


ITC 1/48 Journal of Information Technology and Control Vol. 48 / No. 1 / 2019 pp. 129-145 DOI 10.5755/j01.itc.48.1.17270	An Authentication Framework for Roaming Service in Global Mobility Networks	
	Received 2016/12/15	Accepted after revision 2018/10/09
	 http://dx.doi.org/10.5755/j01.itc.48.1.17270	

An Authentication Framework for Roaming Service in Global Mobility Networks

Jangirala Srinivas

Jindal Global Business School, O. P. Jindal Global University, Haryana 131 001, India

Dheerendra Mishra

Department of Mathematics, The LNM Institute of Information Technology, Jaipur, India;
e-mail: dheerendra.mishra@lnmiit.ac.in

Sourav Mukhopadhyay

Department of Mathematics, Indian Institute of Technology, Kharagpur 721 302, India

Saru Kumari

Department of Mathematics, Ch. Charan Singh University, Meerut, India

Vandana Guleria

Department of Mathematics, Birla Institute of Technology, Mesra, Ranchi, India

Corresponding author: dheerendra.m@gmail.com

In global mobility networks (GLOMONET), to provide secure and privacy-preserving communication among authorized mobile users in roaming services is not an easy task. To achieve authorized communication, mutual authentication is performed among legal users in GLOMONET. Therefore, security as well as privacy should be addressed in designing the security protocols for GLOMONET. In recent years, most of the research work is focused on one-way authentication and does not have desirable security attributes. In this paper, we discuss the development of authentication protocol for GLOMONET. To address security and privacy issues in authorized communication, we proposed a provably secure authentication protocol for GLOMONET. To identify the resistance against known attacks, we have analyzed the scheme against all known attacks. The comparative study on the security and performance with the related results manifests that the proposed scheme addresses the security and privacy challenges and avails comparable performance.

KEYWORDS: Global mobility networks, authentication, anonymity, untraceability, security.

1. Introduction

Global Mobility Networks (GLOMONET) provide the roaming services for the mobile users, which enable them to use the extended services in their home agent whenever they enter into a foreign agent zone irrespective of their locations [2]. When a mobile user (*MU*) enters a foreign agent zone, there should be some mechanism of communication between a mobile user, home agent (*HA*), and foreign agent (*FA*). A conventional frame for roaming service is shown in Figure 1. To communicate with *FA*, firstly, a session is developed with the home agent by *MU*. Then, *MU* communicates with *FA* with the help of *HA*. A registered *MU* gets the services only when he/she is successfully authenticated by *FA*. Meanwhile, in practice, two major concerns (privacy and security) are identified in roaming services. Thus, in attaining the roaming services, the mutual authentication mechanism is designed to address the issues of authorized communication. The secure communication is being achieved using key agreement [1, 21, 27]. As the communication technologies are developing rapidly, the entire world is reciprocally more connected. Due to this privacy risks are at stake. Privacy concerns have increasingly got attention from governments, corporations, and individuals. It is desirable to control sensitive information. However, in the environment of Internet where information sharing is very easy, this problem (controlling sensitive information) does not have an easy solution. Thus, a balanced approach is required between information sharing and privacy. For roaming services in GLOMONET, the first authentication scheme was introduced by Zhu and Ma

[36]. Since then, there were many proposals for a secure design with low computation cost [6, 10-11, 18-19, 29, 31]. It is observed that most of the existing similar schemes failed to provide the user anonymity along with untraceability.

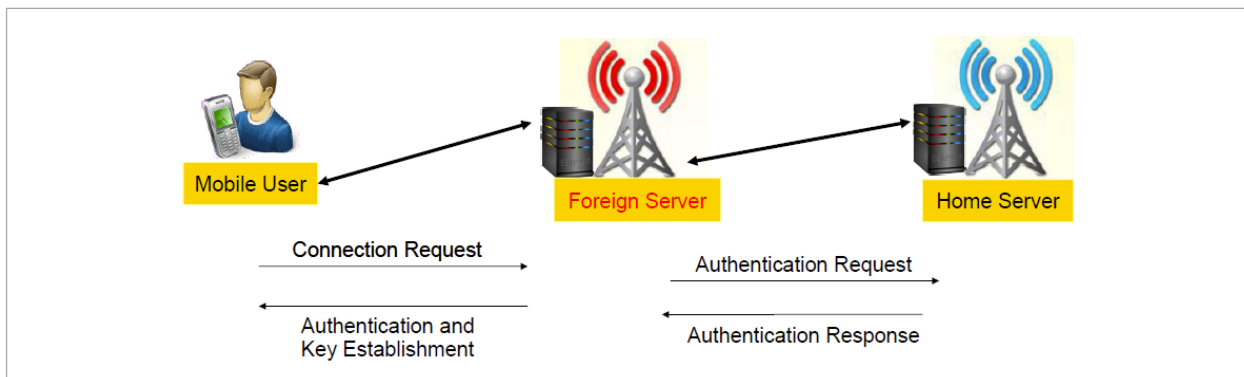
In 2011, two new authentication schemes were proposed by Chen *et al.* [4-5]. However, Xie *et al.* [33] enlightened the security shortcomings of both the schemes. For roaming services, Mun *et al.* [22] gave an anonymous authentication scheme in 2012. However, Mun *et al.*'s scheme is shown failure by Kim and Kwak [15] due to the design flaws in their scheme.

Based on quadratic residue, Jiang *et al.* [13] gave an authentication scheme. But, Wen *et al.* [30] and He *et al.* [12] both pointed out that service attack, replay attack and impersonation attack are not addressed in Jiang *et al.*'s scheme. To overcome the security flaws, Wen *et al.* [30] and He *et al.* [12] independently presented improved schemes. Using the modular exponentiation operations, Shin *et al.* [26] gave an efficient authentication scheme. Unfortunately, Farash *et al.* [9] pointed out that both [30] and [26] failed to resist user traceability, impersonation attack and session key disclosure attack. Due to these failures, a lightweight authentication scheme has been presented by them.

Recently (2015), Zhang *et al.* [34] designed an authentication scheme by adopting symmetric key and rational points multiplication in elliptic curve to preserve the privacy in GLOMONET. As a part of our case study, we figured out that their scheme fails to ensure the security goals. We came across some security shortcomings

Figure 1

Conventional frame for roaming services



in Zhang *et al.*'s scheme, such as (a) password guessing attack; (b) user anonymity and traceability; (c) replay attack; and (d) user impersonation attack. In a thorough study, we observed that Zhang *et al.*'s scheme can be enhanced such that improved design can collectively address all the security attributes.

1.1. Security Goals

In providing secure roaming services, an efficient and anonymous authentication scheme should possess the following attributes:

Mobile user's privacy: Roaming authentication has issues with two types of privacy: (i) *MU*'s real identity; and (ii) *MU*'s roaming line. Both types should be required to provide privacy when running the roaming services.

Untraceability: No tracking of *MU* by looking at the connections, i.e., the third party must not be able to identify *MU* by looking at his interactions with different *FAs*.

User's validation checking: The registration of *MU* with *HA* should be correctly identified by the foreign agent.

Prohibit impersonation attack: Only the legitimate mobile user and the network agent (home or foreign) should be able to authenticate.

Ensure a private session key: The parties should agree upon a fresh session key. The session key can be established among the authorized participants.

Prohibit replaying attack: Interception of the messages by attacker should not reveal any sensitive information of the participants even by replaying the previous messages.

1.2. Our Contributions

The contributions are as follows:

- In a thorough study of the recently proposed Zhang *et al.*'s [34] scheme, we concluded that their scheme suffers from several attacks such as (a) password guessing attack; (b) user anonymity and traceability; (c) replay attack; and (d) user impersonation attack.
- To overcome the shortcomings of Zhang *et al.*'s scheme, we present an improved scheme which inherits Zhang *et al.*'s scheme and successfully withstands the possible known attacks.
- Moreover, the proposed scheme is proved secure assuming the hardness of ECDH assumption.

The scheme is presented with a valid proof which preserves all the security attributes.

- Furthermore, the proposed scheme is computationally efficient in comparison to Zhang *et al.*'s scheme and performs better in comparison to other existing similar schemes.

1.3. Cryptographic Preliminaries

1.3.1. Elliptic Curve

An elliptic curve E over a finite field F_p consists of points satisfying the equation $y^2 = x^3 + ax + b \pmod{p}$ along with the point at infinity, where $a, b \in F_p$ and $4a^3 + 27b^2 \pmod{p} \neq 0$. We omit \pmod{p} and draw out the following assumptions [17].

Assumption1:

Elliptic curve discrete logarithm problem (ECDLP): Suppose $P, Q \in G$ with $Q = aP$, it is computationally hard to compute the integer a , where G is the group of rational points on the elliptic curve $E(F_p)$.

Assumption2:

Elliptic curve Diffie–Hellman problem (ECDHP):

If $\alpha P, \beta P \in G$ and α, β are positive integers, it is hard to compute $\alpha\beta P$.

