symmetry

MDPI

Article

A Privacy Preserving Authentication Scheme for Roaming in IoT-Based Wireless Mobile Networks

Bander A. Alzahrani [1,*], Shehzad Ashraf Chaudhry [2], Ahmed Barnawi [1], Abdullah Al-Barakati [1] and Mohammed H. Alsharif [3,*]

1 Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; ambarnawi@kau.edu.sa (A.B.); aaalbarakati@kau.edu.sa (A.A.-B.)
2 Department of Computer Engineering, Faculty of Engineering and Architecture Istanbul Gelisim University Istanbul, Avcılar, 34310 Istanbul, Turkey; sashraf@gelisim.edu.tr
3 Department of Electrical Engineering, College of Electronics and Information Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Korea
* Correspondence: baalzahrani@kau.edu.sa (B.A.A.); malsharif@sejong.ac.kr (M.H.A.)

Received: 16 January 2020; Accepted: 10 February 2020; Published: 15 February 2020

Abstract: The roaming service enables a remote user to get desired services, while roaming in a foreign network through the help of his home network. The authentication is a pre-requisite for secure communication between a foreign network and the roaming user, which enables the user to share a secret key with foreign network for subsequent private communication of data. Sharing a secret key is a tedious task due to underneath open and insecure channel. Recently, a number of such schemes have been proposed to provide authentication between roaming user and the foreign networks. Very recently, Lu et al. claimed that the seminal Gopi-Hwang scheme fails to resist a session-specific temporary information leakage attack. Lu et al. then proposed an improved scheme based on Elliptic Curve Cryptography (ECC) for roaming user. However, contrary to their claim, the paper provides an in-depth cryptanalysis of Lu et al.'s scheme to show the weaknesses of their scheme against Stolen Verifier and Traceability attacks. Moreover, the analysis also affirms that the scheme of Lu et al. entails incorrect login and authentication phases and is prone to scalability issues. An improved scheme is then proposed. The scheme not only overcomes the weaknesses Lu et al.'s scheme but also incurs low computation time. The security of the scheme is analyzed through formal and informal methods; moreover, the automated tool ProVerif also verifies the security features claimed by the proposed scheme.

Keywords: roaming user; authentication; internet of things; mobile networks; anonymity; elliptic curve cryptography; ProVerif

1. Introduction

The emerging Internet of Things (IoT) is an infrastructure of all globally connected devices, including home appliances, vehicles, mobiles, tablets, surveillance systems, smart grids, etc. The IoT facilitate the heterogeneity of networks to seamlessly communicate with each other. The roaming service in IoT-based networks enables a remote user to enjoy seamless and scuffle free services during roaming outside the home network. A typical roaming scenario is shown in Figure 1. Involving three entities, namely mobile user, home network, and foreign network, the mobile user, using his digital communication device, like smart-phone, smart-vehicle, Laptop, PDA, etc., can access the services of his home network remotely in the coverage area of a foreign network. The roaming service extends the handover of connections from home network to foreign network, when both the networks belong to different types and are located at different geographical locations. The home and foreign network enter

into a roaming agreement in order to facilitate their users. The user registers himself with the home network and, when he roams out of the coverage of his home network and enters into the coverage range of another network (foreign network having roaming agreement with home network), can access and enjoy the services of his home network through the foreign network. The roaming service is getting importance rapidly, due to millions of subscribers traveling abroad per year. The main issue restricting wide usage of roaming services is the security and privacy of the connecting parties. All the services provided are subject to communicate through an open/insecure wireless channel, causing an inherited effect on the security of such networks. The roaming process requires proper security mechanisms and is equally important for the three participants because the foreign networks cannot allow the user's resources and services to be used illegitimately and without payment, whereas the home network avoids becoming a source of illegal access to foreign network, and the user does not want to be charged for the services used by some adversary. Moreover, as per user's perspective, privacy and anonymity has gotten much importance. Without privacy and anonymity, the adversary can track user movements and current location [1,2]. The proper countering of security-related issues requires the development of customized authentication protocol, in which the authentication protocols not only verify the authenticity of the communicating parties but also ensure a session key for subsequent confidential data/services extended between the participating entities. The authentication is required when a user roams out of the coverage area of his home network and enters into the coverage area of a foreign network. The user has to get authenticated by the foreign network by the help of his home network. The successful authentication process can ensure that the access to the network is limited to legitimate users only [3].

