applied sciences

MDPI

Article

A Secure and Efficient Three-Factor Authentication Protocol in Global Mobility Networks

SungJin Yu [1], JoonYoung Lee [1], YoHan Park [2], YoungHo Park [1,*], SangWoo Lee [3] and BoHeung Chung [3]

1 School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; darkskiln@knu.ac.kr (S.Y.); harry250@naver.com (J.L.)
2 School of Computer Engineering, Keimyung University, Daegu 42601, Korea; yhpark@kmu.ac.kr
3 Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; ttomlee@etri.re.kr (S.L.); bhjung@etri.re.kr (B.C.)
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Received: 24 March 2020; Accepted: 19 May 2020; Published: 21 May 2020

Abstract: With the developments in communication and mobile technologies, mobile users can access roaming services by utilizing a mobile device at any time and any place in the global mobility networks. However, these require several security requirements, such as authentication and anonymity, because the information is transmitted over an open channel. Thus, secure and efficient authentication protocols are essential to provide secure roaming services for legitimate users. In 2018, Madhusudhan et al. presented a secure authentication protocol for global mobile networks. However, we demonstrated that their protocol could not prevent potential attacks, including masquerade, session key disclosure, and replay attacks. Thus, we proposed a secure and efficient three-factor authentication protocol to overcome the security weaknesses of Madhusudhan et al.'s scheme. The proposed scheme was demonstrated to prevent various attacks and provided a secure mutual authentication by utilizing biometrics and secret parameters. We evaluated the security of the proposed protocol using informal security analysis and formal security analysis, such as the real-or-random (ROR) model and Burrows–Abadi–Needham (BAN) logic. In addition, we showed that our scheme withstands man-in-the-middle (MITM) and replay attacks utilizing formal security validation automated validation of internet security protocols and applications (AVISPA) simulation. Finally, we compared the performance of our protocol with existing schemes. Consequently, our scheme ensured better security and efficiency features than existing schemes and can be suitable for resource-constrained mobile environments.

Keywords: authentication; global mobility networks; roaming service; BAN logic; ROR model; AVISPA simulation

1. Introduction

With the advances in wireless communication technology, the global mobility network (GLOMONET) [1–3] has become a popular means of communication. Users can access roaming services through mobile devices; therefore, people's access to knowledge has been improved significantly. In GLOMONET, each mobile user depends on a specific home agent (HA) where they are registered. If the mobile user is in the domain of a foreign agent (FA), the FA must ensure service after authenticating the mobile user. However, as a mobile device has limited resources available in terms of computing power, memory, and battery capacity [4,5], it is not suitable to apply symmetric and asymmetric cryptosystems that generate high computational overheads. In this case, mobile users can face delays during processing and service availing. In addition, a malicious adversary may attempt

various attacks using sensitive data transmitted via an insecure channel in GLOMONET. Therefore, secure and efficient mutual authentication has become an essential security requirement to provide secure roaming services for legitimate mobile users. The security requirements for GLOMONET are summarized as follows:

- Secure and efficient authentication schemes are required to provide various services in GLOMONET.
- Authentication schemes must resist various attacks, including stolen mobile devices, masquerades, and trace attacks.
- Authentication schemes must consider the limitations of mobile devices relative to the computing power, memory, and battery capacity [4,5].

In the last few years, many authentication schemes have been presented for GLOMONET to ensure the security of users [6–9]. In 2004, Zhu et al. [10] presented an efficient two-factor authentication scheme to provide the roaming facility. However, Lee et al. [11] indicated that Zhu et al.'s [10] protocol did not resist impersonation attacks and also could not achieve user authentication. In 2006, Lee et al. [11] presented an improved protocol for wireless environments to overcome the security flaws of Zhu et al.'s scheme. However, Wu et al. [12] assessed that Lee et al.'s [11] scheme did not withstand perfect backward secrecy and did not ensure user anonymity. In 2012, Li et al. [13] assessed that Wu et al.'s [12] scheme could not withstand replay and masquerade attacks and also could not provide user anonymity.

To overcome these security flaws, Li et al. [13] then proposed a novel user authentication scheme based smart-card to provide efficient high computational and communication overheads. However, Das et al. [14] demonstrated that Li et al.'s protocol [13] was sensitive to replay attacks and did not achieve proper user password updates in the password change processes. In 2015, Marimuthu and Saravanan [15] presented a secure authentication protocol in GLOMONET. However, Madhusudhan et al. [16] proved that their protocol could not withstand offline guessing, insider, stolen-verifier, denial of service, and forgery attacks.

In 2018, Madhusudhan et al. [16] presented a secure and efficient user authentication scheme for GLOMONET using a mobile device to resolve the security problems of Marimuthu and Saravanan's scheme. Madhusudhan et al. claimed that their scheme could prevent replay and masquerade attacks and provide secure mutual authentication. Unfortunately, we analyzed that Madhusudhan et al.'s scheme [16] could not prevent various security threats and could not provide secure mutual authentication. Moreover, Madhusudhan et al.'s scheme [16] was unsuitable for resource-constrained mobile devices as it uses symmetric key encryption and modular multiplication, which generate high computational overheads. Thus, we proposed a secure and efficient three-factor user authentication scheme for roaming services in GLOMONET to resolve the security flaws of Madhusudhan et al.'s scheme.

1.1. Motivation and Contributions

We have studied numerous user authentication schemes [6,8,15,16] for roaming services and found that they had the following in common:

1. Many authentication protocols [6,8,15,16] are exposed to well-known attacks, such as masquerade, replay, mobile device theft, and session key disclosure attacks in global mobility environments.
2. Many authentication schemes must provide secure convenience for mobile users in the GLOMONET and must take into account all the security requirements specified in Section 1.2.
3. Secure and lightweight authentication schemes are essential, which take into account limitations for resource-constrained mobile devices relative to computing power, memory, and battery capacity.

Recently, Madhusudhan et al. [16] presented a secure and efficient user authentication scheme for GLOMONET using a mobile device. They claimed that their scheme could resist various attacks

and could ensure secure mutual authentication and anonymity. However, our paper presents a brief review of Madhusudhan et al.'s scheme [16], and we demonstrated that their scheme could not prevent various security threats. To resolve the security threats of Madhusudhan et al.'s scheme, we proposed a secure and efficient three-factor authentication protocol. The proposed scheme demonstrated several advantages compared with previous related authentication schemes.

First, the proposed scheme could prevent various attacks, such as mobile device theft, masquerade, session key disclosure, and replay attacks and also provided secure mutual authentication, user anonymity, and user friendliness. Second, the proposed scheme used the fuzzy extractor mechanism to improve the security level of the protocol. Even if two of the three factors were compromised, the proposed scheme was still secure. Finally, the proposed scheme provided better effective computation costs with related schemes as it only utilized the one-way hash function. Therefore, the proposed scheme was secure, efficient, and more suitable for practical mobile and wireless environments.

1.2. Security Requirements

The research on the security of communication for GLOMONET has shown that the security requirements are essential to produce a secure and efficient authentication protocol. Table 1 shows the security requirements for authenticaiton and key agreement protocol.

Table 1. Security requirements for authentication and key agreement protocols.

Properties	Description
Three-factor security	This should remain secure even if any two of the three factors are compromised.
Resisting known attacks	This requires that the authentication protocol for GLOMONET is secure from various known attacks, including privileged insider, replay, session key disclosure, MITM, and masquerade attacks.
Resisting stolen mobile device attack	If an unauthorized person obtains the lost/stolen mobile device, it is impossible for him to impersonate a valid user with a counterfeit login request by using the information extracted from the mobile device.
Forward and backward secrecy	This requires that the attacker is not able to obtain the previous session keys or future ones by using the compromised session key.
Secure mutual authentication and key agreement	This is an essential requirement in the GLOMONET scenario, and requires the communication parties to be able to authenticate each other and generate a shared session key to provide confidentiality of messages in public channels.
User friendliness	The mobile user should freely select his/her own identity and password. In addition, the mobile user should be allowed to update the password without the assistance of the home agent.
Anonymity and untraceability	A malicious attacker is incapable of revealing and tracking the real identity of the legitimate user, and this is an important privacy-preserving requirement for users.

1.3. Organization

The remainder of this paper is organized as follows. In Section 2, we present the preliminaries, and in Section 3, we review Madhusudhan et al.'s scheme [16]. In Sections 4 and 5, we assess the security flaws of Madhusudhan et al.'s scheme [16] and present a secure and efficient authentication scheme for GLOMONET to overcome the security flaws of Madhusudhan et al.'s scheme [16]. In Section 6, we demonstrate the security of our scheme using informal security analysis and formal security analysis, including Burrows–Abadi–Needham (BAN) logic and the real-or-random (ROR) model. In Section 7,

we report a formal security validation utilizing the Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool. In Section 8, we compare the performance properties of our protocol to existing protocols. We present our conclusions in the final Section 9.

2. Preliminaries

This section presents preliminaries to facilitate reader comprehension.

2.1. Attacker Model

To examine the security of our protocol, we describe the Dolev–Yao (DY) model [17], which is described as follows:

- An adversary is able to eavesdrop, intercept, modify, delete, or insert messages exchanged through an open channel.
- An adversary is able to obtain the lost or stolen mobile device of legitimate mobile users [18,19] and can extract the important data stored in the mobile device by utilizing a power-analysis attack [20,21].
- An adversary is able to perform various types of attacks, including replay, masquerade, man-in-the-middle (MITM) and mobile device theft attacks.

2.2. Fuzzy Extractors

This section discusses the basic concepts of a fuzzy extractor. According to [22], this mechanism involves two procedures, such as *Gen* and *Rep*. The detailed description for *Gen* and *Rep* are below:

1. *Gen*: After a user imprints the biometric input *Bio*, the probabilistic function *Gen* selects a consistent random string $\rho \in \{0,1\}^l$ and a random auxiliary string $\sigma \in \{0,1\}^*$.
2. *Rep* : After a new user imprints the biometric input Bio_{new} and the string value σ in a session, *Rep* successfully recovers the value ρ.

3. Review of Madhusudhan et al.'s Protocol

Madhusudhan et al.'s scheme [16] is comprised of three processes: (1) user registration, (2) authentication, and (3) password update. The notations utilized in this paper are defined in Table 2 and each process is detailed as follows.

Table 2. Notations.