1.3.2. Biohashing

Biometric systems are applicable for human authentication in validating the security task to enable the authorized access, however, these systems face specific security challenges such as noisy data input which causes denial of service attack. It has impact on the usability of the system by failing to identify authorized consumers [23]. To overcome these problems, BioHashing technique is introduced [28]. It has substantial functional advantages such as clean separation of the genuine, zero error rate and imposter populations [14]. It has the following functions:

- Extraction of biometric parameter represented in a vector form $\Gamma \in \mathcal{R}^n$, where n is the feature length.
- Input token is used to generate m pseudo-random vector $\{r_i \in \mathcal{R}^M \mid i = 1, 2, \dots, m\}$.
- The Gram-Schmidt process is employed for $\{r_i \in \mathcal{R}^M \mid i = 1, 2, \dots, m\}$ and orthonormal pseudorandom vectors are obtained $\{r \perp i \in \mathcal{R}^n \mid i = 1, 2, \dots, m\}$, $n \geq m$.
- Calculate $\{\langle \Gamma \mid r \perp i \rangle \mid i = 1, 2, \dots, m\}$, where $\langle \cdot \mid \cdot \rangle$ indicates inner product.

- Compute m -bit BioHash template, $b = \{b_i \mid i = 1, 2, \dots, m\}$ using a threshold κ obtained from

$$b_i = \begin{cases} 1, & \text{if } \langle \Gamma \mid r \perp i \rangle \leq \kappa, \\ 0, & \text{if } \langle \Gamma \mid r \perp i \rangle > \kappa. \end{cases} \quad (1)$$

1.4. Road Map of the Paper

The details are as follows. In Section 2, we present the review of Zhang *et al.*'s scheme. Security weaknesses of Zhang *et al.*'s scheme are shown in Section 3. The proposed authentication scheme is provided in Section 4. Further, in Section 5, we present our provably secure scheme in formal model.

The informal security analysis and discussion of the proposed scheme is done in Section 6. In Section 7, the performance of our scheme with the other existing similar schemes is compared. Finally, the paper is concluded in Section 8.

2. Review of Zhang *et al.*'s Scheme

2.1. Registration Phase

MU registers with the home agent HA if he/she wishes to get services from Global Mobility Network [34]. In this phase, the communication with the participants is done through secure communication channel.

R1. MU selects his/her identity ID_{MU} and password PW_{MU} . After computing $V = h(PW_{MU} \parallel m)$, MU sends the message $\{ID_{MU}, V\}$ to home agent HA where $m \in Z_p^*$ is a random number.

R2. HA receives the message $\{ID_{MU}, V\}$ and a random number n is generated to undergo computations using HA 's master key K , $MID = Enc_K(ID_{MU}, n)$, $C = V \oplus h(ID_{MU} \parallel K)$. Further HA sends $\{MID, C, ID_{HA}\}$ to MU .

R3. MU stores $\{ID_{MU}, ID_{HA}, C, MID, m\}$ into SC .

2.2. Authentication and Key Establishment Phase

The registered MU opts services from FA . Before this, both the parties agree upon a common session key. Moreover, the communication between FA and HA is done through the secure channel. The details are shown as follows:

A1. MU input his/her login credentials into the smart-card to compute $V = h(PW_{MU} \parallel m)$ and $V' = C \oplus V$. Further, MU chooses a random number $a \in Z_p^*$ and computes aP and $Auth_{MU} = Enc_{V'}(ID_{MU}, aP)$. MU sends the message $m_1 = \{ID_{HA}, MID, Auth_{MU}\}$ to the foreign agent FA .

A2. FA receives the message m_1 , and selects a random number $b \in Z_p^*$, computes bP and $D_{FA} = Enc_{K_{FH}}(ID_{FA}, bP, T_{FA})$, where T_{FA} is FA 's chosen timestamp, and K_{FH} is a pre-shared secret known to FA and HA . The message $m_2 = \{ID_{FA}, MID, Auth_{MU}, T_{FA}, D_{FA}\}$ is sent to HA .

A3. HA receives the message m_2 , decrypts MID and D_{FA} to obtain ID_{MU} , T_{FA} . HA verifies the correctness of ID_{MU} and T_{FA} . If the verification doesn't hold, HA rejects the process. Else, HA calculates $V' = h(ID_{MU} \parallel K)$ using HA 's master key and decrypts $Auth_{MU}$ to fetch aP . A random number n' is generated by HA to compute $MID' = Enc_K(ID_{MU}, n')$, $D_{HA} = Enc_{K_{FH}}(ID_{HA}, ID_{MU}, aP, bP)$ and $Auth_{HA} = Enc_{V'}(MID', aP, bP)$, where T_{HA} is a timestamp of HA . Finally, the message $m_3 = \{D_{HA}, Auth_{HA}\}$ is sent to FA .

A4. FA receives the message m_3 and decrypts D_{HA} and looks for the correctness of bP and T_{HA} . If the correctness holds, FA computes $Auth_{FA} = h(abP \parallel ID_{MU} \parallel ID_{FA})$, $SK_{MF} = h(aP \parallel bP \parallel abP \parallel ID_{MU} \parallel ID_{FA})$. The message $m_4 = \{Auth_{FA}, Auth_{HA}\}$ is sent to MU .

A5. To obtain aP , bP and MID' , MU decrypts $Auth_{HA}$. MU verifies whether the decrypted aP is same as that of the value in step A1. If the verification holds with the chosen secret value a , then MU needs to verify the correctness of $Auth_{FA}$ to compute the session key $SK_{MF} = h(aP \parallel bP \parallel abP \parallel ID_{MU} \parallel ID_{FA})$. The smart-card updates MID with MID' in its memory.

2.3. Update Session Key

Interested readers can refer to [34].

3. Security Pitfalls of Zhang *et al.*'s Scheme

3.1. Adversary Model

The security of Zhang *et al.*'s scheme is analyzed under the following security model [7-8, 16, 20]:

- 1 Over the public channel, Adversary/Attacker (\mathcal{A}) has the ability to eavesdrop all the communica-

tions between the parties.

- 2 \mathcal{A} attains the potential to delete, modify, resend or to redirect the eavesdropped transmitted messages.
- 3 By analyzing the method of power analysis and consumption or from the leaked information, the information from the smart card can be extracted by \mathcal{A} .
- 4 \mathcal{A} can be an insider of the system.

3.2. Disadvantages:

- 1 No verification mechanism for user's login credentials to check the legitimacy of the user.
- 2 The communication message m_1 is transmitted to FA even if illegal credentials (Wrong credentials) are being used by a user/adversary.
- 3 Incorrect login credentials induce huge communication and computational wastage.

3.3. Stolen Smartcard Attack

Suppose that the smartcard of MU is lost or stolen for a deliberate amount of time and replaced. The adversary \mathcal{A} can get control over the smartcard as discussed in Section 3.1. Using this captured smartcard parameters and the transmitted messages on the insecure channel, \mathcal{A} can perform the following attacks:

3.3.1. Password Guessing Attack

- 1 Initially, \mathcal{A} guesses the PW_{MU}^A and computes $V^A = h(PW_{MU}^A || m)$, $V' = V^A \oplus C$.
- 2 \mathcal{A} decrypts $Auth_{MU}$ and $Auth_{HA}$ using the computed V' .
- 3 \mathcal{A} verifies the parametric value of αP in both $Auth_{MU}$ and $Auth_{HA}$ from the above decryption. If the value is the same in both the decryption computations, \mathcal{A} guesses the password of the MU successfully.
- 4 Otherwise, \mathcal{A} repeats the above steps until the password PW_{MU} is guessed correctly.

Therefore, \mathcal{A} will be able to guess the MU 's password.

3.3.2. User Anonymity and Traceability

According to Subsection 3.3.1, adversary guesses the password of MU and computes V' . Further \mathcal{A} decrypts $Auth_{MU}$ using V' . It is evident that the decryption of $Auth_{MU}$ discloses the identity ID_{MU} of the MU . Thus, ID_{MU} allows \mathcal{A} to differentiate the users in every different login sessions which may breach the privacy of the user. Therefore, we claim that privacy of the user was not taken proper care as referred in [34].

3.4. User Impersonation Attack

We have seen that \mathcal{A} is able to compute V' (i.e., $V' = h(ID_{MU} || K)$) successfully. Using the adversary capabilities described in Section 3.1 and V' , \mathcal{A} modifies $Auth_{MU}$ using his selected random number (say $\alpha \in Z_p^*$). The details are as follows:

- 1 \mathcal{A} computes αP and $Auth_A = Enc_{V'}(ID_{MU}, \alpha P)$ and transmits the message $m_1 = \{ID_{HA}, MID, Auth_A\}$ to FA .
- 2 FA receives the message m_1 and selects a random number $b \in Z_p^*$ computes bP and $D_{FA} = Enc_{K_{FH}}(ID_{FA}, bP, T_{FA})$. The message $m_2 = \{ID_{FA}, MID, Auth_A, T_{FA}, D_{FA}\}$ is sent to HA .
- 3 On receiving m_2 , HA decrypts MID and D_{FA} to obtain ID_{MU} and T_{FA} . HA verifies the correctness of ID_{MU} and T_{FA} . If verification fails, HA rejects the process. Else, \mathcal{A} makes use of HA 's master key K , computes $V' = h(ID_{MU} || K)$ and decrypts $Auth_A$ to fetch $\alpha P, T_1$. Then a random number n' is chosen from Z_p^* to compute $D_{HA} = Enc_{K_{FH}}(ID_{HA}, ID_{MU}, \alpha P, bP, T_1, T_{HA})$ and $Auth_{HA} = Enc_{V'}(MID', \alpha P, T_1, bP)$. Finally, the message $m_3 = \{D_{HA}, Auth_{HA}\}$ is sent to FA .
- 4 FA receives the message m_3 and decrypts D_{HA} and looks for the correctness of bP and T_{HA} . If the correctness holds, FA computes $Auth_{FA} = h(\alpha bP || ID_{MU} || ID_{FA})$, $SK_{MF} = h(\alpha P || bP || \alpha bP || ID_{MU} || ID_{FA})$. The message $m_4 = \{Auth_{FA}, Auth_{HA}\}$ is sent to MU .
- 5 \mathcal{A} captures the message m_4 .