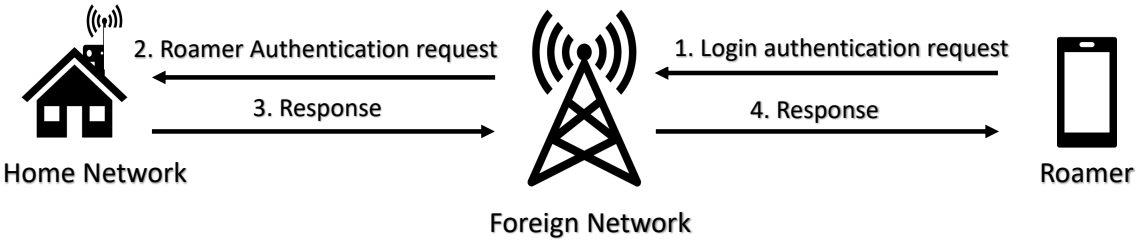

Figure 1. Roaming user authentication.

In recent years, various authentication protocols were proposed [4–20] based on different cryptographic mechanisms. The schemes [15–18] are based on lightweight symmetric key primitives, as per the criteria laid down by Wang and Wang [21], the symmetric key mechanisms cannot provide privacy except for keeping a very large number of pseudo identities in smart-card with low memory or getting dynamic identity from home network at each login request. The schemes [4–7,12–14] based on bilinear pairing/modular exponentiation operations consume much more computation and in turn drains more battery power of already limited power wireless/mobile devices. Some of such schemes [8–11] are based on public but still low resource sucker Elliptic Curve Cryptography (ECC).

In 2009, Chang et al. [17] proposed an authentication scheme to secure GLOMONET. However, soon it was realized by Youn et al. [22] that the scheme proposed in Reference [17] could not achieve user anonymity. In 2012, Mun et al. [8] proposed and ECC based authentication scheme for roaming user on the principles of EC Diffie–Hellman problem (ECDHP). Soon after Mun et al.'s proposal, Reddy et al. [9] and Kim et al. [23] found various weaknesses in Mun et al.'s scheme, including insecurity against replay attacks. Reddy et al. [9] then proposed a slightly modified version to resist replay and other attacks against Mun et al.'s scheme. In 2017, another symmetric key based scheme for GLOMONET was proposed by Chaudhry et al. [18]. However, authors in Reference [24] found various weaknesses, including vulnerability to impersonation and related attacks in Chaudhry et al.'s scheme [18]. The scheme proposed by Lee et al. [24] is susceptible to traceability attack, as the dynamic identity is sent by the home agent during the session in plain text and this plain text dynamic identity

sent through open channel can be used to trace future login requests. Recently, Gope and Hwang [25] proposed an authentication scheme for roaming user in GLOMONET using pseudo identity to counter DoS attack. Very recently in 2019, Lu et al. [26] pointed out various weaknesses in Gopi-Hwang's scheme, including its insecurity against known session-specific parameters in leakage attacks. Moreover, Lu et al. claimed the Password Renewal Phase of Gopi-Hwang as faulty, and they proposed an ECC based new scheme.

1.1. The Contributions

Quite recently, in 2019, Lu et al. [26] found some weaknesses in Gopi-Hwang [25] authentication scheme for roaming users. To combat, Lu et al. proposed a new roaming user authentication scheme using ECC and claimed that their proposal extends required security features and resists known attacks. Contrary to their [26] claim, the cryptanalysis in this article shows that the roaming scheme presented in Reference [26] cannot protect the remote user against Stolen Verifier and Traceability attacks. Moreover, the analysis also affirms that the scheme of Lu et al. entails incorrect login and authentication phases and is prone to scalability issues. Therefore, an improved scheme based on ECC is designed by just modifying some of the steps in Lu et al.'s proposal. The scheme not only overcomes the weaknesses of Lu et al.'s scheme but also incurs low computation time. The proposed scheme entails following merits:

- The scheme provides provable security under the hardness of ECDLP (elliptic-curve discrete logarithm and elliptic-cure deffie-Hellman problems.
- The scheme provides security and anonymity under automated security model of ProVerif.
- The scheme provides authentication among user and foreign network with the help of home network.
- The scheme achieves low computation power as compared with baseline scheme presented in Reference [26].