Notation	Description
ID_{MU}	MU's identity
ID_{FA}	FA's identity
ID_{HA}	HA's identity
R_S	HA's random number
R_{MU}, R_{FA}, R_{HA}	Random nonce of MU, FA, and HA
PW_{MU}	MU's password
BIO_i	MU's biometrics
K_S	HA's master key
SK_i	Session key between MU and FA
K_{FH}	Shared secret key between FA and HA
$(X)_K$	Symmetric encryption/decryption
T	Timestamp
$h(\cdot)$	Hash function
$\oplus$	Bitwise XOR operation
$\|\|$	Concatenation operation

3.1. Initialization Process

The home agent (HA) selects two prime numbers p, q and generator g of a finite field in Z_p^*, of which Z_p^* is a nonsingular elliptic curve $y^2 = x^3 + ax + b \pmod p$. The HA calculates $n = p \times q$ and $\phi(n) = (p-1) \times (q-1)$. Then, the HA chooses an integer e, such that $1 < e < \phi(n)$ and $gcd(e, \phi(n)) = 1$. After that, the HA computes the value of an integer d, such that $d = e^{-1}$, where d is the secret key of the HA, and $y = g^d \bmod n$, where y is the public key of the HA. The HA keeps $\{p, q, d\}$ securely.

3.2. Registration Process

In Madhusudhan et al.'s protocol, a new MU who requests roaming services must register their identity with the HA. Figure 1 indicates the user registration process of Madhusudhan et al.'s protocol [16] and this process is described in detail as follows.

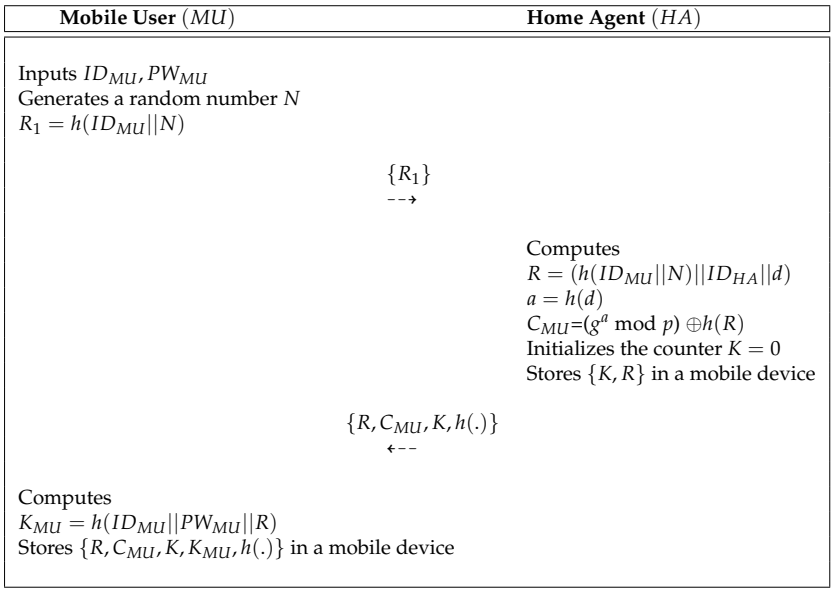

Figure 1. The user registration process of Madhusudhan et al.'s protocol.

Step 1: A mobile user MU inputs $ID_{MU} and PW_{MU}$ and selects a random number N. Then, MU computes $R_1 = h(ID_{MU}||N)$ and sends a request message to the HA via a public channel.

Step 2: After obtaining messages $\{R_1\}$, the HA calculates $R = (h(ID_{MU}||N)||ID_{HA}||d)$, $a = h(d)$ and C_{MU} = (g^a mod p) $\oplus h(R)$. After that, HA sets the value of the counter $K = 0$ and stores $\{K, R\}$ in a secure database. Then, HA sends $\{R, C_{MU}, K, h(.)\}$ to MU over a secure channel.

Step 3: After obtaining messages $\{R, C_{MU}, K, h(.)\}$, the MU computes $K_{MU} = h(ID_{MU}||PW_{MU}||R)$ and stores it in a mobile device. Finally, the mobile device of the MU contains $\{R, C_{MU}, K, K_{MU}, h(.)\}$.

3.3. Login and Authentication Process

In Madhusudhan et al.'s protocol [16], they considered a scenario in which the MU associated with the HA visits a foreign network from the foreign agent FA and attempts to access the roaming service. A MU who requests roaming service must send a login request message to the HA. The MU, FA, and HA then perform mutual authentication with each other, then MU and FA share the session key. Figure 2 indicates the login and authentication process of Madhusudhan et al.'s protocol [16]. The process is described in detail as follows.

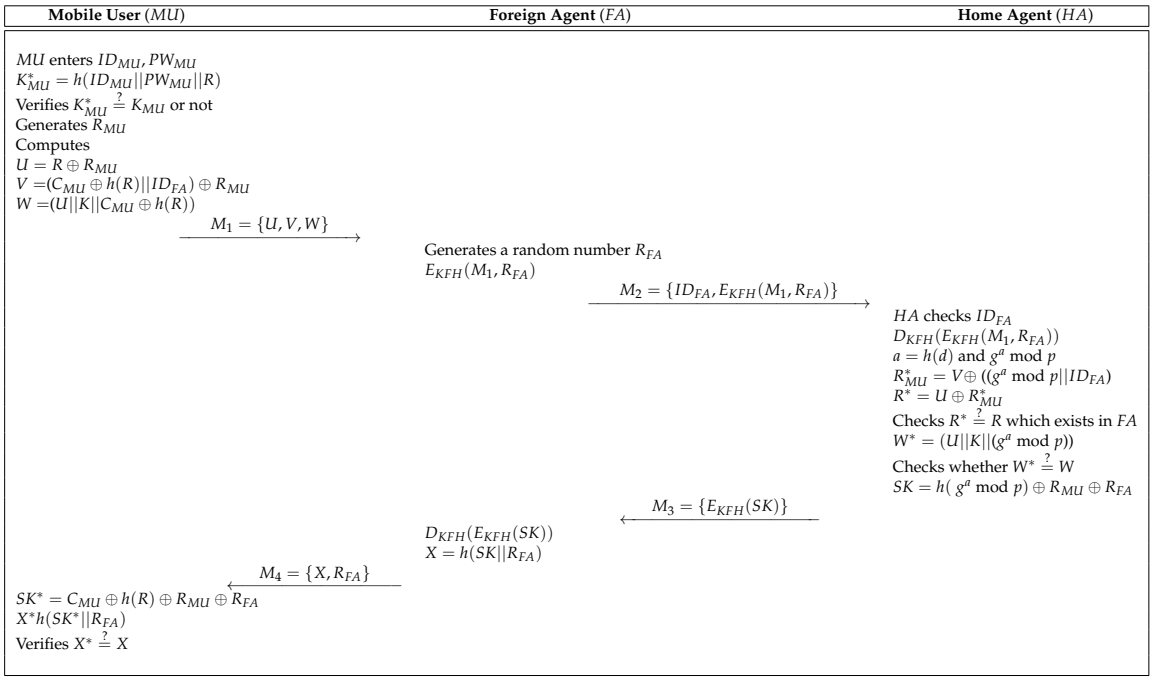

Figure 2. The login and authentication process of Madhusudhan et al.'s protocol.

Step 1: The MU retrieves the authentication data stored in the mobile device and enters ID_{MU} and PW_{MU}. After that, the mobile device computes $K^*_{MU} = h(ID_{MU}||PW_{MU}||R)$ and checks whether $K^*_{MU} \stackrel{?}{=} K_{MU}$. If this condition holds, the MU generates a random number R_{MU} and computes $U = R \oplus R_{MU}$, $V = (C_{MU} \oplus h(R)||ID_{FA}) \oplus R_{MU}$ and $W = (U||K||C_{MU} \oplus h(R))$. Then, MU sends the login request message $M_1 = \{U, V, W\}$ to FA through an insecure channel.

Step 2: After obtaining the $M_1 = \{U, V, W\}$, FA selects a random number R_{FA} and encrypts M_1 and R_{FA} using the shared secret key. The FA sends the $M_2 = \{ID_{FA}, E_{KFH}(M_1, R_{FA})\}$ to the HA.

Step 3: Upon reception of $M_2 = \{ID_{FA}, E_{KFH}(M_1, R_{FA})\}$, the HA checks the identity ID_{FA} of FA and retrieves the secret key corresponding to ID_{FA}. After that, the HA decrypts $D_{KFH}(E_{KFH}(M_1, R_{FA}))$ and computes $a = h(d)$, $g^a \bmod p$, $R^*_{MU} = V \oplus ((g^a \bmod p)||ID_{FA})$ and $R^* = U \oplus R^*_{MU}$. The HA then checks whether there exists $R^* \stackrel{?}{=} R$ in a secure database. If the condition is valid, the HA computes $W^*=(U||K||(g^a \bmod p))$ and checks whether $W^* \stackrel{?}{=} W$. If it is correct, the HA computes $SK=h(g^a \bmod p) \oplus R_{MU} \oplus R_{FA}$ and sends $M_3 = \{E_{KFH}(SK)\}$ to the FA.

Step 4: After obtaining the $M_3 = \{E_{KFH}(SK)\}$, the FA decrypts $D_{KFH}(E_{KFH}(SK))$ and computes $X = h(SK||R_{FA})$. Finally, the FA sends $M_4 = \{X, R_{FA}\}$ to the MU.

Step 5: Upon reception of $M_4 = \{X, R_{FA}\}$, the MU computes $SK^* = C_{MU} \oplus h(R) \oplus R_{MU} \oplus R_{FA}$ and $X^* = h(SK^*||R_{FA})$. After that, the MU checks whether $X^* \stackrel{?}{=} X$. If this holds, the MU and FA achieve the SK successfully.

3.4. Password Update Process

In Madhusudhan et al.'s protocol, the MU can freely update their password. The process is described in detail as follows.

Step 1: When a legitimate MU wants to update the password, the MU inputs ID_{MU}, PW_{MU} and the request messages are transmitted via a terminal.

Step 2: The mobile device of MU calculates $K^*_{MU} = h(ID_{MU}||PW_{MU})$ and checks whether $K^*_{MU} \stackrel{?}{=} K_{MU}$. If this holds, the MU is legitimate user. Otherwise, the mobile device terminates the password change process.

Step 3: The MU selects new password PW^{NEW}_{MU} and computes $K^{NEW}_{MU} = h(ID_{MU}||PW^{NEW}_{MU})$. Finally, the mobile device of MU replaces $\{K_{MU}\}$ with $\{K^{NEW}_{MU}\}$.

4. Cryptanalysis of Madhusudhan et al.'s Protocol

We demonstrated the security shortcomings of the existing protocol [16]. They claimed that their scheme can resist replay and masquerade attacks and achieve secure user authentication. However, we demonstrated that Madhusudhan et al.'s protocol [16] is insecure against various attacks, including session key disclosure, replay, and masquerade attacks. Furthermore, we show that the existing protocol [16] does not provide mutual authentication.