The above procedure indicates that the adversary can successfully impersonate MU by making both FA and HA believe that they are communicating with MU .

3.5. Replay Attack

The adversary \mathcal{A} can capture the previously communicated messages as described in Section 3.1. \mathcal{A} uses the communicated messages and replays the same message (say m_1) to FA and the same is transmitted to HA via FA as there is no fresh verification of the messages sent by MU at HA . Therefore, Zhang *et al.*'s scheme does not prevent \mathcal{A} from sending the replay messages.

3.6. Absence of Unauthorized Login Detection

A user may sometime fail to correctly map different password to his/ her different accounts. Thus, login credential verification should be supported at initial

stage. However, Zhang *et al.*'s scheme lacks user credential verification mechanism in login phase. Lack of efficient login phase makes the scheme inefficient.

4. The Proposed Scheme

In proposed scheme, $K_H \in Z_p^*$ is the secret key of HA and $K_F \in Z_p^*$ is the secret key of FA . FA and HA also share a common secret key K_{FH} . Furthermore, FA and HA also compute their public keys $K_H P$ and $K_F P$, respectively.

Table 1

Login, authentication and key agreement phase of our scheme

MU	FA	HA
Inputs ID_{MU} and PW_{MU} . Computes $m^* = L_i \oplus h(H(Bio) \ ID_{MU}^*)$, Checks if $V_i \stackrel{?}{=} h(PW_{MU}^* \oplus m^* \oplus ID_{MU}^*)$ Computes $Y_i = h(PW_{MU}^* \ m)$, $X_i = Z_i \oplus Y_i^*$, Chooses a random number $\alpha \in Z_p^*$. Computes $\alpha K_F P$, αP , $DID_{MU} = ID_{MU} \oplus h(\alpha K_F P)$, $A_1 = h(ID_{MU} \ \alpha K_F P \ ID_{HA} \ T_1)$	$m_1 = \{ID_{HA}, DID_{MU}, \alpha P, A_1, T_1\}$ Checks if $ T_2 - T_1 < \Delta T$ Computes $K_F \alpha P$, $ID_{MU} = DID_{MU} \oplus h(\alpha K_F P)$ Verifies $A_1 \stackrel{?}{=} h(ID_{MU} \ K_F \alpha P \ ID_{HA} \ T_1)$ Chooses a random number $\beta \in Z_p^*$ Computes βP . $D_{FA} = Enc_{K_{FH}}(ID_{MA}, \beta P, \alpha P, T_2)$, $A_2 = h(ID_{MU} \ \beta K_H P \ K_{FH} \ D_{FA} \ T_2)$	$m_2 = \{ID_{FA}, D_{FA}, A_2\}$ Decrypts D_{FA} to obtain $ID_{MA}, \beta P, \alpha P, T_2$ Verifies $ T_3 - T_2 < \Delta T$ and $A_2 \stackrel{?}{=} h(ID_{MU} \ K_H \beta P \ K_{FH} \ D_{FA} \ T_2)$ Chooses a random number $\gamma \in Z_p^*$ Computes $X_i = h(ID_{MU} \ K_H)$, $h(X_i \ \gamma)$, $D_{HA} = Enc_{K_{FH}}(h(X_i \ \gamma), T_3, \gamma)$, $A_3 = h(h(X_i \ \gamma) \ \beta P \ K_{FH} \ D_{HA} \ T_3)$,
	$m_3 = \{D_{HA}, A_3\}$ Decrypts D_{HA} to obtain $h(X_i \ \gamma), T_3, \gamma$, Verifies $ T_4 - T_3 < \Delta T$ and $A_3 \stackrel{?}{=} h(h(X_i \ \gamma) \ \beta P \ K_{FH} \ D_{HA} \ T_3)$ Computes $\beta \alpha P$, $S K_{MF} = h(ID_{MU} \ \beta \alpha P \ \alpha K_F P \ h(X_i \ \gamma) \ T_1 \ T_4)$ $A_4 = h(ID_{FA} \ S K_{MF} \ T_4 \ \gamma)$	
$m_4 = \{A_4, \gamma \oplus h(\alpha K_F P), \beta P, T_4\}$ Verifies $ T_5 - T_4 < \Delta T$ Computes γ from $\gamma \oplus h(\alpha K_F P)$ Computes $\alpha \beta P, h(X_i \ \gamma)$ $S K_{MF} = h(ID_{MU} \ \alpha \beta P \ \alpha K_F P \ h(X_i \ \gamma) \ T_1 \ T_4)$, Verifies $A_4 \stackrel{?}{=} h(ID_{FA} \ S K_{MF} \ T_4 \ \gamma)$		

4.1. Registration Phase

This phase utilizes the secure channel to communicate with the participants. MU registers with HA , if he/she wishes to get services from Global Mobility Network.

- MU selects his/her identity ID_{MU} , password PW_{MU} , computes $Y_i = h(PW_{MU} \| m)$ and transmits the message $\{ID_{MU}, Y_i\}$ to home agent, where $m \in Z_p^*$ is a random number.
- HA receives the message $\{ID_{MU}, Y_i\}$ and

generates a random number n . HA performs computations using its master key K . HA computes $X_i = h(ID_{MU} \| K_H)$, $Z_i = Y_i \oplus X_i$. Further, HA stores $\{Z_i, ID_{HA}\}$ into the smartcard SC and sends to MU .

- 3 MU imprints his/her biometric Bio and computes $V_i = h(PW_{MU} \oplus m \oplus ID_{MU})$, $L_i = h(H(Bio) \| ID_{MU}) \oplus m$. MU stores L_i, V_i in SC .

4.2. Login, Authentication and Key Establishment Phase

In this phase, the communication between FA and HA is done through the secure channel. The registered MU opts services from FA . Before this, both the parties agree upon a common session key which is described as follows:

- 1 MU inputs his/her login credentials into the smartcard to compute $m^* = L_i \oplus h(H(Bio) \| ID_{MU}^*)$. MU verifies $V_i \stackrel{?}{=} h(PW_{MU}^* \oplus m^* \oplus ID_{MU}^*)$. If verification holds, MU computes $Y_i = h(PW_{MU}^* \| m)$, and $X_i = Z_i \oplus Y_i^*$. MU randomly selects $\alpha \in Z_p^*$ and timestamp T_1 and computes $\alpha K_F P$, αP , $DID_{MU} = ID_{MU} \oplus h(\alpha K_F P)$, $A_1 = h(ID_{MU} \| \alpha K_F P \| ID_{HA} \| T_1)$. MU sends the message $m_1 = \{ID_{HA}, DID_{MU}, \alpha P, A_1, T_1\}$ to FA .
- 2 Once FA receives the message m_1 at time T_2 , FA checks $|T_2 - T_1| < \Delta T$. If the verification fails, FA rejects the query. Otherwise, FA computes $K_F \alpha P$, and $ID_{MU} = DID_{MU} \oplus h(\alpha K_F P)$. FA verifies $A_1 = h(ID_{MU} \| K_F \alpha P \| ID_{HA} \| T_1)$. If verification holds, FA selects $\beta \in Z_p^*$ randomly and computes βP and $D_{FA} = Enc_{K_{FH}}(ID_{MA}, \beta P, \alpha P, T_2)$, where K_{FH} is a pre-shared secret held between FA and HA . FA sends the message $m_2 = \{ID_{FA}, D_{FA}, A_2\}$ to HA , where $A_2 = h(ID_{MU} \| \beta K_H P \| K_{FH} \| D_{FA} \| T_2)$.
- 3 Upon receiving the message m_2 from FA at time T_3 , HA decrypts D_{FA} to obtain $ID_{MA}, \beta P, \alpha P, T_2$. HA verifies $|T_3 - T_2| < \Delta T$ and $A_2 = h(ID_{MU} \| K_H \beta P \| K_{FH} \| D_{FA} \| T_2)$. If the verification does not hold, HA rejects the process. Otherwise, HA selects $\gamma \in Z_p^*$ and computes $X_i = h(ID_{MU} \| K_H)$ and $h(X_i \| \gamma)$. HA sends the message $m_3 = \{D_{HA}, A_3\}$ to FA , where $D_{HA} = Enc_{K_{FH}}(h(X_i \| \gamma), T_3, \gamma)$ and $A_3 = h(h(X_i \| \gamma) \| \beta P \| K_{FH} \| D_{HA} \| T_3)$.
- 4 FA receives the message m_3 at time T_4 , FA decrypts D_{HA} to obtain $h(X_i \| \gamma), T_3, \gamma$. FA looks for the correctness of $A_3 \stackrel{?}{=} h(h(X_i \| \gamma) \| \beta P \| K_{FH} \| D_{HA} \| T_3)$ and $|T_4 - T_3| < \Delta T$. The process termi-

nates FA , if the verification fails. Otherwise, FA computes $\beta \alpha P$, the session key $SK_{MF} = h(ID_{MU} \| \beta \alpha P \| \alpha K_F P \| h(X_i \| \gamma) \| T_1 \| T_4)$ and $A_4 = h(ID_{FA} \| SK_{MF} \| T_4 \| \gamma)$. FA sends the message $m_4 = \{A_4, \gamma \oplus h(\alpha K_F P), \beta P, T_4\}$ to MU .

