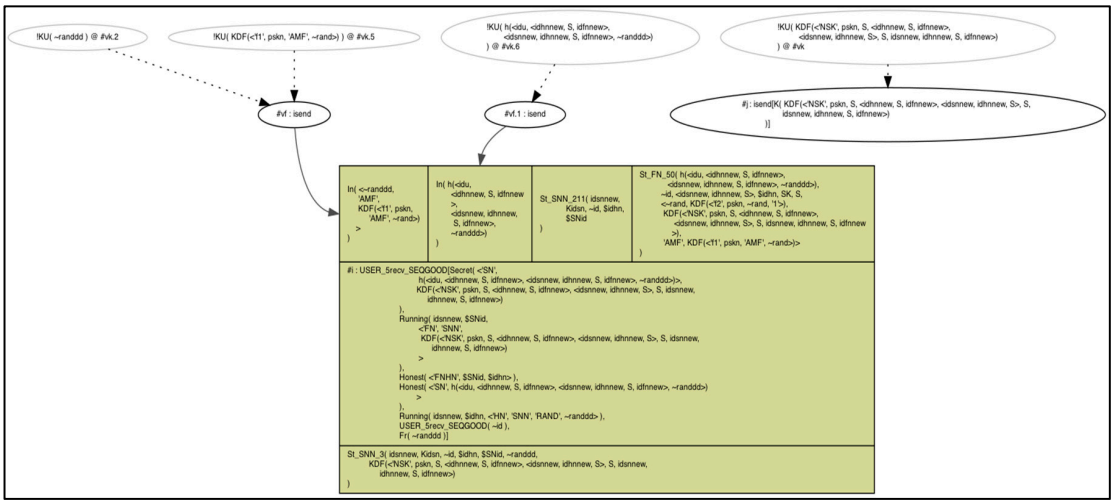

Figure 15. Secrecy of the scheme parameters during reception from the public channel by *FN*.

In the next section, we provided the performance analysis to the scheme to show its efficiency in comparison with other schemes.

7. Performance Analysis

In the following, we deliberated the efficiency of the proposed protocols regarding their computation, communication, and storage.

7.1. Computational Cost

In this section, the computation cost calculation is performed for the proposed protocols that employed a cryptographic hash which takes 0.00032 s along with a biometric reproduce operation that takes 0.0171 s based on the metrics in [38]. The computational cost of the proposed scheme is better than all other schemes in the foreign network side by 60% and 65% and *HN* side [17,19,22] with a 80% by using P-I and 85% by using P-II. Besides, P-I and P-II perform better than [12,16] in the foreign network side along with *HN* side with 5% and 15% reduction, respectively, as depicted in Figures 16 and 17. Furthermore, we chose a hash function with a 224 bit key size to allow the foreign

network to have an adequate level of security better than [12] which takes a 160 bit key size, and [19,22] which takes a 128 bit size key, (see Tables 4–7).

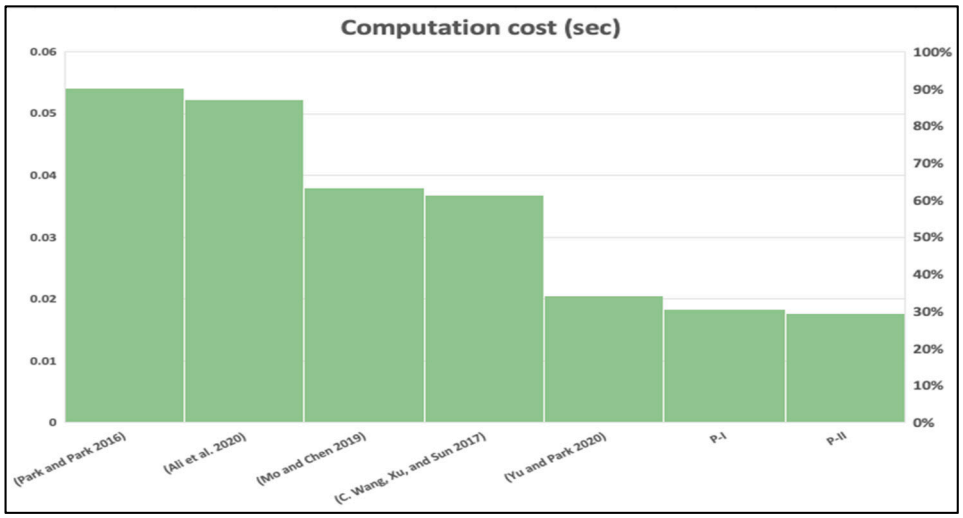

Figure 16. Computation time in foreign network node.