4.1. Masquerade Attack

If a malicious adversary MU_a can attempt to impersonate a legitimate user, MU_a can easily generate the message $M_1 = \{U, V, W\}$ of the legitimate user. As discussed in Section 2.1, MU_a obtains the mobile device of MU and extracts the stored secret parameters in it. In addition, MU_a intercepts the message exchanged over a public channel. Finally, MU_a performs the masquerade attack and its detailed procedures.

Step 1: A MU_a calculates $R_{MU} = U \oplus R$, $V = (C_{MU} \oplus h(R)||ID_{FA}) \oplus R_{MU}$ and $W = (U||K||C_{MU} \oplus h(R))$. Then, MU_a generates a random number R_a. After that, MU_a computes $U_a = R \oplus R_a$, $V_a = (C_{MU} \oplus h(R)||ID_{FA}) \oplus R_a$ and $W_a = (U_a||K||C_{MU} \oplus h(R))$ and sends $M_{1a} = \{U_a, V_a, W_a\}$ to the FA.

Step 2: After obtaining the $M_{1a} = \{U_a, V_a, W_a\}$, the FA selects a random number R_{FA} and encrypts $E_{KFH}(M_1, R_{FA})$ using a shared secret key. Then, the FA sends $M_2 = \{ID_{FA}, E_{KFH}(M_{1a}, R_{FA})\}$ to the HA.

Step 3: Upon reception of $M_2 = \{ID_{FA}, E_{KFH}(M_{1a}, R_{FA})\}$, the HA decrypts $D_{KFH}(E_{KFH}(M_1, R_{FA}))$ and computes $a = h(d)$, $g^a \bmod p$, $R^*_a = V_a \oplus ((g^a \bmod p||ID_{FA})$ and $R^* = U_a \oplus R^*_a$. Then, the HA checks whether $R^* \stackrel{?}{=} R$. After that, HA computes W^*_a=$(U_a||K||\ (g^a \bmod p))$ and checks whether $W^* \stackrel{?}{=} W$. Finally, HA computes $SK = h(g^a \bmod p) \oplus R_a \oplus R_{FA}$ and sends $M_3 = \{E_{KFH}(SK)\}$ to the FA.

Step 4: After obtaining the $M_3 = \{E_{KFH}(SK)\}$, the FA decrypts $D_{KFH}(E_{KFH}(SK))$ and computes $X = h(SK||R_{FA})$, then sends $M_4 = \{X, R_{FA}\}$ to the MU_a.

Step 5: Upon reception of $M_4 = \{X, R_{FA}\}$, the MU_a computes the $SK^* = C_{MU} \oplus h(R) \oplus R_a \oplus R_{FA}$, $X^* = h(SK^*||R_{FA})$ and checks whether $X^* \stackrel{?}{=} X$. If it is correct, MU_a computes the SK

MU_a obtains the session key between MU_a and FA and performs mutual authentication successfully. As a result, Madhusudhan et al.'s protocol [16] is insecure against the masquerade attacks.

4.2. Replay Attack

Madhusudhan et al. claimed that their protocol can withstand replay attacks because a MU_a cannot calculate the correct $SK = h(g^a \bmod p) \oplus R_{MU} \oplus R_{FA}$ without the random number R_{FA} and R_{MU}. However, according to Section 4.1, MU_a computes $R_{MU} = U \oplus R$ and obtains R_{FA} in an open channel. Furthermore, MU_a can extract the secret parameter $\{C_{MU}, R\}$ stored in the mobile device. MU_a computes $SK = C_{MU} \oplus h(R) \oplus R_{MU} \oplus R_{FA}$. In addition, according to Section 2.1, MU_a can obtain the counter value K in the mobile device. Thus, Madhusudhan et al.'s protocol [16] is insecure against replay attacks.

4.3. Session Key Disclosure Attack

According to Section 4.1, a MU_a can successfully impersonate a legitimate mobile user MU and calculate the SK. According to the discussion presented in Section 2.1, MU_a can extract the $\{C_{MU}, R\}$ in the mobile device and obtain random number R_{FA} of FA over an open channel, and then compute $R_{MU} = U \oplus R$. Therefore, MU_a can compute $SK = C_{MU} \oplus h(R) \oplus R_{MU} \oplus R_{FA}$. Therefore, Madhusudhan et al.'s protocol [16] is insecure against session key disclosure attacks.

4.4. Mutual Authentication

In the existing protocol [16], they indicated that their scheme preserves secure mutual authentication among the MU, FA, and HA. However, according to Section 4.1, their protocol cannot prevent masquerade attacks and the MU_a can successfully calculate authentication request message $W = (U||K||C_{MU} \oplus h(R))$ and authentication message $X^* = h(SK^*||R_{FA})$. Consequently, Madhusudhan et al.'s protocol [16] cannot achieve mutual authentication.

5. Proposed Secure and Efficient Authentication Protocol for GLOMONET

Many biometric-based user authentication protocols [23,24] have been presented to improve the security flaws associated with mobile device authentication. Biometric-based schemes are difficult to guess, duplicate, and forge and cannot be stolen or lost. Therefore, biometric-based three-factor authentication mechanisms are more secure than mobile device and password based two-factor authentication mechanisms. Therefore, we present a secure and efficient authentication protocol using biometrics to overcome the security problems of the existing protocol [16].

5.1. Registration Process

A new MU should register with HA to receive the roaming services. Figure 3 presents the user registration process of our protocol.

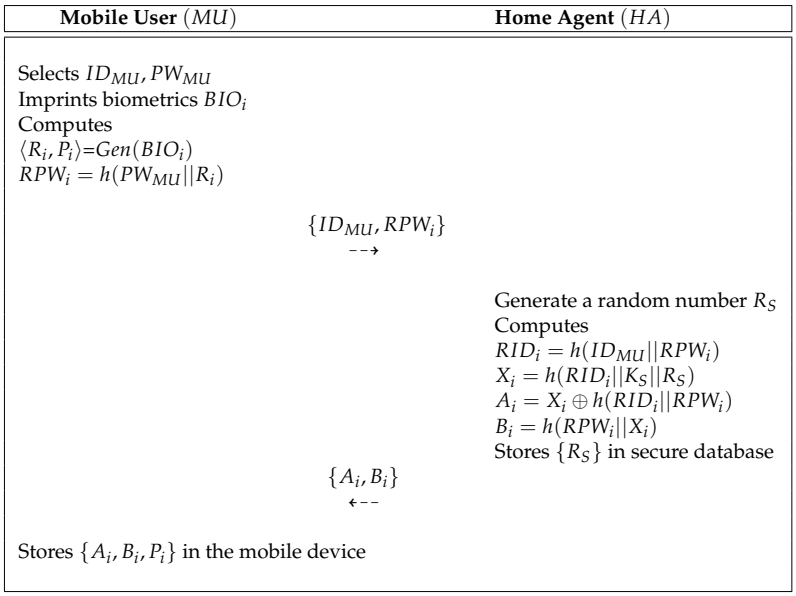

Figure 3. The user registration process of the proposed protocol.

Step 1: A MU selects ID_{MU}, PW_{MU} and imprints biometric BIO_i. After that, MU computes $\langle R_i, P_i \rangle = Gen(BIO_i)$, $RPW_i = h(PW_{MU}||R_i)$ and sends $\{ID_{MU}, RPW_i\}$ to the HA over a secure communication.

Step 2: After obtaining messages $\{ID_{MU}, RPW_i\}$, the HA computes $RID_i = h(ID_{MU}||RPW_i)$, $X_i = h(RID_i||K_S||R_S)$, $A_i = X_i \oplus h(RID_i||RPW_i)$ and $B_i = h(RPW_i||X_i)$. After that,

HA stores $\{R_S\}$ in a secure database. Finally, the HA sends $\{A_i, B_i\}$ to the MU via a secure communication.

Step 3: Upon reception of $\{A_i, B_i\}$, the MU stores $\{A_i, B_i, P_i\}$ in the mobile device.

5.2. Login and Authentication Process

Before performing a session, the MU requests authentication to the HA in order to establish the session key. Figure 4 presents the user authentication process of our protocol. The process is described in detail as follow.

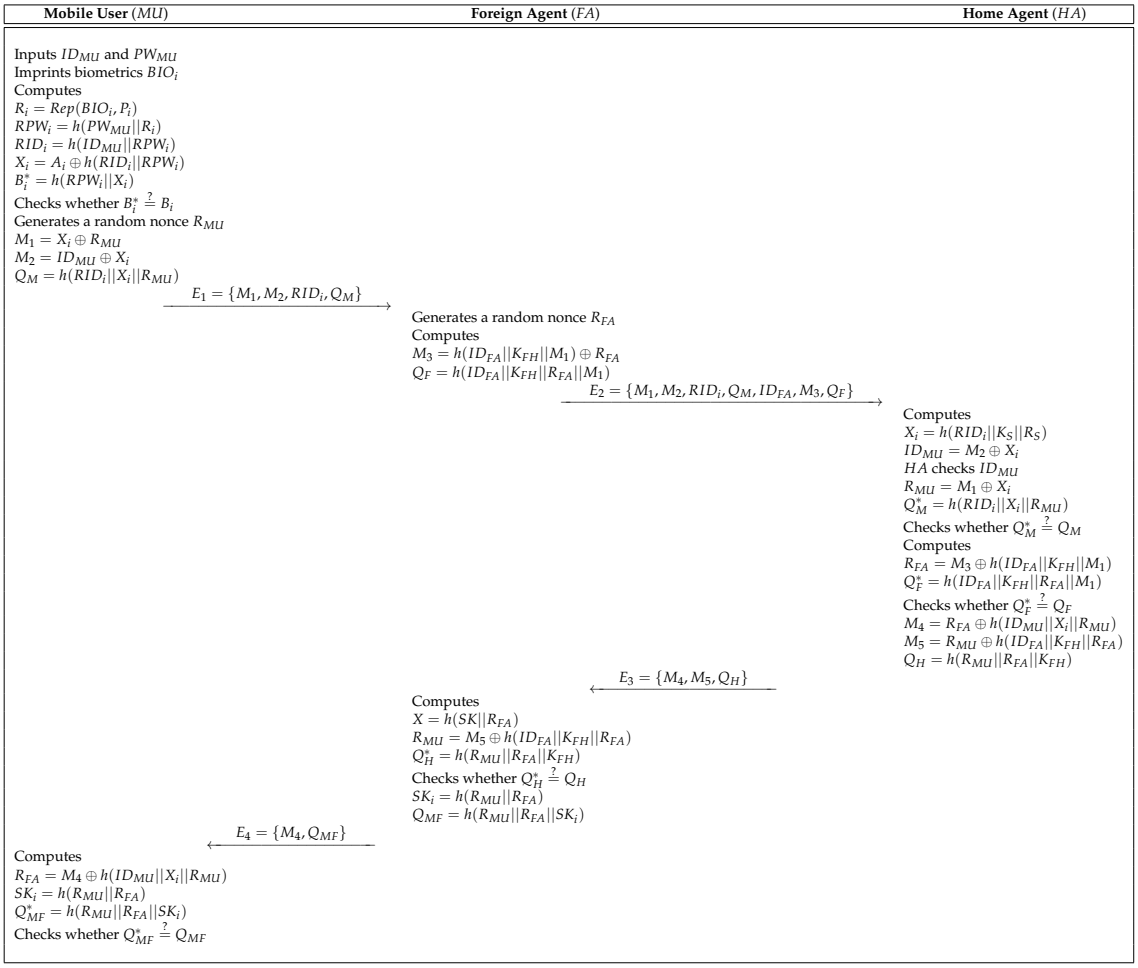

Figure 4. Login and authentication process of the proposed protocol.