- 5 On receiving the message m_4 at time T_5 , MU verifies $|T_5 - T_4| < \Delta T$. If verification holds, MU computes γ from $\gamma \oplus h(\alpha K_F P)$. MU computes $\alpha \beta P, h(X_i \| \gamma)$ and $SK_{MF} = h(ID_{MU} \| \alpha \beta P \| \alpha K_F P \| h(X_i \| \gamma) \| T_1 \| T_4)$. The mutual authentication succeeds if the verification $A_4 = h(ID_{FA} \| SK_{MF} \| T_4 \| \gamma)$ holds.

The illustration of this phase is shown in Table 3.

4.3. Update Session Key

Assume that MU stays in the FA network zone for consecutive sessions. In this scenario, this would be inefficient for MU to perform re-authentication procedure with FA and by involving HA to attain a session key. So, in this case, without communicating with HA , both MU and FA undergo one-time authentication where they can update the session key using the earlier established session key.

U1: Suppose that a session key $SK_{MF_{i-1}}$ between MU and FA was already established. If MU wants to establish another session with FA , MU generates $\alpha_i \in Z_p^*$ and computes $\alpha_i P$. Further, $\alpha_i P$ is sent to FA .

U2: Upon receiving $\alpha_i P$, FA selects β_i and computes $\alpha_i \beta_i P$ and $SK_{MF_i} = h(\alpha_i \beta_i P)$. FA computes $TempSK_{MF_i} = f_{SK_{MF_i}}(\alpha_{i-1} \beta_{i-1} P \| \alpha_i \beta_i P)$ and sends $\beta_i P$ and $TempSK_{MF_i}$ to MU .

U3: Upon receiving the parameters, MU computes $SK'_{MF_i} = h(\alpha_i \beta_i P)$, and verifies the correctness of $TempSK'_{MF_i} \stackrel{?}{=} f_{SK'_{MF_i}}(\alpha_{i-1} \beta_{i-1} P \| \alpha_i \beta_i P)$. If this verification holds, both the parties update the session key as $SK_{MF_i} = h(\alpha_i \beta_i P)$.

5. Formal Security Analysis of Our Scheme

We first present the security model and algorithm assumptions that are used in proving our scheme. We present the formal security analysis by the method of provable security [9, 32].

5.1. Security Model

The adaptability of provable security is to evaluate the invincibility of our scheme against the well-known attacks.

The participants are mobile user $MU \in \mathcal{MU}$, home agent $HA \in \mathcal{HA}$ and foreign agent $FA \in \mathcal{FA}$.

Adversary Capabilities

Let \mathcal{A} controls the simulator and queries oracles to destroy the privacy of authentication or the session keys. The dictionary \mathcal{D} size is a fixed constant which does not change upon the security parameter \mathcal{K} which \mathcal{A} tries to destroy in PPT. The security parameter \mathcal{K} is the session key bit-length. On the following queries, \mathcal{A} performs simulation in the oracles:

Execute($\Pi_{MU}^i, \Pi_{HA}^k, \Pi_{FA}^j$): It denotes that the adversary queries an execution of the protocol between the instances $\Pi_{MU}^i, \Pi_{HA}^k, \Pi_{FA}^j$ by eavesdropping and gets the access. This query model is for the passive attacks against the protocol.

Encryption/decryption($\Pi_{MU}^i, m, text$): By applying encryption query, an input message m is encrypted to a ciphertext $text$ as output. Furthermore, by applying decryption query, the cipher-text is decrypted and results in an output message m .

Send(Π_E^i, m): The active attacks in the channel are carried out by the transmitted messages between the instances Π_{MU}^i and Π_{FA}^j which are prone to dictionary attacks, man-in-the-middle attacks, impersonation attacks, and unknown key-share attacks. Π_E^i sends a message m to the requested partner. If the message m is valid, the query is accepted by the simulator. Otherwise, the session is rejected. An interaction with *Send*($\Pi_{MU}^i, start : \langle HA, FA \rangle$) indicates that Π_{MU}^i initiates a session with instances of HA and FA .

Reveal(Π_E^i): This query model is for known-key attacks. It outputs a terminator \perp , if the oracle has not been accepted. Otherwise, it outputs a session key SK_E^i .

CorruptSC(\mathcal{MU})/*CorruptLL*(\mathcal{MU}): The lost smartcard problem and the threat of smartcard breach are handled by this query model. The attacker \mathcal{A} imposes offline password guessing by eavesdropping on messagees.

CorruptLL(Π_E^i): This query is in correspondence to strong forward security. The attacker \mathcal{A} obtains all the information of Π_E^i . We list the possible queries for \mathcal{A} as follows:

CorruptLL(Π_{MU}^i): \mathcal{A} gets all information from the smartcard with PW .

CorruptLL(Π_{HA}^i): This query model is for the privileged insider attacks.

CorruptVFR(Π_{HA}^i): In this query model, the passwords which are stored by HA can be prone to stolen verifier attacks.

CorruptLL(Π_{FA}^j): This query model for the long-lived secrets of the foreign agent FA . This can be done by modeling agent node capture attacks.

TestAKE(Π_E^i): This query model gives the session key. A target session is chosen by \mathcal{A} to challenge after multiple queries. If no session key is found for instance SK_E^i , it outputs \perp . Otherwise, a coin namely, b is flipped. If $b = 1$, the session key for instance, SK_E^i is returned. Otherwise, it outputs a random string of the same size. For $MF - AKE - fresh$ instance, it can be queried once. $MF - fresh$ is introduced below.

We specify few definitions to illustrate our proof as shown below in Figures 1 and 2. We have given the simulation of the queries in the interest of readers:

1 **Partnering:** MU and FA creates the session key. We call MU and FA as partners if and only if $sid_{MU} = sid_{FA}$, $pid_{MU} = FA$, $pid_{FA} = MU$, and $SK_{MU} = SK_{FA}$ are accepted by them.

2 **MF-AKE-fresh:** (This shows the freshness of strong forward security) This notion is defined only for MU and FA . We say that Π_E^i is $MF-AKE-fresh$ if the following queries do not occur:

- *AReveal*(Π_E^i) appears.
- *AReveal*($pid_{\Pi_E^i}$) appears.
- Before *Test* happens, *Corrupt*(Π_E^i) or *Corrupt*($pid_{\Pi_E^i}$) has been asked.

3 **Security:** The adversary \mathcal{A} 's advantage against our scheme Π is the probability that \mathcal{A} correctly guesses the bit b generated in *Test*(Π_E^i) query with $MF - AKE - fresh$ Π_E^i is accepted. The advantage of \mathcal{A} is

$$Adv_{\Pi}^{MF-AKE}(\mathcal{A}) = 2Pr[b = b'] - 1.$$

Our scheme is $MF - AKE$ secure if $Adv_{\Pi}^{MF-AKE}(\mathcal{A})$ is negligibly greater than $O(q_{send})/|\mathcal{D}|$, depending on security parameters l_h , l_r and l_n . Here q_{send} is the query time of *Send*(Π_E^i, m), l_h , l_r and l_n are the length of hash results, length of random numbers and length of parameter n , respectively.