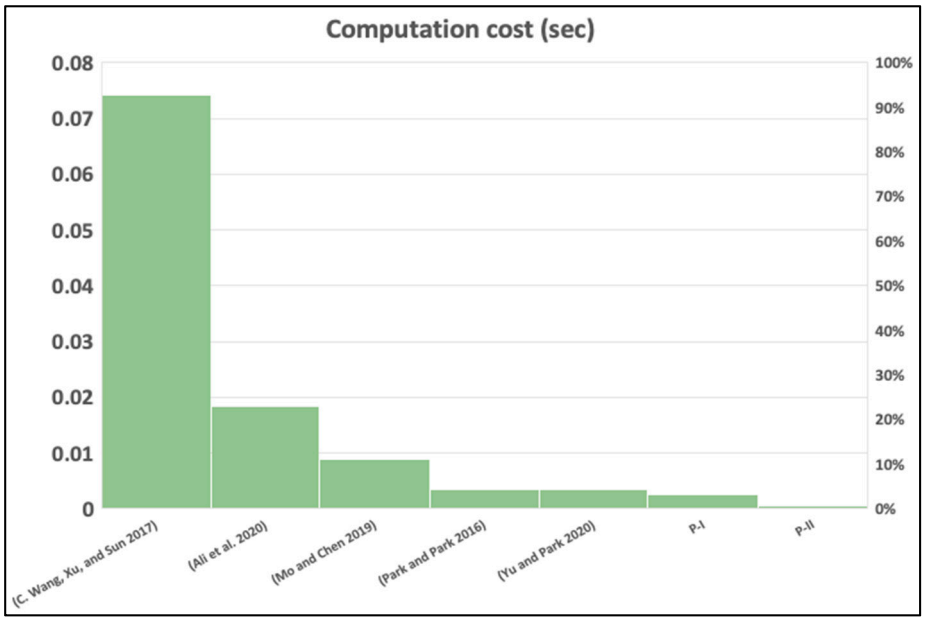

Figure 17. Computation time in hub node.

Table 4. Rough estimated time for various schemes [38].

Notation	Description	Computation Time in Seconds
T_h	One-way hash function	0.00032
T_{ecm}	ECC point multiplication	0.0171
T_{eca}	ECC point addition	0.0044
T_{senc}	Symmetric key encryption	0.0056
T_{sdec}	Symmetric key decryption	0.0056
T_{me}	Modular exponentiation	0.0192
T_{fe}	Fuzzy extractor	0.0171
T_R	Reproduce operation	0.0171

Table 5. Key sizes of the schemes.

Scheme	Key Size
[12]	160 bits
[16]	1024 bits
[17]	1024 bits
[19]	128 bits
[22]	128 bits
P-I and P-II	224 bits

Table 6. Comparison of the scheme computation cost.

Scheme	[12]	[16]	[17]	[19]	[22]	P-I	P-II
Foreign Network	$11Th + TR$	$9Th + TF + 2T_{ecm}$	$2T_{ecm} + 8Th$	$12Th + TR + 2Tme$	$2Tecm + 3Th + 1Tfe$	$4\ Th + TR$	$2Th + TR$
Hub	$11Th$	$11Th$	$4T_{ecm} + 18Th$	$10Th + Tsenc$	$1Tecm + 4Th$	$8\ Th$	$2Th$

Table 7. Comparison of the scheme computation time.

Scheme	[12]	[16]	[17]	[19]	[22]	P-I	P-II
Foreign Network	0.02062 s	0.03708 s	0.03676 s	0.05514 s	0.05226 s	0.01838 s	0.01774 s
Hub	0.00352 s	0.00352 s	0.07416 s	0.0088 s	0.01838 s	0.00256 s	0.00064 s

7.2. Communication Overhead

We assumed the length of the hash function, keys, and security parameters = 224 bits, and the timestamp = 32 bits. Besides, our system contains four tuples in the foreign network side ($F_{(i+1)}$, A, B, t_{i1}) that results in = 224 + 224 + 224 + 32 = 704 bit. Moreover, we have (J, K, t_{j2}) from *HN* to *FN* that results in = 224 + 224 + 32 = 480 bit. Those results demonstrate that our system has the least overhead in a GW side more than all schemes in the comparison [16,17,19,22], with more strength versus numerous attacks, as shown in Table 8.

Table 8. Comparison of the scheme communication overhead.