Step 1: The mobile device inputs ID_{MU}, PW_{MU} and imprints biometrics BIO_i. The MU computes $R_i = Rep(BIO_i, P_i)$, $RPW_i = h(PW_{MU}||R_i)$, $RID_i = h(ID_{MU}||RPW_i)$, $X_i = A_i \oplus h(RID_i||RPW_i)$, and $B_i^* = h(RPW_i||X_i)$ and checks whether $B^* \stackrel{?}{=} B_i$. If this holds, the MU generates a random nonce R_{MU} and computes $M_1 = X_i \oplus R_{MU}$, $M_2 = ID_{MU} \oplus X_i$ and $Q_M = h(RID_i||X_i||R_{MU})$. After that, MU sends $\{E_1\}$ to the FA over an open channel.

Step 2: Upon reception of E_1, the FA selects a random nonce R_{FA} and computes $M_3 = h(ID_{FA}||K_{FH}||M_1) \oplus R_{FA}$, $Q_F = h(ID_{FA}||K_{FH}||R_{FA}||M_1)$. After that, the FA sends $\{E_2\}$ to the HA.

Step 3: Upon reception of E_2, the HA computes $X_i = h(RID_i||K_S||R_S)$, $ID_{MU} = M_2 \oplus X_i$ and checks the identity ID_{MU} of the mobile user. Then, HA computes $R_{MU} = M_1 \oplus X_i$, $Q_M^* = h(RID_i||X_i||R_{MU})$ and checks whether $Q_M^* \stackrel{?}{=} Q_M$. If it is valid, the HA calculates $R_{FA} = M_3 \oplus h(ID_{FA}||K_{FH}||M_1)$, $Q_F^* = h(ID_{FA}||K_{FH}||R_{FA}||M_1)$ and checks whether $Q_F^* \stackrel{?}{=} Q_F$. Then,

the HA computes $M_4 = R_{FA} \oplus h(RID_i||X_i||R_{MU})$, $M_5 = R_{MU} \oplus h(ID_{FA}||K_{FH}||R_{FA})$ and $Q_H = h(R_{MU}||R_{FA}||K_{FH})$. Finally, the HA sends an authentication message $\{E_3\}$ to the FA.

Step 4: Upon reception of E_3, the FA computes $R_{MU} = M_5 \oplus h(ID_{FA}||K_{FH}||R_{FA})$, $Q_H^* = h(R_{MU}||R_{FA}||K_{FH})$ and checks whether $Q_H^* \stackrel{?}{=} Q_H$. If it is correct, the FA computes SK_i=$h(R_{MU}||R_{FA})$, $Q_{MF} = h(R_{MU}||R_{FA}||SK_i)$ and sends $\{E_4\}$ to the MU.

Step 5: Upon reception of E_4, the MU calculates $R_{FA} = M_4 \oplus h(RID_i||X_i||R_{MU})$, $SK_i = h(R_{MU}||R_{FA})$, and $Q_{MF}^* = h(R_{MU}||R_{FA}||SK_i)$. Finally, the MU checks whether $Q_{MF}^* \stackrel{?}{=} Q_{MF}$. If it holds, the MU and FA establish the SK_i successfully.

5.3. Password Update Process

In the proposed protocol, a MU can easily update their password. Figure 5 presents the password change process of the proposed protocol.

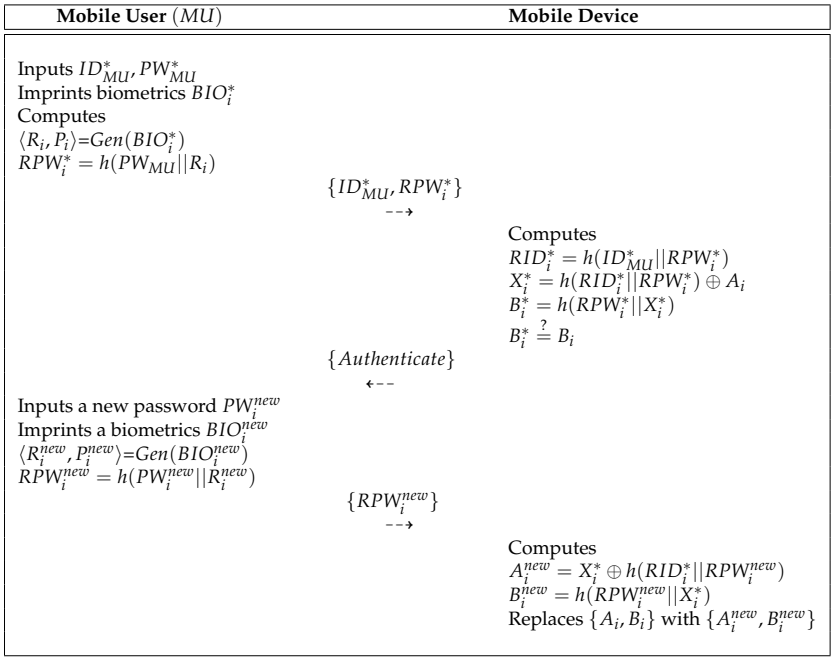

Figure 5. Password change process of the proposed protocol.

Step 1: The MU inputs ID_{MU}^*, PW_{MU}^* and imprints biometrics BIO_i^*. After that, MU computes $\langle R_i, P_i \rangle$=$Gen(BIO_i^*)$, $RPW_i^* = h(PW_{MU}||R_i)$ and sends $\{ID_{MU}^*, RPW_i^*\}$ to the mobile device.

Step 2: Upon reception of $\{ID_{MU}^*, RPW_i^*\}$, the mobile device computes $RID_i^* = h(ID_{MU}^*||RPW_i^*)$, $X_i^* = h(RID_i^*||RPW_i^*) \oplus A_i$, $B_i^* = h(RPW_i^*||X_i^*)$, and the mobile device checks whether $B_i^* \stackrel{?}{=} B_i$. If it is correct, the mobile device sends the authentication response message to the MU.

Step 3: Upon reception of the authentication response message, the MU inputs a new password PW_i^{new} and imprints a new biometrics BIO_i^{new}. MU computes $\langle R_i^{new}, P_i^{new} \rangle$=$Gen(BIO_i^{new})$, $RPW_i^{new} = h(PW_i^{new}||R_i^{new})$ and sends $\{RPW_i^{new}\}$ to the mobile device.

Step 4: Upon reception of $\{RPW_i^{new}\}$, the mobile device computes $A_i^{new} = X_i^* \oplus h(RID_i^*||RPW_i^{new})$, $B_i^{new} = h(RPW_i^{new}||X_i^*)$ and replaces $\{A_i, B_i\}$ with $\{A_i^{new}, B_i^{new}\}$.

6. Security Analysis

We utilized the BAN logic to evaluate the user authentication of our protocol and then we used the ROR model to prove the session key security. In addition, we performed AVISPA simulation to evaluate the security of our protocol to replay and MITM attacks.

6.1. Informal Security Analysis

This section presents an informal security analysis to evaluate the security of the proposed protocol. We proved that our scheme can prevent various attacks and allow user authentication and anonymity.

6.1.1. Masquerade Attack

If a MU_a attempts to impersonate a legal mobile user, MU_a must calculate a request message $\{M_1, M_2, RID_i, Q_M\}$ and response message $\{M_4, Q_{MF}\}$ successfully. However, MU_a cannot compute this because MU_a does not know MU's real identity ID_{MU}, password PW_{MU}, secret parameters X_i, random nonce R_{MU}, and biometrics BIO_i. Consequently, the proposed protocol can withstand masquerade attacks because MU_a cannot generate correct messages successfully.

6.1.2. Replay Attack

Our protocol can resist replay attacks utilizing random nonce that is changed every session. If a MU_a may try to impersonate a mobile user by resending messages that were exchanged in a previous session, MU_a cannot obtain the previous messages because the HA checks whether $R^*_{MU} \stackrel{?}{=} R_{MU}$ and $R^*_{FA} \stackrel{?}{=} R_{FA}$. Consequently, the proposed protocol can withstand replay attacks because MU_a does not know R_{MU} and R_{FA}.

6.1.3. Stolen Mobile Device Attack

We assume that a MU_a can steal the mobile device of a legitimate user and extract the data $\{A_i, B_i, P_i\}$ from the mobile device by utilizing a power analysis attack [20]. However, MU_a still cannot obtain a legitimate user's information because the parameters stored in the mobile device are masked using bitwise XOR operations and hash functions. Thus, the proposed scheme can defend against mobile device theft attacks.

6.1.4. Session Key Disclosure Attack

In our protocol, a MU_a cannot compute $\{M_1, M_2, Q_M\}$ because a legitimate mobile user MU generates an authentication request message by using the dynamic random nonce R_{MU} and secret parameter X_i. Consequently, the proposed protocol protects against session key disclosure attacks.

6.1.5. Anonymity

In our protocol, a MU_a cannot obtain the identity ID_{MU} of a legitimate mobile user because the parameters are masked by using XOR operations and hash functions, such as $M_2 = ID_{MU} \oplus X_i$ and $Q_M = h(RID_i||X_i||R_{MU})$. Consequently, our protocol provides anonymity because a MU_a cannot obtain ID_{MU} without X_i and R_{MU}.

6.1.6. Mutual Authentication

After obtaining the login request messages $\{M_1, M_2, RID_i, Q_M\}$ from MU, the HA checks whether $Q^*_M \stackrel{?}{=} Q_M$. If this holds, HA authenticates MU. After obtaining the messages $\{M_3, Q_F\}$ from FA, the HA checks whether $Q^*_F \stackrel{?}{=} Q_F$. If it is valid, HA authenticates FA. After obtaining the messages $\{M_4, M_5, Q_H\}$ from HA, FA checks whether $Q^*_H \stackrel{?}{=} Q_H$. If this holds, FA authenticates HA. Finally, MU checks whether $Q^*_{MF} \stackrel{?}{=} Q_{MF}$. If this holds, MU authenticates HA. Consequently, our protocol ensures secure mutual authentication among MU, FA and HA because a MU_a does not know the secret parameter of MU and FA.