Figure 2

Simulation of queries

<p>On querying a hash function $h(m)$, if there exists a record (m,n) in L_h, an output n is returned else, the simulator chooses a random bit string $n \in \{0, 1\}^l$, and responses as n and puts (m, n) into L_h</p> <p>For initiate a $Send(MU, INIT)$ query, the following steps were done by the simulator: Computes $m^* = L_A \oplus h(ID_{MU}^* PW_{MU}^*)$, $V^* = h(PW_{MU}^* ID_{MU}^* m^*)$ and verifies $A \stackrel{?}{=} h(V^* ID_{MU}^*)$. If this verification is false, reverts the query. Computes $S_{HM} = V \oplus C$, selects $\alpha \in Z_p^*$ random number and T_1 Computes αP and $Auth_{MU} = Enc_{S_{HM}}(ID_{MU}, \alpha P, T_1)$. Returns $m_1 = \{ID_{HA}, MID, Auth_{MU}, T_1\}$ as the answer.</p> <p>For a $Send(MU, FA^j, m_1)$ query, the following steps were done by the simulator: Checks $T_{FA} - T_1 < \Delta T$. If the verification fails, rejects the query. Selects $\beta \in Z_p^*$ random number, Computes βP and $D_{FA} = Enc_{K_{FH}}(ID_{FA}, \beta P, T_{FA})$. Then answers the query with $m_2 = \{ID_{FA}, MID, Auth_{MU}, D_{FA}\}$. For a $Send(FA^j, HA^t, m_2)$ query, the following steps were done by the simulator: Decrypt MID and D_{FA} to obtain ID_{MU} and T_{FA}. and verifies the correctness of ID_{MU}, T_{FA}. Computes $S_{MH} = h(ID_{MU} K)$, and decrypts $Auth_{MU}$ to extract $\alpha P, T_1$. Selects a random number $n' \in Z_p^*$ and computes $MID' = Enc_K(ID_{MU}, n' \oplus n)$ $D_{HA} = Enc_{K_{FH}}(ID_{HA}, ID_{MU}, \alpha P, \beta P, T_1, ID_{FA}, T_{HA})$ and $Auth_{HA} = Enc_{S_{MH}}(MID', ID_{FA}, \alpha P, T_1, \beta P, T_{HA})$. Then answers the query with $m_3 = \{D_{HA}, Auth_{HA}\}$.</p> <p>For a $Send(HA^t, FA^j, m_3)$ query, the following steps were done by the simulator: Decrypts D_{HA} and verifies the correctness of βP and T_{HA}. Computes $Auth_{FA} = h(T_{HA} \alpha \beta P ID_{MU} ID_{FA})$, $SK_{MF} = h(ID_{MU} \alpha P \beta P \alpha \beta P ID_{FA})$. Then answers the query with $m_4 = \{Auth_{FA}, Auth_{HA}\}$. For a $Send(FA^j, MU, m_4)$ query, the following steps were done by the simulator: Decrypts $Auth_{HA}$ to extract $\alpha P, \beta P, ID_{FA}, T_{HA}$ and MID'. and verifies the correctness of αP. Computes $Auth_{FA} \stackrel{?}{=} h(T_{HA} \alpha \beta P ID_{MU} ID_{FA})$. If the verification holds computes the session key $SK_{MF} = h(ID_{MU} \alpha P \beta P \alpha \beta P ID_{FA})$</p> <p>To perform the $Execute(MU, FA^j, HA^t)$ query, the successive computation of all $Send$ queries are done to return the messages (m_1, m_2, m_3, m_4).</p> <p>To perform the $Reveal(I^k)$ query, if the instance I^k has been accepted and formed a session key, returns sk_{MU} or sk_{FA} otherwise a \perp is the answer.</p> <p>To perform the $Corrupt(SC)$ query, the U^s smartcard stored information is returned.</p> <p>To perform the $Corrupt(I^k)$ query, all the stored information of I^k is returned.</p> <p>To perform the $Test(I^k)$ query, if the verification I^k is not equal to $MF - fresh$, returns \perp. else, a coin b is tossed. The session key is returned, if the outcome is $b = 1$. A random string of length l will be returned, if the outcome is $b = 0$.</p>

4 Elliptic Curve Diffie-Hellman (ECDH) assumption: Let P be a generator of G and, mP and nP be two elements of G , where $m, n \in Z_q^*$ and G is an additive group of prime order q . If \mathcal{A} is successful in computing mnP from (mP, nP) , we denote it as $Adv_A^{ECDH}(t)$ and which can also be considered as the maximal success probability among the adversary which runs within time t . The ECDH assumption holds if $Adv_A^{ECDH}(t)$ is negligible.

5.2. MF - AKE Security

Theorem 1. Consider an elliptic curve with group G and let $|D_{PW}|$ be the dictionary size of the password. Let our proposed scheme be π . The advantage of an

adversary is considered if \mathcal{A} can compute the following within a specific time bound t by taking less than send-queries time (q_{send}), execute-queries time (q_{exec}), hash-queries time (q_{hash}), and encryption/decryption-queries time ($q_{E/D}$):

$$\begin{aligned}
 Adv_{\pi}^{MF-AKE}(\mathcal{A}) &\leq \frac{2q_{send}}{|D_{PW}|} + \frac{(q_{E/D}^2 + q_{send} + q_{exec})^2}{2(q-1)} \\
 &+ Adv_A^{ECDH}(t + (1 + q_{send} + q_{exec}) \cdot \tau_G)^{(1)}, \\
 &+ \frac{q_{hash}^2}{2^l} + \frac{2(q_{send}(Bio) + q_{send} + q_{exec})}{2^l}
 \end{aligned}$$

where, τ_G denotes the time to compute the point mul-

tiplication in G and l is a security parameter string of $\{0,1\}^l$.

Proof: The communication among $\pi_{i,j}^s$ and $\pi_{j,i}^t$ is done by considering the definitions [1-3] which are intended to be conveyed faithfully, i.e., both $\pi_{i,j}^s$ and $\pi_{j,i}^t$ will receive the fairly established message by which both the oracles can agree upon a session key. Since, $\alpha, \beta \in Z_p^*$, the session key $SK_{MF} = h(ID_{MU} \parallel \alpha\beta P \parallel \alpha K_{FP} \parallel h(X_i \parallel \gamma) \parallel T_1 \parallel T_4)$ is considered to be random for every established session which is from the key space.

Now, to prove that our scheme meets the second and third conditions. A consecutive sequence of games $G_i (0 \leq i \leq 5)$ is played which starts at G_0 and ends at G_5 . For each game, if \mathcal{A} correctly guesses the bit c by posing the *Test* query, then it is considered as $Succ_i$. Thus, the *Test* query is done on

$$Diff_i = |Pr[Succ_i] - Pr[Succ_{i-1}]|, \forall (1 \leq i \leq 5),$$

which is the difference of probability of success between G_i and G_{i-1} .

Once \mathcal{A} finishes the last game G_5 , the *MF - AKE Security* game can only be won with probability $1/2$. The detail description of the games are as follows:

Game G_0 : In this game, \mathcal{A} is facilitated with several oracles such as *E/D* oracle, hash oracle. Even the participants' instances $\pi_{i,j}^s$ are also available to \mathcal{A} . Thus, by definition of Adv_{π}^{MF-AKE} in Section 3.1:

$$Adv_{\pi}^{MF-AKE}(\mathcal{A}) = 2.Pr[Succ_0] - 1.$$

Our goal is to substantiate that $Adv_{\pi}^{MF-AKE}(\mathcal{A})$ is negligible.

Thus, we have

$$Adv_{\pi}^{MF-AKE}(\mathcal{A}) = 2.Pr[Succ_5] - 1 + 2.\sum_{i=1}^5 Diff_i.$$

Game G_1 : In this game, the *E/D* oracle and hash oracle are simulated by maintaining a hash list and an encryption/decryption list as $List_h, List_{E/D}$. $List_h$ is of the form $\langle x, h(x) \rangle \forall x$ where x is the input value and $h(x) \in \{0,1\}^l$ is an output random value. In addition, considering all the queries such as (*Send, Execute, Reveal, and Test*), all the instances are simulated as in the real protocol. The result of the simulation is indistinguishable from the real attack unless the following events take place:

Event 1: The permutation property of encryption/decryption fails which means there will be some collisions on the encryption/decryption.

Event 2: The hash function is prone to collisions.

Figure 3

Simulation of *Send* queries

To perform the *Execute*(MU^i, FA^j, G^k) query, the successive computation of all *Send* queries are done to return the messages (m_1, m_2, m_3, m_4) .

To perform the *Reveal*(I^k) query, if the instance I^k has been accepted and formed a session key, returns sk_{MU} or sk_{FA} . Otherwise a \perp is the answer.

To perform the *Corrupt*(SC) query, the U 's smartcard stored information is returned.

To perform the *Corrupt*(I^k) query, all the stored information of I^k is returned.

Since \mathcal{A} knows some information at first,

the concrete results are as follows:

(a) $I = MU$: \mathcal{A} can retrieve all the information of user's smartcard with password PW .

(b) $I = HA$: \mathcal{A} can get the secret key K .

(c) $I = FA$: \mathcal{A} can obtain P .

To perform the *Test*(I^k) query, if the verification I^k is not equal to *MF - AKE - fresh*, returns \perp .

Here I is MU or FA , like *Corrupt*(I^k). Else, a coin b is tossed by the simulator.

The session key is returned, if the outcome is $b = 1$.

A random string of length l will be returned, if the outcome is $b = 0$.

Thus, the probability of the two events is restricted to

$$\frac{q_h^2}{2^{l+1}} + \frac{q_{E/D}}{2(q-1)} \text{ i.e.,}$$

$$Diff_1 = |Pr[Succ_1] - Pr[Succ_0]|$$

$$\leq \frac{q_h^2}{2^{l+1}} + \frac{q_{E/D}}{2(q-1)}.$$

Game G_2 : This game stops when a collision occurs on the transcripts $(ID_{HA}, DID_{MU}, *, A_1, T_1), (ID_{MU}, *)$, $(ID_{MU}, *, ID_{HA}, T_1)$. Otherwise, \mathcal{A} undergoes the simulations where all the oracles in game G_1 function. Suppose the collisions occur this imply (this case is similar to replay attack) that \mathcal{A} can easily win the game. According to the birthday paradox, the probability of this event is restricted to:

$$Diff_2 = |Pr[Succ_2] - Pr[Succ_1]|$$

$$\leq \frac{(q_{send} + q_{exec})^2}{2(q-1)}.$$

Game G_3 : This game stops if \mathcal{A} is successful in guessing the correct authenticate values A_1, A_2 without checking the corresponding hash oracles with the queries. Otherwise, \mathcal{A} performs the simulations on all the oracles in game G_2 . Thus G_3 and G_2 are indistinguishable unless the oracle rejects a valid authentication value. The probability of this event is restricted to:

$$Diff_3 = |Pr[Succ_3] - Pr[Succ_2]|$$

$$\leq \frac{q_{send} + q_{exec}}{2^l}.$$

Game G_4 : This game stops if \mathcal{A} is successful in guessing the password and biometric key of MU . Then, using the guessed password and biometric key, \mathcal{A} can get the secret value V_i and further choose αP as a component of the Diffie-Hellman tuple. As a result, \mathcal{A} can correctly distinguish the random value and the session key. Otherwise, \mathcal{A} simulates all the oracles in game G_3 . To show the effectiveness of the attack, \mathcal{A} should continuously attempt to guess the password of MU . So the probability of this event is restricted to $\frac{q_{send}}{|D_{PW}|} + \frac{q_{send(Bio)}}{2^l}$, i.e.,