1.2. Security Requirements

The user friendly security requirements for a roaming user authentication scheme are as follows:

1. The mobile roaming user should have facility to change his password credentials in an easy manner and he should be facilitated not to memorize a complicated and/or long password.
2. Along with traditional security requirements, The scheme should ensure user privacy and anonymity. Any insider/outsider, including foreign agents, should remain unaware regarding the original identity of the roaming user. Moreover, current location of the user should not be exposed to anyone with some previous knowledge.
3. Home network should facilitate the authentication process between user and foreign network.
4. The authentication should result into a shared secret key among user and foreign network for subsequent confidential communication over insecure link.
5. The scheme should at least resist all known attacks.

1.3. Adversarial Model

The common model for adversary capabilities, as mentioned in Reference [27–31], is adopted and explained below:

1. Adversary ($\mathcal{MU}_a$) fully controls the link and can listen, modify, replay a message from all the legal communicating parties. $\mathcal{MU}_a$ is also able to inject a self created false message.
2. $\mathcal{MU}_a$ can easily get identity related information.
3. $\mathcal{MU}_a$ knows all public parameters.
4. Being an insider, $\mathcal{MU}_a$ can extract verifier table stored in home network database.

5. Home Network's private key is considered as secret and no other entity can extract the key.
6. The pre-shared key between home and foreign networks is assumed to be secure.

2. Review of the Scheme of Lu et al.

A brief review of Lu et al.'s roaming user authentication scheme is explained here. Before moving further, please refer to Table 1 for understanding the notations used in this paper. The three main phases of Lu et al.'s scheme are detailed in below subsections:

Table 1. Notations.

Notation	Definition
$\mathcal{MU}_x$,$\mathcal{HA}_z$, $\mathcal{FA}_y$	Mobile Node, Home Network, foreign Network
ID_{mx} ID_{hz}, ID_{fy}	Identities of $\mathcal{MU}_x$,$\mathcal{HA}_z$ and $\mathcal{FA}_y$
PW_{mx}, PWU_{hz}	Password and concealed password of $\mathcal{MU}_x$
K_{xz}, K_{yz}	Shared keys between $\mathcal{MU}_x$, $\mathcal{HA}_z$ and $\mathcal{FA}_y$, $\mathcal{HA}_z$
$E_p(a,b)$, P	Elliptic curve and a base point over curve
S_h, $P_h = S_hP$	Private and public key pair of $\mathcal{HA}_z$
E_k/D_k	Symmetric Encryption/decryption
$h()$, $H()$	Two one-way hash Functions
$()_x$, $\oplus$	x-coordinate of a EC point, Exclusive-OR
Mac_k	Key based Mac

2.1. Home Network Agent Setup Phase

For system-setup purposes, Home Network Agent $\mathcal{HA}_z$ selects an Elliptic curve $E_p(a,b) : y^2 = x^3 + ax + b \mod p$, where a, $\in F_p$ a finite field, such that $4a^3 + 27b^2 \neq 0$, along with an infinite point O. HA then selects a base point P over $E_p(a,b)$. $\mathcal{HA}_z$ selects a secret key S_h and computes public key $P_h = S_hP$. $\mathcal{HA}_z$ also selects irreversible Hash and keyed MAC functions $h()$, $H()$, $Mac_k()$, along with symmetric encryption/decryption algorithms $E_k()$, $D_k()$.

2.2. Registration Phase

Step LRP1: The mobile user $\mathcal{MU}_x$ selects identity/password pair $\{ID_{mx}, PW_{mx}\}$, along with r_{mx} (generated randomly), and computes $PWU_{hz} = h(PW_{mx}, r_{mx})$. $\mathcal{MU}_x$ sends the pair $\{ID_{mx}, PWU_{hz}\}$ to $\mathcal{HA}_z$.

Step LRP2: Upon reception of $\{ID_{mx}, PWU_{hz}\}$ to $\mathcal{HA}_z$ pair from $\mathcal{MU}_x$, $\mathcal{HA}_z$ generates random x_1, x_2 and r_{mx} and stores ID_{mx} and a sequence number $SNum_{mx}$ against i^{th} registration request of $\mathcal{MU}_x$. $\mathcal{HA}_z$ then computes $PID_{mx} = h(h(ID_{mx}, x_1), x_2)$, $K_{xz} = h(PID_{mx}, S_h)$, $\alpha_{hz} = E_{PWU_{hz}}(K_{xz})$, and $\beta_{hz} = h(h(ID_{mx}), PWU_{hz})$. $\mathcal{HA}_z$ then sends a *smart-card* containing $\{\alpha_{hz}, \beta_{hz}, PID_{mx}\}$ to $\mathcal{MU}_x$. $\mathcal{HA}_z$ stores K_{xz} in a verifier table maintained by $\mathcal{HA}_z$.