6.1.7. User Friendliness

In our protocol, MU can easily change his/her own ID_i and PW_i without the assistance of the HA. In particular, the proposed protocol allows the MU to change the original password PW_i in a short time. Because, the MU need not go through the entire login process, which saves the time as well as minimizes the computation complexity of the proposed scheme. Consequently, the proposed protocol is user-friendly.

6.2. Security Properties

Table 3 presents the better security properties ensured by the proposed scheme compared to related schemes [6,8,15,16]. The existing schemes are insecure various attacks and their scheme cannot ensure mutual authentication and user anonymity. In contrast, the proposed scheme can provide essential security properties and can achieve user anonymity and mutual authentication.

Table 3. Security features compared to existing schemes.

Security features	He et al. [6]	Kuo et al. [8]	Karuppiah et al. [15]	Madhusudhan et al. [16]	Ours
User anonymity	×	×	○	○	○
User friendliness	○	○	○	○	○
Mutual authentication	×	○	○	×	○
Insider attack	○	○	○	○	○
Replay attack	○	×	○	×	○
Perfect forward secrecy attack	○	○	○	○	○
Session key disclosure attack	×	×	○	×	○
Masquerade attack	×	○	○	×	○

○: it supports the security feature; ×: it does not support the security feature.

6.3. Authentication Proof Using BAN Logic

We present the security analysis utilizing the BAN logic [25] to prove the secure user authentication of our protocol. In Table 4, we present the notations used for BAN logic. We present the security rules, the security goals, the idealized forms and the assumptions that are essential to BAN logic. We assessed that our scheme ensured mutual authentication among MU, FA, and HA.

Table 4. Notations used for BAN logic.

Notation	Description
$A\mid\equiv B$	A **believes** that B
$\#B$	B is updated and **fresh**
$A \lhd B$	A **sees** that B
$A\mid\sim B$	A once **said** B
$A \Rightarrow B$	A **controls** that B
$< B >_W$	B is **combined** with W
$\{B\}_K$	B is **encrypted** utilizing symmetric key K
$A \overset{K}{\leftrightarrow} P$	A and P can make secure contact utilizing K as the **shared secret key**
SK	Session key used in communication session

6.3.1. Rules of BAN Logic

The rules of BAN logic are summarized as follows.

1. Message meaning rule :

$$\frac{A \mid \equiv A \overset{K}{\leftrightarrow} P, \quad A \lhd \{B\}_K}{A \mid\equiv P \mid \sim B}$$

2. Nonce verification rule :

$$\frac{A \mid\equiv \#(B), \quad A \mid \equiv P \mid \sim B}{A \mid\equiv P \mid \equiv B}$$

3. Jurisdiction rule :

$$\frac{A \mid\equiv P \mid \Longrightarrow B, \quad A \mid\equiv P \mid \equiv B}{A \mid \equiv B}$$

4. Freshness rule :

$$\frac{A \mid \equiv \#(B)}{A \mid \equiv \#(B, W)}$$

5. Belief rule :

$$\frac{A \mid \equiv (B, W)}{A \mid \equiv B}$$

.

6.3.2. Goals

To analyze mutual authentication, we define the goals of our protocol as below.

Goal 1: $MU \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA)$
Goal 2: $FA \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA)$
Goal 3: $MU \mid\equiv FA \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA)$
Goal 4: $FA \mid\equiv MU \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA)$.

6.3.3. Idealized Forms

The idealized form of messages of our protocol are as below.

Msg_1**:** $MU \rightarrow FA$: $(RID_i, ID_{MU}, R_{MU})_{X_i}$
Msg_2**:** $FA \rightarrow HA$: $(RID_i, ID_{MU}, R_{MU}, X_i, R_{FA}, ID_{FA})_{K_{FH}}$
Msg_3**:** $HA \rightarrow FA$: $(ID_{MU}, ID_{FA}, R_{FA}, R_{MU})_{K_{FH}}$
Msg_4**:** $FA \rightarrow MU$: $(ID_{MU}, R_{MU}, R_{FA}, (MU \overset{SK}{\longleftrightarrow} FA))_{X_i}$.

6.3.4. Assumptions

The following assumptions are applied in the BAN logic analysis.

A_1**:** $FA \mid\equiv (MU \overset{X_i}{\longleftrightarrow} FA)$
A_2**:** $FA \mid\equiv \#(R_{MU})$
A_3**:** $HA \mid\equiv (HA \overset{K_{FH}}{\longleftrightarrow} FA)$
A_4**:** $HA \mid\equiv \#(R_{FA})$
A_5**:** $FA \mid\equiv (HA \overset{K_{FH}}{\longleftrightarrow} FA)$
A_6**:** $FA \mid\equiv \#(R_{FA})$
A_7**:** $MU \mid\equiv (MU \overset{X_i}{\longleftrightarrow} FA)$
A_8**:** $MU \mid\equiv \#(R_{FA})$
A_9**:** $MU \mid\equiv FA \Rightarrow (MU \overset{SK}{\longleftrightarrow} FA)$
A_{10}**:** $FA \mid\equiv MU \Rightarrow (MU \overset{SK}{\longleftrightarrow} FA)$.

6.3.5. Proof Using BAN Logic

The proof then proceeds as below:

Step 1: According to Msg_1, we obtain the following

$$(S_1) : FA \lhd (RID_i, ID_{MU}, R_{MU})_{X_i}.$$

Step 2: Utilizing S_1 and A_1 with the "message meaning rule", the following is obtained

$$(S_2) : FA \mid\equiv MU \mid\sim (RID_i, ID_{MU}, R_{MU})_{X_i}.$$

Step 3: Now, using S_2 and A_2 with the "freshness rule", the following is obtained

$$(S_3) : FA \mid\equiv \#(RID_i, ID_{MU}, R_{MU})_{X_i}.$$

Step 4: Utilizing S_2 and S_3 with the "nonce verification rule", we obtain

$$(S_4) : FA \mid\equiv MU \mid\equiv (RID_i, ID_{MU}, R_{MU})_{X_i}.$$

Step 5: Utilizing S_4 and the "belief rule", we obtain

$$(S_5) : FA \mid\equiv MU \mid\equiv (R_{MU})_{X_i}.$$

Step 6: According to Msg_2, we obtain

$$(S_6) : HA \lhd (RID_i, ID_{MU}, R_{MU}, X_i, R_{FA}, ID_{FA})_{K_{FH}}.$$

Step 7: Utilizing the S_6 and A_3 with the "message meaning rule", the following is obtained

$$(S_7) : HA \mid\equiv FA \mid\sim (RID_i, ID_{MU}, R_{MU}, X_i, R_{FA}, ID_{FA})_{K_{FH}}.$$

Step 8: Now, using S_7 and A_4 with the "freshness rule", we obtain

$$(S_8) : HA \mid\equiv \#(RID_i, ID_{MU}, R_{MU}, X_i, R_{FA}, ID_{FA})_{K_{FH}}.$$

Step 9: Utilizing S_7 and S_8 with the "nonce verification rule", the following is obtained

$$(S_9) : HA \mid\equiv FA \mid\equiv (RID_i, ID_{MU}, R_{MU}, X_i, R_{FA}, ID_{FA})_{K_{FH}}.$$

Step 10: According to Msg_3, we obtain

$$(S_{10}) : FA \lhd (ID_{FA}, R_{FA}, R_{MU})_{K_{FH}}.$$

Step 11: Utilizing S_{10} and A_5 with the "message meaning rule", the following is obtained

$$(S_{11}) : FA \mid\equiv HA \mid\sim (ID_{FA}, R_{FA}, R_{MU})_{K_{FH}}.$$

Step 12: Now, using S_{11} and A_6 with the "freshness rule", we obtain

$$(S_{12}) : FA \mid\equiv \#(ID_{FA}, R_{FA}, R_{MU})_{K_{FH}}.$$

Step 13: Utilizing S_{11} and S_{12} with the "nonce verification rule", the following is obtained

$$(S_{13}) : FA \mid\equiv HA \mid\equiv (ID_{FA}, R_{FA}, R_{MU})_{K_{FH}}.$$

Step 14: According to Msg_4, we could obtain

$$(S_{14}) : MU \lhd (ID_{MU}, R_{MU}, R_{FA}, (MU \overset{SK}{\longleftrightarrow} FA))_{X_i}.$$

Step 15: Utilizing S_{14} and A_7 with the "message meaning rule", we obtain

$$(S_{15}) : MU \mid\equiv FA \mid\sim (ID_{MU}, R_{MU}, R_{FA}, (MU \overset{SK}{\longleftrightarrow} FA))_{X_i}.$$

Step 16: Now, using S_{15} and A_8 with the "freshness rule", the following is obtained

$$(S_{16}) : MU \mid\equiv \#(ID_{MU}, R_{MU}, R_{FA}, (MU \overset{SK}{\longleftrightarrow} FA))_{X_i}.$$

Step 17: Utilizing S_{15} and S_{16} with the "nonce verification", we obtain

$$(S_{17}) : MU \mid\equiv FA \mid\equiv (ID_{MU}, R_{MU}, R_{FA}, (MU \overset{SK}{\longleftrightarrow} FA))_{X_i}.$$

Step 18: Utilizing S_{17} and the belief rule, we obtain

$$(S_{18}) : MU \mid\equiv FA \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA). \quad \textbf{(Goal 3)}$$

Step 19: Now, using S_{18} and A_9 with the "jurisdiction rule", the following is obtained

$$(S_{19}) : MU \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA). \quad \textbf{(Goal 1)}$$

Step 20: Because of $SK = h(R_{MU}||R_{FA})$, from the S_5, S_9, S_{13} and S_{17} we obtain

$$(S_{20}) : FA \mid\equiv MU \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA). \quad \textbf{(Goal 4)}$$

Step 21: Utilizing S_{19} and A_{10} with the "jurisdiction rule", we obtain

$$(S_{21}) : FA \mid\equiv (MU \overset{SK}{\longleftrightarrow} FA). \quad \textbf{(Goal 2)}$$

Based on goals 1 to 4, we proved that MU, FA, and HA are securely mutually authenticated. We assessed that the proposed scheme ensured mutual authentication between MU, FA, and HA.

6.4. ROR Model Analysis

To evaluate the session key (SK) security of the protocol from the malicious adversary U_A, the proposed protocol performs the ROR model [26], which is a widely known formal security analysis. We first introduce the ROR model before doing a SK security proof for the proposed protocol.