$$Diff_4 = |Pr[Succ_4] - Pr[Succ_3]|$$

$$\leq \frac{q_{send}}{|D_{PW}|} + \frac{q_{send(Bio)}}{2^l}$$

Game G_5 : This game is the same as **Game G_4** except that the executions are simulated using the random self-reducibility of the Elliptic Curve Diffie-Hellman (ECDH) problem which means one ECDH instance $(\alpha P, \beta P)$ is given to compute $\alpha\beta P$ with unknown α and β . Firstly, a test session is chosen. When the adversary \mathcal{A} chooses the test session, we choose and win the authenticated key exchange game simulated by us. We can solve the ECDH problem by using \mathcal{A} as the subroutine. More precisely, we choose $\pi_{MU,FA}^s$ as the test session and $\pi_{FA,MU}^s$ as the matching session of it. For example, $\pi_{MU,FA}^s(send, receive) = \{(ID_{HA}, DID_{MU}, *, A_1, T_1)\}$, $\pi_{FA,MU}^s(send, receive) = \{(ID_{HA}, DID_{MU}, *, A_1, T_1), (ID_{MU}, *), (ID_{MU}, *, ID_{HA}, T_1)\}$. When \mathcal{A} asks the *Execute* query or *Send* query to $\pi_{MU,FA}^s$, we simulate the protocol and send $(ID_{HA}, DID_{MU}, A_1, T_1)$ to \mathcal{A} , where $A_1 = h(ID_{MU} \| \alpha K_F P \| ID_{HA} \| T_1)$. We also imbed αP into the protocol and substitute αP in game G_4 for αP . Now a random oracle is used to output a random value as the values α and β are unknown. Therefore, we cannot compute $\alpha\beta P, \alpha K_F P$ as well as the session key. Suppose the output of the random oracle is h , then we let the session key $SK_{MF} = h(ID_{MU} \| \alpha\beta P \| \alpha K_F P \| h(X_i \| \gamma) \| T_1 \| T_4)$. In such case, if \mathcal{A} has computed the real session key himself, i.e., he has computed $SK_{MF} = h(ID_{MU} \| \alpha\beta P \| \alpha K_F P \| h(X_i \| \gamma) \| T_1 \| T_4)$, we can obtain $\alpha\beta P$ and solve the ECDH problem by using \mathcal{A} as a subroutine. Now we see, the games G_5 and G_4 are indistinguishable unless \mathcal{A} has queried the random oracles h on $SK_{MF} = h(ID_{MU} \| \alpha\beta P \| \alpha K_F P \| h(X_i \| \gamma) \| T_1 \| T_4)$. We bound the probability of this event by $q_h Adv_A^{ECDH}(t)$. Then we have:

$$Diff_5 = |Pr[Succ_5] - Pr[Succ_4]|$$

$$\leq q_h Adv_A^{ECDH}(t + (1 + q_{exec} + q_{send}) \cdot \tau_G).$$

It is clear from the above game G_5 , there is no collisions on the hash function and *encryption / decryption* query and no collision on the transcripts of the instances. Therefore, \mathcal{A} does not guess the password correctly and is unable to solve the ECDH problem. Therefore, in random oracle model, all sessions are independent of each other. The adversary \mathcal{A} cannot get any advantage in game G_5 . Hence, we have: $Pr[Succ_5] = 1/2$.

From definition 2,3 definition 4, consequently from the aforementioned equations, one can get the result

of Theorem 1:

$$q_h Adv_A^{ECDH}(t + (1 + q_{exec} + q_{send}) \cdot \tau_G)$$

is negligible because $Adv_A^{ECDH}(t)$ (the probability of breaking ECDH problem) is negligible;

q is a large prime and l is the security parameter. The time complexity of q_{send} , q_{exec} and q_{hash} are considered to be executed in polynomial time. Therefore,

$$\frac{2q_{send} + (q_{E/D}^2 + q_{send} + q_{exec})^2}{|D_{PW}|} + \frac{q_{hash}^2 + 2(q_{send(Bio)} + q_{send} + q_{exec})}{2^l}$$

is negligible.

Hence the proof.

6. Discussions on Some Attacks

6.1. Stolen Smartcard Attack

In this case, either the smartcard of the user is stolen or found by an attacker. Then the information stored in the smartcard $\{ID_{HA}, Z_i, L_i, V_i\}$ can be extracted by the power analysis attack [7-8, 16, 20] as described in the adversary model. In order to login and enjoy the services of FA , a valid user ID_{MU} , $H(Bio)$ and PW_{MU} should be provided by the attacker. However, we show that the login credentials cannot be obtained by an attacker. The details are as follows:

a Offline Password/Biometric Key Guessing Attack

Using the offline method, an attacker wants to guess the user's password/biometric key. But due to the known fact that the user's biometric key cannot be guessed, \mathcal{A} tries to guess user's password PW_{MU} which was used in the computation of $m = L_i \oplus h(H(Bio) || ID_{MU})$, and $V = h(PW_{MU} \oplus m \oplus ID_{MU})$. Therefore, to guess the password correctly, the attacker has to correctly guess the login credentials ID_{MU} and PW_{MU} concurrently, but the probability of guessing ID_{MU} of length exactly l bits and PW_{MU} of length exactly n characters correctly at the same time is approximately $\frac{1}{2^{6n+l}}$ which is negligible. Thus, it is not possible to guess the user's password/biometric key in polynomial time. Hence, our proposed scheme withstands the offline password/biometric key guessing attack.

b User Anonymity

The user's real identity ID_{MU} is preserved by mask-

ing it with the master secret key of GWN. Therefore, when a user enters the roaming region with different FA networks, MU provides the pseudo-identity DID_{MU} to FA . The user's identity is encrypted with the elliptic point $\alpha K_r P$, thereby it is evident that only the participants can decrypt and obtain the user's real identity.

c User's Traceability

When a user enters the roaming region with different FA networks, MU is allowed to provide his/her pseudo-identity which was transmitted during the login request m_1 . The pseudo-identity varies each time the user tries to get access FA . This pseudo-identity works as a temporary identity of the MU and can be dynamically updated as and when the MU visits different FAs . Thus, user traceability cannot be achieved.

6.2. Privileged Insider Attack

The password, PW_{MU} of MU is not sent as plaintext. MU computes and sends $Y_i = h(PW_{MU} || m)$ by using a random $m \in Z_p^*$. Therefore, an insider at HA fails to obtain MU 's password from a registration request phase. In addition, due to the non-invertible nature of one-way hash function retrieving PW_{MU} from Y_i is computationally infeasible. Therefore, our proposed scheme resists the insider attack.

6.3. Replay Attack

As described in our proposed scheme, every transmitted message is included with a built-in timestamp of MU , and FA . The legal participants could figure out the replay attack by checking the freshness of the incoming message. Thus our proposed scheme can resist the replay attack.

6.4. Impersonation Attack

In the roaming authentication, we need to consider three scenarios of impersonation attack, i.e., on MU , on visiting FA , and finally on HA . The details are as follows:

- 1 MU : Suppose, the adversary \mathcal{A} wishes to enjoy the services from FA on behalf of MU . Firstly, \mathcal{A} must overcome the authentication process by HA . It is evident that \mathcal{A} does not possess MU 's secret $\alpha K_r P$. Therefore, a trial to forge A_1 fails to pass the authentication process by FA . Thus, this shows that \mathcal{A} cannot impersonate MU .

2 *FA/HA*: It is observed that, in our proposal, each βP value computed by *FA* is sent to *HA* which cannot be extracted by \mathcal{A} as the communication between *FA* and *HA* is over a secure channel. Furthermore, the same βP value is sent to *MU* by masking the elliptic point and compute D_{FA} and A_2 using the secret keys K_{FH} and $\beta K_H P$. Thus, to impersonate *FA/HA*, \mathcal{A} needs to know or extract the secret keys. As the key K_{FH} is kept secret to *FA* and *HA*, extracting the shared secret key by \mathcal{A} is computationally infeasible. Moreover, \mathcal{A} does not know *HA*'s master key K_H , so S_{MH} cannot be computed. Thus, \mathcal{A} fails to impersonate *HA* to *MU* or *HA* to *FA*.

6.5. Perfect Forward Secrecy

Suppose \mathcal{A} wants to compute the session key $SK_{MF} = h(ID_{MU} || \alpha\beta P || \alpha K_H P || h(X_i || \gamma) || T_1 || T_4)$ so as to communicate with the roaming network for timely communication. However, in order to compute the session key, \mathcal{A} needs to compute $\alpha\beta P$ and $\alpha K_H P$ although αP and βP are known which is an ECDH problem. Thus, the session key is not acquired. As a result, the forward secrecy is provided in the proposed scheme.