Step LRP3: Upon reception of *smart-card*, $\mathcal{MU}_x$ inserts r_{mx}. Finally, the *smart-card* contains: $\{\alpha_{hz}, \beta_{hz}, PID_{mx}, r_{mx}, h(), H(), E_k, D_k, Mac_k, P\}$.

2.3. Login & Authentication Phase

Step LLA1: After inserting smart-card, $\mathcal{MU}_x$ inputs ID_{mx} and PW_{mx}, the smart-card computes $PWU_{hz} = h(PW_{mx}, r_{mx})$ and verifies $h(h(ID_{mx}), h(r_{mx}, PWU_{hz})) \stackrel{?}{=} \beta_{hz}$. Terminates the session if verification is unsuccessful. Otherwise, generates time-stamp T_1, random N_{mx} and computes $K_{xz} = D_{PWU_{hz}}(\alpha_{hz})$, $A_{mx} = N_{mx}P + H(K_{xz}, ID_{mx}, ID_{hz})P$, $B_{mx} = E_{K_{xz}}(ID_{mx}, T_1, PID_{mx})$ and $C_{mx} = Mac_{K_{xz}}(N_{mx}P, ID_{mx}, T_1)$. $\mathcal{MU}_x$ sends $M_{uf1} = \{A_{mx}, B_{mx}, C_{mx}, PID_{mx}, T_1\}$ to $\mathcal{FA}_y$.

Step LLA2: $\mathcal{FA}_y$ upon reception of request, checks freshness of T_1 and generates fresh time-stamp T_2, random N_{fy}. $\mathcal{FA}_y$ then computes $A_{fy} = N_{fy}P + H(K_{yz}, ID_{fy}, T_2)P$, $B_{fy} = Mac_{(N_{fy}P)_x}(ID_{hz}, T_1)$ and sends $M_{fh2} = \{M_{uf1}, A_{fy}, B_{fy}, T_2\}$ to $\mathcal{HA}_z$.

Step LLA3: $\mathcal{HA}_z$ verifies freshness of T_2 after receiving message from $\mathcal{FA}_y$. Rejects the message, if T_2 is not fresh. Otherwise, $\mathcal{HA}_z$ based on PID_{mx} extracts corresponding shared key K_{xz}

from verifier database and decrypts B_{mx} to get ID_{mx}. $\mathcal{HA}_z$ verifies originality of ID_{mx} by comparing with the once stored in verifier in a tuple consisting of ID_{mx}, PID_{mx} and K_{xz}. Upon successful verification, $\mathcal{HA}_z$ computes $N_{mx}P = A_{mx} - H(K_{xz}, ID_{mx}, ID_{hz})P$ and verifies whether $C_{mx} \stackrel{?}{=} Mac_{K_{xz}}(N_{mx}P, ID_{mx}, T_1)$. Upon successful verification, $\mathcal{HA}_z$ computes $N_{fy}P = A_{fy} - H(K_{yz}, ID_{fy}, T_2)P$ and then checks $B_{fy} \stackrel{?}{=} Mac_{(N_{fy}P)_x}(ID_{hz}, T_1)$. On success, $\mathcal{HA}_z$ updates $K_{yz} = K_{yz} \oplus h(ID_{fy}, N_{fy}P, T_3)$ and computes $A_{hz} = N_{mx}P + H(ID_{mx})P + H(K_{yz}, ID_{hz}, N_{fy}P)P$, $B_{hz} = Mac_{K_{yz}}(N_{fy}P, N_{mx}P + H(ID_{mx}P, T_3))$. $\mathcal{HA}_z$ also updates $K_{xz} = K_{xz} \oplus h(ID_{mx}, N_{mx}P, T_3)$ and computes $C_{hz} = N_{fy}P + H(K_{xz}, ID_{hz}, N_{mx}P)P$, $D_{hz} = Mac_{K_{xz}}(ID_{fy}, N_{fy}P, T_3, PID_{mx})$. HA then sends $M_{hf3} = \{A_{hz}, B_{hz}, C_{hz}, D_{hz}, T_3\}$ to $\mathcal{FA}_y$ and increments $SNum_{mx}$.