Participants: There are three participants: the mobile user $P_{MU}^{t_1}$, the foreign agent $P_{FA}^{t_2}$, and the home agent $P_{HA}^{t_3}$ are instances t_1^{th} of the MU, t_2^{th} of the FA, and t_3^{th} of the HA, respectively.

Partnering: The instances t_1^{th} and t_2^{th} are partners if they satisfy the following conditions: (1) t_1^{th} and t_2^{th} are in the accept state, (2) t_1^{th} and t_2^{th} authenticate each other mutually sharing the same *sid*, and (3) t_1^{th} and t_2^{th} are mutually authenticated.

Freshness: If the U_A does not obtain the SK between MU and FA by utilizing the reveal query *Reveal*, the instance t_1^{th} or t_2^{th} is considered fresh.

Adversary: In the ROR model, the U_A can eavesdrop, modify, delete, or insert the exchanged messages during the communication. Furthermore, the U_A will have the access to the following queries.

- $Execute(P_{MU}^{t_1}, P_{FA}^{t_2}, P_{HA}^{t_3})$: It denotes that U_A performs the eavesdropping attack by eavesdropping exchanged messages between MU, FA, and HA over wireless communication.
- $CorruptDevice(P_{MU}^{t_1})$: It is modeled from the mobile device lost/stolen attack, in which the U_A is able to extract the secret data in the mobile device.

- $Send(P^t, M)$: In this query, the U_A can dispatch a message M to the instance P^t and can also reply accordingly.
- $Test(P^t)$: It corresponds to the semantic security of the SK_{ij} between MU and FA following the indistinguishability style in the ROR model [26]. In this query, before the experiment starts, an unbiased coin c is tossed. If the U_A executes *Tset* query and the established SK_{ij} is fresh, then P^t returns SK_{ij} for the case when $c = 1$ or a random value when $c = 0$. On the other cases, it returns a null value ($\perp$).
- $Reveal(P^t)$: With this query, the U_A can reveal the SK_i created by its partner to U_A in the current session.

Semantic security of the session key: In this formal security model, the malicious adversary U_A must distinguish between an instance's actual SK and a random secret key. The U_A can perform *Test* queries to either $P_{MU}^{t_1}$ or $P_{FA}^{t_2}$, and its output is checked for consistency against the random bit c. If the condition $c' = c$ is valid, the U_A wins the game. Otherwise, the U_A loses the game. Let $Succ$ denote an event that is U_A winning the game. Therefore, the advantage of U_A in breaking the semantic security of our protocol P is shown in Equation (1). The proposed protocol P is secure relative to the ROR model when $Adv_P \leq \psi$, for any sufficiently small $\psi > 0$.

$$Adv_P = |2 \cdot Pr[Succ] - 1| \tag{1}$$

Random oracle: In this paper, all the participants and the malicious adversary U_A can access a collision-resistant one-way hash function $h(\cdot)$. We model $h(\cdot)$ as a random oracle, say $Hash$.

6.4.1. Security Proof

We utilized Zipf's law [27] to assess the SK security of our protocol and the detailed theorems are given as follows:

Theorem 1. *If Adv_{U_A} denotes the advantage function of the U_A in violating SK security of our protocol. Then, we obtain the following.*

$$Adv_{U_A} \leq \frac{q_h^2}{|Hash|} + 2\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\}$$

where $Hash$, q_{send}, and q_h are the number of Hash queries, the number of Send queries, and the range space of the hash function $h(.)$, respectively; l_b is the number of bits present in the MU_i's biometric secret key b_i; and s and C are the Zipf's parameters [27].

Proof. We follow the proof as presented in [28,29]. A sequence of five games denoted by GM_i, where $i \in [0, 3]$, are defined to demonstrate the SK security of our protocol. $Succ_i$ denotes the probability of U_A winning the game GM_i. Each game is described in detail as follows.

- **Game** GM_0: This game is considered as an actual attack by the U_A for the proposed protocol P. Since the bit c is guessed at the beginning of G_0. According to this game, we obtain the following:

$$Adv_P = |2 \cdot Pr[Succ_0] - 1|. \tag{2}$$

- **Game** GM_1: This game is modeled so that the U_A performs an eavesdropping attack in which the exchanged messages $\{M_1, M_2, RID_i, Q_M\}$, $\{M_1, RID_i, Q_M, ID_{FA}, M_3, Q_F\}$, $\{M_4, M_5, Q_H\}$, and $\{M_4, Q_{MF}\}$ are intercepted during the authentication phase using the $Execute(P_{MU}^{t_1}, P_{FA}^{t_2}, P_{HA}^{t_3})$ query. Then, U_A performs the *Test* query to check whether it is the real SK or a random number. In the proposed protocol, the SK_i is calculated as $SK_i = h(R_{MU}||R_{FA})$. To derive SK_i, the U_A needs secret credentials, such as R_{MU}, R_{FA}, and X_i. Consequently, the U_A's probability in winning GM_1 by eavesdropping on the exchanged messages does not increase. We can obtain

$$Pr[Succ_1] = Pr[Succ_0]. \tag{3}$$

- **Game** GM_2: The difference between GM_1 and GM_2 is that the *Hash* and *Send* queries are included in GM_2. This game can be considered as an active attack in which the U_A may try to fool a legitimate entity to accept the exchanged messages modified by the U_A. All exchanged messages are protected by using the collision-resistant one-way hash function $h(\cdot)$. All exchanged messages are constructed using the random credentials R_{MU}, R_{FA}, and X_i. All exchanged messages are constructed using the random credentials R_{MU}, R_{FA}, and X_i and these messages are protected by using the collision-resistant one-way hash function $h(.)$. Using birthday paradox, we can obtain the following result:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2|Hash|}. \tag{4}$$

- **Game** GM_3: In the final game, the *CorruptDevice* query is modeled. In this case, a U_A can extract the secret parameters $\{A_i, B_i, P_i\}$ from a mobile device's memory utilizing the power-analysis attack. Here, $A_i = X_i \oplus h(RID_i||RPW_i)$, $B_i = h(RPW_i||X_i)$ and $P = Gen(BIO_i)$. It is computationally infeasible for U_A to derive the real identity ID_{MU} and password PW_{MU} of MU correctly via the *Send* query without HA's master key K_s and secret parameter X_i. The probability of guessing the biometric key b_i of l_b bits by the U_A is approximately $\frac{1}{2^{l_b}}$. Consequently, the GM_2 and GM_3 are indistinguishable if password/biometrics guessing attacks are not implemented. Therefore, utilizing Zipf's law [27], we can obtain the following result:

$$|Pr[Succ_3] - Pr[Succ_2]| \leq max\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\} \tag{5}$$

As all the games are executed, the U_A must guess the exact bit c. Thus, we can obtain the following result:

$$Pr[Succ_3] = \frac{1}{2} \tag{6}$$

With Equations (1), (2), and (5), we can obtain the result as below:

$$\begin{aligned} \frac{1}{2}Adv_{U_A} &= |Pr[Succ_0] - \frac{1}{2}| \\ &= |Pr[Succ_1] - \frac{1}{2}| \\ &= |Pr[Succ_1] - Pr[Succ_3]|. \end{aligned} \tag{7}$$

Using Equations (4)–(6), we can obtain the following result, which uses the triangular inequality.

$$\begin{aligned} \frac{1}{2}Adv_{U_A} &= |Pr[Succ_1] - Pr[Succ_3]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ &\quad + |Pr[Succ_2] - Pr[Succ_3]| \\ &\leq \frac{q_h^2}{2|Hash|} + max\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\}. \end{aligned} \tag{8}$$

Finally, multiplying both sides of Equation (7) by a factor of two, we can obtain the result as belows:

$$Adv_{U_A} \leq \frac{q_h^2}{|Hash|} + 2max\{C \cdot q_{send}^s, \frac{q_s}{2^{l_b}}\}.$$

□

7. AVISPA Simulation

We discuss a formal security validation of our protocol utilizing Automated Validation of Internet Security Protocols and Applications (AVISPA) [30,31], which evaluates the security of the protocol to

MITM attacks and replay attacks. To evaluate the AVISPA, the environment and session of the protocol must be implemented utilizing the High-Level Protocols Specification Language (HLPSL).

7.1. HLPSL Specification

According to HLPSL, we consider three roles: the MU, the FA, and the HA. We define the *environment* and *session* using HLPSL in Figure 6, which comprises the security goals. Figure 7 presents the role specification of MU and FA.

As shown in Figure 7, the MU initially receives the message and changes the state value from 1 to 2. Then, the MU sends the registration request messages $\{ID_{MU}, RPW_i\}$ to HA over a secure channel. Then, MU receives the secret parameter $\{A_i, B_i\}$ from HA and MU updates the state value from 1 to 2. When a MU requests access to roaming services, the MU must send a login request message $\{M_1, M_2, RID_i, Q_M\}$ to FA over an open channel. After that, MU declares $witness(MU, HA, mu\_ha\_mu, R'_{MU})$ and changes the state value from 2 to 3. Finally, MU receives the message $\{M_4, Q_{MF}\}$ from FA. Then, MU checks whether $Q^*_{MF} \stackrel{?}{=} Q_{MF}$. If this holds, the MU successfully authenticates the FA. The role specification of FA and HA are similarly defined. Furthermore, Figure 8 presents the role specification of HA.

```
%%%%% role environment()

def=
const mu, ha, fa : agent,
skmuha: symmetric_key,
h: hash_func,
idmu, idfa: text,
sp1, sp2, sp3, sp4: protocol_id,
mu_ha_mu, ha_fa_mu, ha_mu_rfa, fa_ha_rfa:
protocol_id

intruder_knowledge={mu,ha,fa,h,idfa,dimu}
composition
session(mu,ha,fa,skmuha,h)^session(i,ha,fa,skmu,ha,h)
^session(mu,i,fa,skmuha,h)
^session(mu,ha,i,skmuha,h)

end role

%%%%% goal

secrecy_of sp1, sp2, sp3, sp4
authentication_on mu_ha_mu, ha_fa_mu
authentication_on ha_mu_rfa, fa_ha_rfa
end goal

environment()
```

(**a**) Environment

```
%%%%% Role for the session

role session(MU, HA, FA: agent, Skmuha:
symmetric_key, H:hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
mobileuser(MU,HA,FA, Skmuha, H, SN1, RV1)
^homeagent(MU, HA, FA, Skmuha, H, SN2, RV2)
^foreignagent(MU, HA, FA, H, SN3, RV3)
end role
```