6.6. Fair Key Agreement

Our proposed scheme ends up with *MU* and *FA* agreeing on shared session key SK_{MF} where both the parties ensure equal contribution. i.e., the session key $SK_{MF} = h(ID_{MU} || \alpha\beta P || \alpha K_H P || h(X_i || \gamma) || T_1 || T_4)$ is computed using the random points with individual identities produced by *MU* and *FA*, respectively. Such as, $\alpha\beta P$, ID_{MU} and $\alpha K_H P$ are the random number points with identities produced by *MU* and *FA*. This clearly signifies the equal contribution of the participants (*MU/FA*). In this way, our key agreement protocol can assure the fairness property.

6.7. Mutual Entity Authentication

Our proposed scheme supports mutual authentication among *MU*, *FA* and *HA*.

- 1 *FA* authenticates *MU* by checking the validity of the timestamp T_1 and ID_{MU} by applying verification on $A_1 = h(ID_{MU} || \alpha K_H P || ID_{HA} || T_1)$.
- 2 *FA* computes βP and D_{FA} by encrypting with the shared key K_{FH} . Once the mes-

sage $m_3 = \{D_{HA}, Auth_{HA}\}$ is received, *FA* decrypts D_{HA} and validates βP by verification on $A_3 = h(h(X_i || \gamma) || \beta P || K_{FH} || D_{HA} || T_3)$ is right. If this verification holds, *FA* authenticates *HA* directly and computes the session key SK_{MF} and A_4 .

- 3 *MU* verifies the authenticity of $A_4 = h(ID_{FA} || SK_{MF} || T_4 || \gamma)$ by computing SK_{MF} . If this verification holds, *MU* authenticates *FA* and *HA*, respectively.

Table 2

Comparison of security features

Security attributes	Our	[34]	[26]	[30]	[13]	[22]
S_1	√	×	√	×	√	√
S_2	√	√	×	√	×	×
S_3	√	×	×	×	√	×
S_4	√	×	√	×	√	√
S_5	√	×	×	×	×	×
S_6	√	×	√	√	×	×
S_7	√	√	√	×	√	×
S_8	√	×	√	√	√	√
S_9	×	×	√	√	×	×
S_{10}	√	√	×	×	×	√
S_{11}	√	×	×	×	√	×
S_{12}	√	√	×	×	×	√

Note: √ = preserved; × = not preserved.

S_1 : Resists password guessing attack; S_2 : Protects privileged-insider attack; S_3 : Provides user anonymity; S_4 : Resilient against stolen smartcard attack; S_5 : Secure against impersonation attack; S_6 : Protects replay attack; S_7 : Provides proper mutual authentication; S_8 : Provides two-factor security; S_9 : *HA* knows the session key; S_{10} : Provides perfect forward secrecy; S_{11} : Restricts user traceability; S_{12} : Discloses session key.

7. Performance Comparison with Related Schemes

In this section, the performance and functionality features of the proposed scheme are compared with the existing similar authentication schemes proposed for the GLOMONET.

7.1. Security Features Comparison

In Table 3, the security features provided and protected by our proposed scheme are compared with the existing similar schemes [13, 22, 26, 30, 34]. As compared to existing similar schemes, our proposed scheme protects various known attacks and also supports various good features.

7.2. Performance Analysis

In Table 4, according to [24-25, 35], the time consumed by the cryptographic one-way hash function $T_h = 0.0023ms$ and the symmetric encryption/decryption operations $T_\Omega = 0.0046ms$ were considered. In Table 5, the computational time cost and communication overheads required during the login and authentication phases are compared with the existing similar schemes. The output of the one-way hash function $h(\cdot)$ is 128 bits, if SHA-1 hashing algorithm

[3] and symmetric encryption with 256 bits is used. Further, we assume that each timestamp, random nonce/random number, identity of (MU , FA , HA)'s is 160 bits in length.

The key points are as follows:

- From Fig. 6, we are able to show that our scheme takes more storage space in comparison to [13], but takes less storage space than [22, 26, 30, 34].
- Although our scheme takes more computation cost as compared to [26] and nearly equivalent computation cost as compared to [22, 34], but has better computation cost as compared to [13, 30].
- From Fig. 6, we are able to illuminate that the communication cost of the proposed scheme is higher than [26], but less as compared to [13, 22, 30, 34].
- The most important part is the security. The proposed scheme has better security features by overcoming the drawbacks as compared to existing similar schemes. We have also presented our scheme with a formal verification.

Therefore, Table 3, Table 5 and Fig. 6 show that the proposed scheme is better as compared to existing similar schemes in security, computation cost and communication cost.

Table 3

Computation cost analysis [24-25, 35]

Notation	Description	≈ execution time
T_h	One-way hash function	0,0023 ms
T_Ω	Symmetric key encryption/decryption	0,0046 ms
T_{EM}	Elliptic curve point multiplication	2,2260 ms
T_M	1024-bit modular exponentiation	3,8500 ms

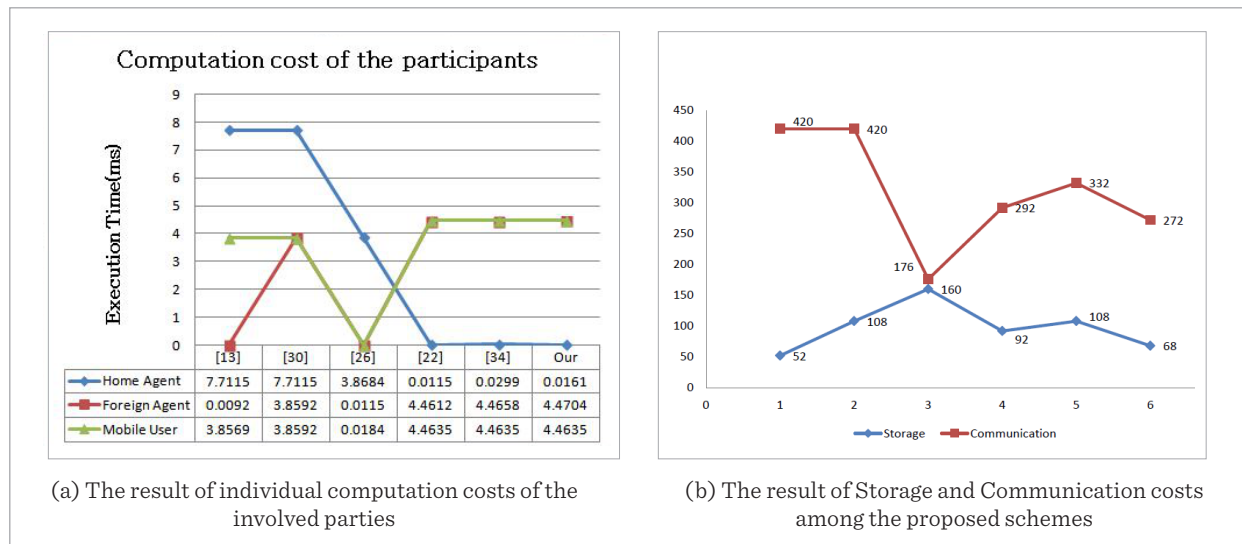
Table 4

Performance comparison among our scheme and other schemes

Scheme	[13]	[30]	[26]	[22]	[34]	Our	
Storage in Smartcard(bytes)	52	108	160	92	108	68	
Computation cost and time for Login and Authentication	MU :	$3T_h + T_M$	$4T_h + T_M$	$8T_h$	$5T_h + 2T_{EM}$	$1T_h + 2T_\Omega + 2T_{EM}$	$5T_h + 2T_{EM}$
	FA :	$4T_h$	$4T_h + T_M$	$1T_h + 2T_\Omega$	$4T_h + 2T_{EM}$	$2T_h + 2T_\Omega + 2T_{EM}$	$4T_h + 2T_\Omega + 2T_{EM}$
	HA :	$5T_h + 2T_M$	$5T_h + 2T_M$	$4T_h + 2T_\Omega + 1T_M$	$5T_h$	$1T_h + 6T_\Omega$	$3T_h + 2T_\Omega$
	<i>Total (ms)</i> :	$12T_h + 3T_M$ ≈ 11.5776	$13T_h + 4T_M$ ≈ 15.4299	$13T_h + 4T_\Omega + 1T_M$ ≈ 3.8983	$14T_h + 4T_{EM}$ ≈ 8.9362	$4T_h + 10T_\Omega + 4T_{EM}$ ≈ 8.9592	$12T_h + 4T_\Omega + 4T_{EM}$ ≈ 8.95
Communication cost(bytes)	420	420	176	292	332	272	
Formal verification	No	No	No	No	Yes	Yes	
Security	No	No	No	No	No	Yes	

Figure 4

Simulation result for our proposed scheme



8. Conclusion

In this paper, we have reviewed the well designed Zhang *et al.*'s authentication scheme for GLOMONET, and presented various security pitfalls which include password guessing attack, impersonation attack, replay attack, user anonymity and traceability attack. To overcome the security weaknesses of Zhang *et al.*'s scheme, we have designed an authentication scheme for GLOMON-

ETs. The security analysis has been done using standard formal provable security proof. The informal cryptanalysis proofs to the resilience of relevant security attacks have been presented. The analysis of proposed scheme demonstrates that the proposed scheme addresses both security and privacy challenges. Results prove that the proposed scheme is efficient.