Scheme	[12]	[16]	[17]	[19]	[22]	P-I	P-II
Foreign Network	672 bits	1536 bits	1408 bits	896 bits	640 bits	704 bits	480 bits
Hub	512 bits	4096 bits	3968 bits	768 bits	544 bits	480 bits	480 bits

7.3. Storage Overhead

We determined the storage cost of our work in contrast to [16,17,19,22] schemes to analyze the schemes' capacities. Assuming that each function and parameter of the following have different storage bytes such that hash, ECC, AES symmetric, RSA asymmetric, parameters identifications, random number, and time are 20, 20, 20, 20, 4, 4, and 16 bytes, respectively, and the prime p in $Ep\ (a,\ b)$ is 20 bytes. The suggested scheme requires storage for the stored arguments $\left\{ F_{(i+1)},\ G_{new} \right\}$ that results in $(20 + 20 = 40\ bytes)$ for the smartcard, and *rand* requires 20 bytes for the gateway. The storage cost distinguishes our scheme from others because it is the lowest of all on the smart card side. Moreover, the number of stored security parameters in the proposed structure will provide better security among other schemes, as shown in Tables 9 and 10.

Table 9. Storage overhead computation.

Scheme	Stored Data (Foreign Network/SC)	Stored Data (Hub)
[12]	Q_i, W_i, MID_i $\approx$ 60 *bytes*	r_g $\approx$ 20 *bytes*
[16]	A_i, B_i, C_i, TID_i $\approx$ 64 *bytes*	TID_i, RN_G $\approx$ 24 *bytes*
[17]	A_i, B_i, n_0, Y, P $\approx$ 100 *bytes*	ID_i, r_i, $Honey_{List}$ $\approx$ 20 *bytes*
[19]	RID_i, f_i, τ $\approx$ 56 *bytes*	K_j $\approx$ 20 *bytes*
[22]	U'_{priv}, t, τ_i, X_i $\approx$ 56 *bytes*	K_{priv} $\approx$ 20 *bytes*
P-I	$F_{(i+1)}$, G $\approx$ 40 *bytes*	*rand* $\approx$ 20 *bytes*

Table 10. Comparison of scheme security requirements.

Features	[12]	[16]	[17]	[19]	[22]	Ours
Offline/online shared secret guessing	×	×	×	×	×	✓
Anonymity	×	×	×	×	×	✓
Brute force attack	×	✓	✓	✓	×	✓
FN-SN Replay attack	✓	✓	✓	✓	✓	✓
FN/*HN* Impersonation	×	×	✓	✓	✓	✓
Session hijacking	×	×	×	×	✓	✓
Blockchain data transmission	N/A	N/A	N/A	N/A	N/A	✓
Integrity	✓	✓	✓	✓	✓	✓
Eavesdropping attack	×	×	✓	✓	✓	✓
Re-authentication	N/A	N/A	N/A	N/A	N/A	✓
Un-traceability	✓	×	×	×	✓	✓
Collision attack	-	✓	✓	✓	×	✓
User revocation	N/A	✓	N/A	✓	N/A	✓
Jamming attack	×	×	×	×	✓	✓
Stolen smartcard	×	×	×	×	×	✓

From Table 6, we compared our proposed scheme computation cost to schemes, and we identified that our scheme performs better than all in both foreign network and hub node sides. Y. Park and Y. Park scheme [16] scheme contains 20 hashes, a fuzzy extractor, and 2 ECC point multiplications. C. Wang et al. scheme [17] requires 8 hashes, and 26 ECC point multiplications. Mo and Chen scheme [19] takes 22 hashes, a reproduction operation, 2 Modular exponentiations, and 1 symmetric encryption. Similarly, Ali et al. scheme [22] scheme needs 7 hashes, 3 ECC point multiplications, and a fuzzy extractor. Yu and Park scheme [12] takes 22 hashes, and a reproduction function. In comparison to the proposed scheme, our authentication protocol requires 12 hashes along with 1 reproduction function, and the re-authentication protocol requires 4 hashes and 1 reproduction function. This manifests that our scheme has a lower computational cost and low energy consumption.

8. Conclusions

WHSN is a modern trend that deals with significant information from the patients that must be protected. It received major attention from the information security developers and vendors, who put big efforts in increasing the guardedness of the WHSN system and speed up the performance. Therefore, we analyzed the latest schemes in the field and we found that [12] to be the most efficient and secure one. So, we cryptanalyze it and we discovered that the scheme needs enhancement to achieve both security and performance. Consequently, a three-factor authentication scheme based on the biometric, smart card, and password is proposed. The scheme was formally validated by BAN logic and simulated with Tamarin prover to confirm its security and mutual authentication. Moreover, the informal analysis proved that the above scheme achieved the suggested security requirements like, anonymity, offline/online shared secret guessing, *FN-SN* replay attack, brute force attack, *FN*/*HN* impersonation, integrity, session hijacking, eavesdropping attack, un-traceability, and collision attack. Finally, we conducted performance evaluation to compute our scheme efficiency and we found out that our scheme has better computation cost, communication cost, storage cost, and energy consumption than the related schemes. To conclude,

the future direction of our research will employ blockchain technology in WHSN authentication in-depth and more attacks simulation in the proverif tool.