(**b**) Session

Figure 6. Role specification for the environment and session.

```
%%%%% Role for MU

role mobileuser(MU, HA, FA : agent, Skmuha : symmetric_key, H:
hash_func, SND, RCV, : channel(dy))

played_by MU
def=
local State:nat,
  IDmu, PWmu, Ri, Pi, RPWi, RIDi, Xi, Ai, Bi, Rs, Ks : text,
  M1, M2, Qm, M3, Of, IDfa, Kfh, Rmu, Rfa, M4, M5, Qh, Ski, Qmf, text
const sp1, sp2, sp3, sp4, mu_ha_mu, ha_fa_mu, ha_mu_rfa, fa_ha_rfa:
protocol_id
init State:=0
transition
1. State=0∧RCV(start)=|>
State':=1∧Ri':=new()∧Pi':=new()
  ∧RPWi':=H(Pwmu.Ri')
  ∧SND({Idmu.RPWi'}_SKmuha)
  ∧secret({IDmu}, sp1, {MU,HA})
∧secret({Pwmu,Ri',Pi'}, sp2, {MU})

2. State=1∧RCV
({xor(H(H(IDmu,h(Pwmu.Ri')).Ks.Rs'),
H(RIDi'.H(Pwmu.Ri'))).H(H(Pwmu.Ri').H(H(IDmu.H(Pwmu.Ri')).Ks.Rs'))}_Skmuha)=|>
State':=2∧Rmu':=new()
  ∧M1':=xor(H(H(Idmu.H(PWmu.Ri')).Ks.Rs'), Rmu')
  ∧M2':=xor(IDmu, H(H(Idmu.H(PWmu.Ri')).Ks.Rs'))
  ∧Qm':=H(H(IDmu.H(PWmu.Ri')).H(H(IDmu.H(PWmu.Ri')).Ks.Rs').Rmu')
  ∧SND(M1'.M2'.H(IDmu.H(PWmu.Ri')).Qm')
  ∧witness(MU, HA, mu_ha_mu, Rmu')

3. State=2∧RCV(xor(Rfa',
H(H(IDmu.H(Pwmu.Ri')).H(H(Idmu.H(Pwmu.Ri')).Ks.Rs').Rmu')).H(Rmu'.Rfa'.H(Rmu'
.Rfa')))=|>
State':=3∧request(HA, MU, ha_mu_rfa, Rfa')
End role
```

(**a**) Mobile user

```
%%%%% Role for FA

role foreignagent(MU, HA, FA : agent, H: hash_func, SND, RCV:
channel(dy))
played_by FA
def=
local State: nat,
  IDmu, PWmu, Ri, Pi, RPWi, RIDi, Xi, Ai, Bi, Rs, Ks : text,
  M1, M2, Qm, M3, Qf, IDfa, Kfh, Rmu, Rfa, M4, M5, Qh, Ski, Qmf : text
const sp1, sp2, sp3, sp4, mu_ha_mu, ha_fa_mu, ha_mu_rfa, fa_ha_rfa:
protocol_id

init State:=0
transition
1.State=0∧RCV(xor(H(H(IDmu.H(PWmu.Ri')).Ks.Rs'), Rmu').xor(IDmu,
H(H(IDmu.H(PWmu.Ri')).Ks.Rs')).H(IDmu.H(Pwmu.Ri')).H(H(IDmu.H(PWmu.Ri')).H(
H(IDmu.H(Pwmu.Ri')).Ks.Rs').Rmu'))=|>
State':=1∧Rfa':=new()
    ∧M3':=xor(H(IDfa, Kfh.xor(H(H(IDmu.H(PWmu.Ri')).Ks.Rs'), Rmu')),Rfa')
    ∧SND(xor(H(H(IDmu.H(PWmu.Ri')).Ks.Rs'),
    Rmu').xor(Idmu.H(H(IDmu.H(PWmu.Ri')).Ks.Rs')).H(IDmu.H(PWmu.Ri')).H(H(ID
    mu.H(PWmu.Ri')).H(H(IDmu.(H(PWmu.Ri')).Ks.Rs').Rmu').IDfa.M3'.Qf')
    ∧secret({Kft},sp4, {FA, HA})
    ∧witness(FA, HA, fa_ha_rfa, Rfa')

2. State=1∧RCV(xor(Rfa',
    H(H(IDmu.H(PWmu.Ri')).H(H(IDmu.H(PWmu.Ri')).Ks.Rs').Rmu')).xor(Rmu',
    H(IDfa.Kfh.Rfa')).H(Rmu'.Rfa'.Kfh))=|>
state':=2∧SK':=H(Rmu'.Rfa')
    ∧Qmf':=H(Rmu'.Rfa'.SKi')
    ∧SND(xor(Rfa',
    H(H(IDmu.H(PWmu.Ri')).H(H(IDmu.H(PWmu.Ri')).Ks.Rs').Rmu')).Qmf)
    ∧request(HA, FA, ha_fa_mu, Rmu')
end role
```

(**b**) Foreign agent

Figure 7. Role specification for *MU* and *FA*.

```
%%%%% Role for HA

role homeagent(MU, HA, FA : agent, SKmuha : symmetric_key, H:
hash_func, SND, RCV: channel(dy))

played_by HA
def=
local State:nat,
  IDmu, PWmu, Ri, Pi, RPWi, RIDi, Xi, Ai, Bi, Rs, Ks, : test,
  M1, M2, Qm, M3, Qf, IDfa, Kfh, Rmu, Rfa, M4, M5, Qh, SKi, Qmf : text
const sp1, sp2, sp3, sp4, mu_ha_mu, ha_fa_mu, ha_mu_rfa, fa_ha_rfa:
protocol_id
init State:=0
transition

1. State=0∧RCV({IDmu.H(PWmu.Ri')}_SKmuha)=|>
 State':1∧secret({IDmu}, sp1, {MU, HA})
    ∧Rs':=new()
    ∧RIDi':=H(IDmu.H(PWmu.Ri'))
    ∧Xi':=H(RIDi'.Ks.Rs')
    ∧Ai':=xor(Xi',H(RIDi'.H(PWmu.Ri')))
    ∧Bi':=H(H(PWmu.Ri').Xi')
    ∧secret({Ks, Rs'}, sp3, {HA})
    ∧SND({Ai', Bi'}_SKmuha)

2.State=1∧RCV(xor(H(H(IDmu.H(PWmu.Ri')).Ks.Rs'),Rmu').xor(IDmu,H(H(IDmu.H(PW
mu.Ri')).Ks.Rs')).H(IDmu.H(PWmu.Ri')).H(H(IDmu.H(PWmu.Ri')).H(H(IDmu.H(PWmu.
Ri')).Ks.Rs').Rmu').IDfa.xor(H(IDfa.Kft.xor(H(H(IDmu.H(PWmu.Ri')).Ks.Rs'),
Rmu')),Rfa').H(IDfa.Kfh.Rfa'.xor(H(H(IDmu.h(PWmu.Ri')).Ks.Rs'), Rmu')))=|>
State':=2∧secret({Kfh},sp4,{FA,HA})
    ∧M4':=xor(Rfa', H(H(IDmu.H(PWmu.Ri')).H(H(IDmu.H(PWmu.Ri')).Ks.Ks').Rmu'))
    ∧Qh':=h(Rmu'.Rfa'.Kfh)
    ∧SND(M4'.M5'.Qh')
    ∧witness(HA, FA, ha_fa_mu, Rmu')
    ∧witness(HA, MU, ha_mu_rfa, Rfa')
    ∧request(MU, HA, mu_ha_mu, Rmu')
    ∧request(FA, HA, fa_ha_rfa, Rfa')
end role
```

Figure 8. Role specification for *HA*.

7.2. Result Analysis of AVISPA Simulation

We show the results of the AVISPA simulation using Constraint-Logic-based ATtack SEarcher (CL-AtSe) and On-the-Fly Model Checker (OFMC) to verify the security of our protocol. The CL-AtSe assessed the security of the protocol to replay attacks. The CL-AtSe verifies whether a legitimate user could perform the scheme by executing a search for a malicious adversary. Furthermore, the OFMC verifies the security of the proposed protocol to MITM attacks. The results, shown in Figure 9, demonstrate that the proposed protocol is secure against both MITM and replay attacks. The OFMC verification shows that the search time was 1.12 s for visiting 130 nodes, and the CL-AtSe verification analyzed three states with 0.08 s to translate.

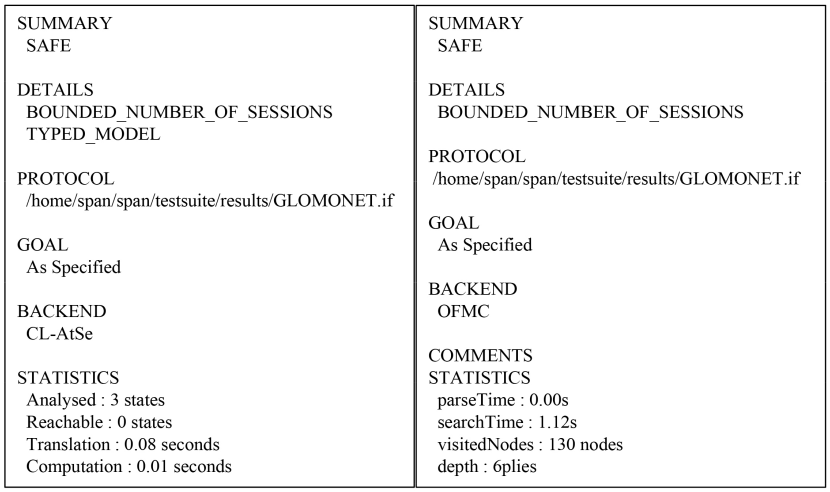

Figure 9. Analysis of the simulation results using CL-AtSe and OFMC.

8. Performance Analysis

This section assesses the performance of our protocol in terms of the computation cost, communication cost, and security properties. We also compared the proposed protocol with other related protocols [6,8,15,16]. We demonstrated that the proposed scheme provides better security properties and efficiency as compared to other related schemes.

8.1. Computation Cost

We compared the computation costs of our protocol to those of existing protocols [6,8,15,16]. Referring to [32,33], we estimated the approximate execution time of each cryptographic operation on the following configurations of the computer system. Windows 7 OS and Android phones were used and the system structure of the mobile phone ws Android 4.4.4KTU84P along with a 2 GB RAM and 1.8 GHz processor. Furthermore, the configurations of the computer system were Windows 7, Professional with an Intel(R) Core(TM) 2 Quad CPU Q8300, 2 GB RAM, @2.50 Hz. The XOR function was not included as it was negligible compared to other functions. The following shows the time complexity for the computational analysis.