References

- Alizadeh, M., Abolfazli, S., Zamani, M., Baharun, S., Sakurai, K. Authentication in Mobile Cloud Computing: A Survey. *Journal of Network and Computer Applications*, 2016, 61, 59-80. <https://doi.org/10.1016/j.jnca.2015.10.005>
- Astorga, J., Aguado, M., Toledo, N., Higuero, M. A High Performance Link Layer Mobility Management Strategy for Professional Private Broadband Networks. *Journal of Network and Computer Applications*, 2013, 36(4), 1152-1163. <https://doi.org/10.1016/j.jnca.2013.01.005>
- Burrows, J. H. Secure Hash Standard. FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995. <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>. Accessed on July 2015.
- Chen, Y.-C., Chuang, S.-C., Yeh, L.-Y., Huang, J.-L. A Practical Authentication Protocol with Anonymity for Wireless Access Networks. *Wireless Communications and Mobile Computing*, 2011, 11(10), 1366-1375. <https://doi.org/10.1002/wcm.933>
- Chen, C., He, D., Chan, S., Bu, J., Gao, Y., Fan, R. Lightweight and Provably Secure User Authentication with Anonymity for the Global Mobility Network. *International Journal of Communication Systems*, 2011, 24(3), 347-362. <https://doi.org/10.1002/dac.1158>
- Chang, C.-C., Tsai, H.-C. An Anonymous and Self-Verified Mobile Authentication with Authenticated Key Agreement for Large-Scale Wireless Networks. *IEEE Transactions on Wireless Communications*, 2010, 9(11), 3346-3353. <https://doi.org/10.1109/TWC.2010.092410.090022>

7. Dolev, D., Yao, A. C. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 1983, 29(2), 198-208. <https://doi.org/10.1109/TIT.1983.1056650>
8. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Shalmani, M., Manzuri, T. On the Power of Power Analysis in the Real World: A Complete Break of the Keeloq Code Hopping Scheme. In *Advances in Cryptology - CRYPTO-2008, Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, 5157, 203-220.
9. Farash, M. S., Chaudhry, S. A., Heydari, M., Sadough, S., Mohammad, S., Kumari, S., Khan, M. K. A Lightweight Anonymous Authentication Scheme for Consumer Roaming in Ubiquitous Networks with Provable Security. *International Journal of Communication Systems*, 2015, 30(4), 1-20.
10. Gope, P., Hwang, T. An Efficient Mutual Authentication and Key Agreement Scheme Preserving Strong Anonymity of the Mobile User in Global Mobility Networks. *Journal of Network and Computer Applications*, 2016, 62, 1-8. <https://doi.org/10.1016/j.jnca.2015.12.003>
11. He, D., Kumar, N., Khan, M. K., Lee, J.-H. Anonymous Two-Factor Authentication for Consumer Roaming Service in Global Mobility Networks. *IEEE Transactions on Consumer Electronics*, 2013, 59(4), 811-817. <https://doi.org/10.1109/TCE.2013.6689693>
12. He, D., Zhang, Y., Chen, J. Cryptanalysis and Improvement of an Anonymous Authentication Protocol for Wireless Access Networks. *Wireless Personal Communications*, 2014, 74(2), 229-243. <https://doi.org/10.1007/s11277-013-1282-x>
13. Jiang, Q., Ma, J., Li, G., Yang, L. An Enhanced Authentication Scheme with Privacy Preservation for Roaming Service in Global Mobility Networks. *Wireless Personal Communications*, 2013, 68(4), 1477-1491. <https://doi.org/10.1007/s11277-012-0535-4>
14. Jin, A. T. B., Ling, D. N. C., Goh, A. Biobhashing: Two Factor Authentication Featuring Fingerprint Data and Tokenised Random Number. *Pattern Recognition*, 2004, 37(11), 2245-2255. <https://doi.org/10.1016/j.patcog.2004.04.011>
15. Kim, J.-S., Kwak, J. Improved Secure Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks. *International Journal of Security and Its Applications*, 2012, 6(3), 45-54.
16. Kocher, P., Jaffe, J., Jun, B. Differential Power Analysis. In *Proceedings of Advances in Cryptology - CRYPTO'99, Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1999, 1666, 388-397. https://doi.org/10.1007/3-540-48405-1_25
17. Kumar, V., Jangirala, S., Ahmad, M. An Efficient Mutual Authentication Framework for Healthcare System in Cloud Computing. *Journal of Medical Systems*, 2018, 42(8), 1-25. <https://doi.org/10.1007/s10916-018-0987-5>
18. Lee, C.-Y., Chang, C.-C., Lin, C.-H. User Authentication with Anonymity for Global Mobility Networks. In *2nd International Conference on Mobile Technology, Applications and Systems*, IEEE, 2005, 1-5.
19. Lee, C.-C., Hwang, M.-S., Liao, I.-E. Security Enhancement on a New Authentication Scheme with Anonymity for Wireless Environments. *IEEE Transactions on Industrial Electronics*, 2006, 53(5), 1683-1687. <https://doi.org/10.1109/TIE.2006.881998>
20. Messerges, T. S., Dabbish, E. A., Sloan, R. H. Examining Smart-Card Security Under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, 2002, 51(5), 541-552. <https://doi.org/10.1109/TC.2002.1004593>
21. Modares, H., Moravejosharieh, A., Lloret, J., Salleh, R. A Survey of Secure Protocols in Mobile IPv6. *Journal of Network and Computer Applications*, 2014, 39, 351-368. <https://doi.org/10.1016/j.jnca.2013.07.013>
22. Mun, H., Han, K., Lee, Y. S., Yeun, C. Y., Choi, H. H. Enhanced Secure Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks. *Mathematical and Computer Modelling*, 2012, 55(1), 214-222. <https://doi.org/10.1016/j.mcm.2011.04.036>
23. Nanni, L., Brahnam, S., Lumini, A. Biobhashing Applied to Orientation-Based Minutia Descriptor for Secure Fingerprint Authentication System. *Electronics Letters*, 2011, 47(15), 851-853. <https://doi.org/10.1049/el.2011.1525>
24. Niinuma, K., Park, U., Jain, A. K. Soft Biometric Traits for Continuous User Authentication. *IEEE Transactions on Information Forensics and Security*, 2010, 5(4), 771-780. <https://doi.org/10.1109/TIFS.2010.2075927>
25. Odelu, V., Das, A. K., Goswami, A. An Efficient Biometric-Based Privacy-Preserving Three-Party Authentication with Key Agreement Protocol Using Smart Cards. *Security and Communication Networks*, 2015, 8(18), 4136-4156. <https://doi.org/10.1002/sec.1330>
26. Shin, S., Yeh, H., Kim, K. An Efficient Secure Authentication Scheme with User Anonymity for Roaming User in Ubiquitous Networks. *Peer-to-peer Networking and Applications*, 2015, 8(4), 674-683. <https://doi.org/10.1007/s12083-013-0218-2>
27. Srinivas, J., Mishra, D., Mukhopadhyay, S. A Mutual Authentication Framework for Wireless Medical Sen-

- sor Networks. *Journal of Medical Systems*, 2017, 41(5), 1-19. <https://doi.org/10.1007/s10916-017-0720-9>
28. Srinivas, J., Mukhopadhyay, S., Mishra, D. Secure and Efficient User Authentication Scheme for Multi-Gateway Wireless Sensor Networks. *Ad Hoc Networks*, 2017, 54, 147-169. <https://doi.org/10.1016/j.adhoc.2016.11.002>
29. Suzuki, S., Nakada, K. An Authentication Technique Based on Distributed Security Management for the Global Mobility Network. *IEEE Journal on Selected Areas in Communications*, 1997, 15(8), 1608-1617. <https://doi.org/10.1109/49.634798>
30. Wen, F., Susilo, W., Yang, G. A Secure and Effective Anonymous User Authentication Scheme for Roaming Service in Global Mobility Networks. *Wireless Personal Communications*, 2013, 73(3), 993-1004. <https://doi.org/10.1007/s11277-013-1243-4>
31. Wu, C.-C., Lee, W.-B., Tsaur, W.-J. A Secure Authentication Scheme with Anonymity for Wireless Communications. *IEEE Communications Letters*, 2008, 12(10), 722-723. <https://doi.org/10.1109/LCOMM.2008.080283>
32. Wu, F., Xu, L., Kumari, S., Li, X. A Privacy-Preserving and Provable User Authentication Scheme for Wireless Sensor Networks based on Internet of Things Security. *Journal of Ambient Intelligence and Humanized Computing*, 2016, 1-16.
33. Xie, Q., Hu, B., Tan, X., Bao, M., Yu, X. Robust Anonymous Two-Factor Authentication Scheme for Roaming Service in Global Mobility Network. *Wireless Personal Communications*, 2014, 74(2), 601-614. <https://doi.org/10.1007/s11277-013-1309-3>
34. Zhang, G., Fan, D., Zhang, Y., Li, X., Liu, X. A Privacy Preserving Authentication Scheme for Roaming Services in Global Mobility Networks. *Security and Communication Networks*, 2015, 8(16), 2850-2859. <https://doi.org/10.1002/sec.1209>
35. Zhang, Q., Yin, Y., Zhan, D.-C., Peng, J. A Novel Serial Multimodal Biometrics Framework Based on Semisupervised Learning Techniques. *IEEE Transactions on Information Forensics and Security*, 2014, 9(10), 1681-1694. <https://doi.org/10.1109/TIFS.2014.2346703>
36. Zhu, J., Ma, J. A New Authentication Scheme with Anonymity for Wireless Environments. *IEEE Transactions on Consumer Electronics*, 2004, 50(1), 231-235. <https://doi.org/10.1109/TCE.2004.1277867>