Author Contributions: Conceptualization, A.M.A. and K.S.A.; methodology, A.M.A. and K.S.A.; software, A.M.A. and K.S.A.; validation, A.M.A. and K.S.A.; formal analysis, A.M.A. and K.S.A.; writing–original draft preparation, A.M.A. and K.S.A.; writing–review and editing, A.M.A. and K.S.A.; supervision, A.M.A.; funding acquisition, A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Deanship of Scientific Research at Imam Abdulrahman Bin Faisal University, Saudi Arabia for providing financial support for this research via grant number 2021-062-CSIT.

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

References

1. Park, Y.; Lee, S.; Kim, C.; Park, Y. Secure biometric-based authentication scheme with smart card revocation/reissue for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 15501477–16658607. [CrossRef]
2. Chen, C.-M.; Wang, K.-H.; Yeh, K.-H.; Xiang, B.; Wu, T.-Y. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3133–3142. [CrossRef]
3. Chen, C.-M.; Xiang, B.; Wu, T.-Y.; Wang, K.-H. An Anonymous Mutual Authenticated Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks. *Appl. Sci.* **2018**, *8*, 1074. [CrossRef]
4. Rashid, B.; Rehmani, M.H. Applications of wireless sensor networks for urban areas: A survey. *J. Netw. Comput. Appl.* **2016**, *60*, 192–219. [CrossRef]
5. Boni, K.R.C.; Xu, L.; Chen, Z.; Baddoo, T.D. A Security Concept Based on Scaler Distribution of a Novel Intrusion Detection Device for Wireless Sensor Networks in a Smart Environment. *Sensors* **2020**, *20*, 4717. [CrossRef] [PubMed]
6. Das, A.K.; Sutrala, A.K.; Kumari, S.; Odelu, V.; Wazid, M.; Li, X. An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. *Secur. Commun. Netw.* **2016**, *9*, 2070–2092. [CrossRef]
7. Abawajy, J.H.; Hassan, M.M. Federated Internet of Things and Cloud Computing Pervasive Patient Health Monitoring System. *IEEE Commun. Mag.* **2017**, *55*, 48–53. [CrossRef]
8. Rghioui, A.; Lloret, J.; Oumnad, A. Big Data Classification and Internet of Things in Healthcare. *Int. J. E-Health Med Commun.* **2020**, *11*, 20–37. [CrossRef]
9. Teshome, A.K.; Kibret, B.; Lai, D.T. A Review of Implant Communication Technology in WBAN: Progress and Challenges. *IEEE Rev. Biomed. Eng.* **2018**, *12*, 88–99. [CrossRef]
10. Gogate, U.D.; Bakal, J. Healthcare Monitoring System Based on Wireless Sensor Network for Cardiac Patients. *Biomed. Pharmacol. J.* **2018**, *11*, 1681–1688. [CrossRef]
11. Bhatia, T.; Verma, A.; Sharma, G. Towards a secure incremental proxy re-encryption for e-healthcare data sharing in mobile cloud computing. *Concurr. Comput. Pr. Exp.* **2020**, *32*, 5520. [CrossRef]
12. Yu, S.; Park, Y. SLUA-WSN: Secure and Lightweight Three-Factor-Based User Authentication Protocol for Wireless Sensor Networks. *Sensors* **2020**, *20*, 4143. [CrossRef] [PubMed]
13. Almuhaideb, A.M.; Alqudaihi, K.S. A Lightweight and Secure Anonymity Preserving Protocol for WBAN. *IEEE Access* **2020**, *8*, 178183–178194. [CrossRef]
14. Xu, Z.; Xu, C.; Chen, H.; Yang, F. A lightweight anonymous mutual authentication and key agreement scheme for WBAN. *Concurr. Comput. Pr. Exp.* **2019**, *31*, 5295. [CrossRef]
15. 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. [CrossRef]
16. Park, Y.; Park, Y. Three-Factor User Authentication and Key Agreement Using Elliptic Curve Cryptosystem in Wireless Sensor Networks. *Sensors* **2016**, *16*, 2123. [CrossRef]
17. Wang, C.; Xu, G.; Sun, J. An Enhanced Three-Factor User Authentication Scheme Using Elliptic Curve Cryptosystem for Wireless Sensor Networks. *Sensors* **2017**, *17*, 2946. [CrossRef]
18. Challa, S.; Das, A.K.; Odelu, V.; Kumar, N.; Kumari, S.; Khan, M.K.; Vasilakos, A.V. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2018**, *69*, 534–554. [CrossRef]