- T_h : The time complexity of a one-way hash function operation $\approx$ 0.0005 s.
- T_m : The time complexity of a modular multiplication operation $\approx$ 0.00125 s.
- T_{mm} : The time complexity of a modular exponentiation operation $\approx$ 0.522 s.
- T_{pm} : The time complexity of a elliptic curve point multiplication operation $\approx$ 0.0503 s.
- T_{sym} : The time complexity of a symmetric encryption/decryption operation $\approx$ 0.0087 s.
- T_{ecc} : The time complexity of a asymmetric encryption/decryption operation $\approx$ 0.3057 s.

The total computation costs for our protocol and for Madhusudhan et al.'s scheme were $27T_h$ ($\approx$0.0135 s) and $10T_h + 3T_{mm} + 4T_{sym}$ ($\approx$1.6058 s), respectively. Table 5 presents the result for

computation costs. Consequently, we provided better efficient computation costs compared with related schemes because it only uses one-way hash functions. Therefore, the proposed scheme is considered efficient in the application for practical mobile environments.

Table 5. Computation cost comparison.

Schemes	Registration	Login and Authentication	Total	Total Cost (s)
He et al. [6]	$7T_h + 1T_{sym}$	$17T_h + 4T_{sym} + 8T_{asym}$	$24T_h + 5T_{sym} + 8T_{asym}$	2.5272
Kuo et al. [8]	$2T_h$	$17T_h + 6T_{pm}$	$19T_h + 6T_{pm}$	0.3113
Karuppiah et al. [15]	$5T_h + 1T_{sym}$	$24T_h + 1T_m + 3T_{mm} + 3T_{sym}$	$29T_h + 1T_m + 3T_{mm} + 4T_{sym}$	1.60785
Madhusudhan et al. [16]	$3T_h + 1T_{mm}$	$7T_h + 2T_{mm} + 4T_{sym}$	$10T_h + 3T_{mm} + 4T_{sym}$	1.6058
Ours	$5T_h$	$22T_h$	$27T_h$	0.0135

T_m: modular multiplication, T_{mm}: modular exponentiation, T_h: hash function, T_{pm}: elliptic curve point multiplication, T_{sym}: symmetric encryption/decryption, T_{asym}: asymmetric encryption/decryption.

8.2. Communication Cost

We evaluated the communication costs of our protocol with existing schemes [6,8,15,16]. According to [34], we define that the identity, timestamp, and random number are 128 bits, 32 bits, and 64 bits, respectively. In addition, hash functions and symmetric key encryption require 160 bits and 256 bits, respectively. Finally, the modular operation and the scalar multiplication operation on the elliptic curve define 1024 bits and 320 bits, respectively.

Table 6 tabulates the analysis results of the communication costs. In Figure 4, the transmitted messages require (160 + 160 + 160 + 160 = 640 bits), (160 + 160 + 160 + 160 + 128 + 160 + 160 = 1088 bits), (160 + 160 + 160 = 480 bits), and (160 + 160 = 320 bits). Consequently, the total communication cost of our protocol was 3136 bits. Although the proposed protocol had a higher communication cost than Madhusudhan et al.'s protocol [16] and it provided better security than Madhusudhan et al.'s scheme [16].

Table 6. Communication cost comparison.

Schemes	Registration Process	Login and Authentication Process	Total Cost
He et al. [6]	704 bits	4992 bits	5696 bits
Kuo et al. [8]	640 bits	3872 bits	4512 bits
Karuppiah et al. [15]	640 bits	4224 bits	4864 bits
Madhusudhan et al. [16]	1184 bits	1344 bits	2528 bits
Ours	608 bits	2528 bits	3136 bits

9. Conclusions

In this paper, we assessed that Madhusudhan et al.'s authentication scheme did not prevent various attacks. Furthermore, we assessed that their protocol could not achieve user authentication. We proposed a secure and efficient three-factor authentication protocol for roaming services in GLOMONET to improve the security flaws of Madhusudhan et al.'s scheme. Our scheme was able to resist various attacks, such as masquerade, replay, session key disclosure, and mobile device theft attacks and could ensure anonymity and user authentication. We demonstrated that our scheme achieved secure mutual authentication among the mobile user, the foreign agent, and the home agent by performing BAN logic analysis.

Furthermore, we assessed a formal security validation analysis of our protocol utilizing the ROR model and AVISPA simulation. We compared the computation costs and security features with existing schemes. The three-factor based proposed scheme provided a great improvement in terms of the security level compared with two-factor based existing schemes and also preserved the low

computation cost. The principal merit of the proposed scheme was resistance against potential attacks in GLOMONET. Therefore, the proposed scheme satisfies the security requirements for roaming service and is suitable for practical mobile environments.

Author Contributions: Conceptualization, S.Y.; Formal analysis, J.L., and Y.P. (YoHan Park); Software, S.Y., and J.L.; Supervision, Y.P. (YoungHo Park); Validation, S.L., B.C., Y.P., (YoHan Park) and Y.P. (YoungHo Park); Writing–original draft, S.Y.; Writing–review and editing, S.L., B.C., Y.P., (YoHan Park) and Y.P. (YoungHo Park). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government(MIST) (No.2018-0-00312, Developing technologies to predict, detect, respond, and automatically diagnose security threats to automotive Ethernet-based vehicle).

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

References

1. Gope, P.; Hwang, T. An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks. *J. Netw. Comput. Appl.* **2016**, *62*, 1–8. [CrossRef]
2. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Choo, K.K.R. A robust biometrics based three-factor authentication scheme for global mobility networks in smart city. *Future Gener. Comput. Syst.* **2018**, *83*, 607–618. [CrossRef]
3. Lu, Y.; Xu, G.; Li, L.; Yang, Y. Robust privacy-preserving mutual authenticated key agreement scheme in roaming service for global mobility networks. *IEEE Syst. J.* **2019**, *13*, 1454–1465. [CrossRef]
4. Lee, T.F. User authentication scheme with anonymity, unlinkability and untrackability for global mobility networks. *Secur. Commun. Netw.* **2013**, *6*, 1404–1413. [CrossRef]
5. Lee, C.C.; Lai, Y.M.; Chen, C.T.; Chen, S.D. Advanced secure anonymous authentication scheme for roaming service in global mobility networks. *Wirel. Pers. Commun.* **2017**, *94*, 1281–1296. [CrossRef]
6. He, D.; Ma, M.; Zhang, Y.; Chen, C.; Bu, J. A strong user authentication scheme with smart cards for wireless communications. *Comput. Commun.* **2011**, *34*, 367–374. [CrossRef]
7. Jiang, Q.; Ma, J.; Li, G.; Yang, L. An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks. *Wirel. Pers. Commun.* **2013**, *68*, 1477–1491. [CrossRef]
8. Kuo, W.C.; Wei, H.J.; Cheng, J.C. An efficient and secure anonymous mobility network authentication scheme. *J. Inf. Secur. Appl.* **2014**, *19*, 18–24. [CrossRef]
9. Park, K.S.; Park, Y.H.; Park, Y.H.; Reddy, A.G.; Das, A.K. Provably secure and efficient authentication protocol for roaming service in global mobility networks. *IEEE Access* **2017**, *5*, 25110–25125. [CrossRef]
10. Zhu, J.; Ma, J. A new authentication scheme with anonymity for wireless environments. *IEEE Trans. Consum. Electron.* **2004**, *50*, 231–235.
11. Lee, C.C.; Hwang, M.S.; Liao, I.E. Security enhancement on a new authentication scheme with anonymity for wireless environments. *IEEE Transs. Ind. Electron.* **2006**, *53*, 1683–1687. [CrossRef]
12. Wu, C.C.; Lee, W.B.; Tsaur, W.J. A secure authentication scheme with anonymity for wireless communications. *IEEE Commun. Lett.* **2008**, *12*, 722–723.
13. Li, C.T.; Lee, C.C. A novel user authentication and privacy preserving scheme with smart cards for wireless communications. *Math. Comput. Model.* **2012**, *55*, 35–44. [CrossRef]
14. Das, A.K. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. *Netw. Sci.* **2013**, *2*, 12–27. [CrossRef]
15. Karuppiah, M.; Saravanan, R. A Secure Authentication Scheme with User Anonymity for Roaming Service in Global Mobility Networks. *Wirel. Pers. Commun.* **2015**, *84*, 2055–2078. [CrossRef]
16. Madhusudhan, R.; Shashidhara. A secure and lightweight authentication scheme for roaming service in global mobile networks. *J. Inf. Secur. Appl.* **2018**, *38*, 96–110. [CrossRef]
17. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
18. Mohit, P.; Amin, R.; Karati, A.; Biswas, G.P.; Khan, M.K. A standard mutual authentication protocol for cloud computing based health care system. *J. Med. Syst.* **2017**, *41*, 50. [CrossRef]
19. Amin, R.; Islam, S.K.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *80*, 483–495. [CrossRef]
20. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology–CRYPTO*; Lecture Notes in Computer Science; Springer: Santa Barbara, CA, USA, 1999; pp. 388–397.

21. Yu, S.J.; Lee, J.Y.; Lee, K.K.; Park, K.S.; Park, Y.H. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors* **2018**, *18*, 3191. [CrossRef]
22. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Interlaken, Switzerland, 2004; pp. 523–540.
23. Odelu, V.; Das, A.K.; Goswami, A. An efficient biometric-based privacy-preserving three-party authentication with key agreement protocol using smart cards. *Secur. Commun. Netw.* **2015**, *8*, 4136–4156. [CrossRef]
24. Park, Y.H.; Park, Y.H. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [CrossRef] [PubMed]
25. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
26. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password based authenticated key exchange in the three-party setting. In *Public Key Cryptography*; Springer: Les Diablerets , Switzerland, 2005; pp. 65–84.
27. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
28. Yu, S.J.; Park, K.S.; Lee, J.Y.; Park, Y.H.; Park, Y.H.; Lee, S.W.; Chung, B.H. Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment. *Appl. Sci.* **2020**, *10*, 1758. [CrossRef]
29. Park, K.S.; Park, Y.H.; Park, Y.H.; Das, A.K. 2PAKEP: Provably Secure and Efficient Two-Party Authenticated Key Exchange Protocol for Mobile Environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
30. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 8 February 2020).
31. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 8 February 2020).
32. Kumar, V.; Jangirala, S.; Ahmad, M. An efficient mutual authentication framework for healthcare system in cloud computing. *J. Med. Syst.* **2018**, *42*, 142. [CrossRef]
33. Chandrakar, P.; Om, H. A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC. *Comput. Commun.* **2017**, *110*, 26–34. [CrossRef]
34. Lee, H.J.; Lee, D.H.; Moon, J.H.; Jung, J.W.; Kang, D.W.; Kim, H.S.; Won, D.H. An improved anonymous authentication scheme for roaming in ubiquitous networks. *PLoS ONE* **2018**, *13*, e0193366. [CrossRef]