19. Mo, J.; Chen, H. A Lightweight Secure User Authentication and Key Agreement Protocol for Wireless Sensor Networks. *Secur. Commun. Netw.* **2019**, *2019*, 1–17. [CrossRef]
20. Lu, Y.; Xu, G.; Li, L.; Yang, Y. Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wirel. Netw.* **2017**, *25*, 1461–1475. [CrossRef]
21. Garg, N.; Wazid, M.; Das, A.K.; Singh, D.P.; Rodrigues, J.J.; Park, Y. BAKMP-IoMT: Design of Blockchain Enabled Authenticated Key Management Protocol for Internet of Medical Things Deployment. *IEEE Access* **2020**, *8*, 95956–95977. [CrossRef]
22. Ali, Z.; Ghani, A.; Khan, I.; Chaudhry, S.A.; Islam, S.H.; Giri, D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *J. Inf. Secur. Appl.* **2020**, *52*, 102502. [CrossRef]
23. Liu, C.-H.; Chung, Y.-F. Secure user authentication scheme for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2017**, *59*, 250–261. [CrossRef]
24. Mbarek, B.; Ge, M.; Pitner, T. An Efficient Mutual Authentication Scheme for Internet of Things. *Internet Things* **2020**, *9*, 100160. [CrossRef]
25. Shen, J.; Gui, Z.; Ji, S.; Shen, J.; Tan, H.; Tang, Y. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *J. Netw. Comput. Appl.* **2018**, *106*, 117–123. [CrossRef]
26. Tewari, A.; Gupta, B. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. *J. Supercomput.* **2017**, *73*, 1085–1102. [CrossRef]
27. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
28. Kocher, P.C.; Jaffe, J.; Jun, B.; Rohatgi, P. Introduction to differential power analysis. *J. Cryptogr. Eng.* **2011**, *1*, 5–27. [CrossRef]
29. Lee, J.-Y.; Yu, S.-J.; Park, K.-S.; Park, Y.-H.; Park, Y.H. Secure Three-Factor Authentication Protocol for Multi-Gateway IoT Environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
30. Park, K.; Park, Y.; Das, A.K.; Yu, S.; Lee, J.; Park, Y. A Dynamic Privacy-Preserving Key Management Protocol for V2G in Social Internet of Things. *IEEE Access* **2019**, *7*, 76812–76832. [CrossRef]
31. Ren, Y.; Leng, Y.; Zhu, F.; Wang, J.; Kim, H.-J. Data Storage Mechanism Based on Blockchain with Privacy Protection in Wireless Body Area Network. *Sensors* **2019**, *19*, 2395. [CrossRef] [PubMed]
32. Wang, J.; Han, K.; Alexandridis, A.; Chen, Z.; Zilic, Z.; Pang, Y.; Jeon, G.; Piccialli, F. A blockchain-based eHealthcare system interoperating with WBANs. *Futur. Gener. Comput. Syst.* **2020**, *110*, 675–685. [CrossRef]
33. Wang, X.; Zeng, P.; Patterson, N.; Jiang, F.; Doss, R. An Improved Authentication Scheme for Internet of Vehicles Based on Blockchain Technology. *IEEE Access* **2019**, *7*, 45061–45072. [CrossRef]
34. Braeken, A. Highly Efficient Symmetric Key Based Authentication and Key Agreement Protocol Using Keccak. *Sensors* **2020**, *20*, 2160. [CrossRef]
35. Saeed, M.E.S.; Liu, Q.-Y.; Tian, G.; Gao, B.; Li, F. Remote Authentication Schemes for Wireless Body Area Networks Based on the Internet of Things. *IEEE Internet Things J.* **2018**, *5*, 4926–4944. [CrossRef]
36. Basin, D.; Cremers, C.; Dreier, J.; Sasse, R. Symbolically analyzing security protocols using tamarin. *ACM SIGLOG News* **2017**, *4*, 19–30. [CrossRef]
37. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Vasilakos, A.V. Secure Biometric-Based Authentication Scheme Using Chebyshev Chaotic Map for Multi-Server Environment. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 824–839. [CrossRef]
38. He, D.; Kumar, N.; Lee, J.-H.; Sherratt, R.S. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electron.* **2014**, *60*, 30–37. [CrossRef]