Step LLA4: $\mathcal{FA}_y$ checks freshness of T_3 after receiving response of $\mathcal{HA}_z$. On success, $\mathcal{FA}_y$ computes $N_{mx}P + H(ID_{mx})P = A_{hz} - H(K_{yz}, ID_{hz}, N_{fy}P)p$. $\mathcal{FA}_y$ then verifies validity of B_{hz} and on success, computes $C_{fy} = Mac_{(N_{mx}P+H(ID_{mx}P))_x}(ID_{fy}, N_{fy}P, T_3, T_4, C_{mx})$.The session key is computed as $SK = h(N_{fy}(N_{mx}P + H(ID_{mx})P))$. Then, $\mathcal{FA}_y$ sends $M_{fu4} = \{C_{fy}, C_{hz}, D_{hz}, T_3, T_4\}$ to $\mathcal{MU}_x$.

Step LLA5: Upon reception, $\mathcal{MU}_x$ verifies freshness of T_3 and T_4 and on success, computes $N_{fy}P = C_{hz} - H(K_{xz}, ID_{hz}, N_{mx}P)P$. $\mathcal{MU}_x$ further checks validity of D_{hz} and C_{fy}, if both holds, $\mathcal{MU}_x$ computes session key $SK = h((N_{mx} + H(ID_{mx}))N_{fy}P)$, $D_{mx} = Mac_{N_{mx}+H(ID_{mx})P_x}(C_{fy}, N_{fy}P)$ and sends $M_{uf5} = \{D_{mx}, T_5\}$ to $\mathcal{FA}_y$.

Step LLA6: $\mathcal{FA}_y$ verifies freshness of T_5 and checks validity of D_{mx}. If it holds, $\mathcal{FA}_y$ treats $\mathcal{MU}_x$ as legitimate user and now further communication between $\mathcal{FA}_y$ and $\mathcal{MU}_x$ may be carried out using the shared key $SK = h(N_{fy}(N_{mx}P + H(ID_{mx})P))$.

3. Cryptanalysis of the Scheme of Lu et al.

In this section, cryptanalysis of the Lu et al.'s scheme is accomplished, under the realistic assumptions made in the adversarial model of Section 1.3. The following subsections show that the scheme of Lu et al. carries severe weaknesses, including in security against Stolen Verifier and known Session Specific variables attacks. Moreover, the scheme does not provide untraceability and has scalability issues. More seriously, the scheme also entails correctness issues, such incorrectness may stop authentication process before completion and legitimate user may experience denial of services. The following subsections explain the weaknesses:

3.1. Stolen Verifier Attack

Let $\mathcal{MU}_a$ be a dishonest insider and based on his capabilities, as mentioned in Section 1.3, can steal the verifier table with tuples $\{ID_{mx}, PID_{mx}, K_{xz}\}$. Using the verifier parameters, $\mathcal{MU}_a$ can impersonate as any roaming mobile user registered with home agent. The attack is simulated as follows:

Step IA1: $\mathcal{MU}_a$ generates time-stamp T_{a1}, random N_{ma}, and computes:

$$A_{ma} = N_{ma}P + H(K_{xz}, ID_{ma}, ID_{hz})P, \quad (1)$$

$$B_{ma} = E_{K_{xz}}(ID_{mx}, T_1, PID_{mx}), \quad (2)$$

$$C_{ma} = Mac_{K_{xz}}(N_{ma}P, ID_{mx}, T_{a1}). \quad (3)$$

$\mathcal{MU}_a$ sends $M_{A1} = \{A_{ma}, B_{ma}, C_{ma}, PID_{ma}, T_{a1}\}$ to $\mathcal{FA}_y$.

Step IA2: $\mathcal{FA}_y$ upon reception of request, checks freshness of T_{a1}, as well as generates fresh time-stamp T_2 and random N_{fy}. $\mathcal{FA}_y$ then computes:

$$A_{fy} = N_{fy}P + H(K_{yz}, ID_{fy}, T_2)P, \quad (4)$$

$$B_{fy} = Mac_{(N_{fy}P)_x}(ID_{hz}, T_{a1}). \quad (5)$$

$\mathcal{FA}_y$ sends $M_{fh2} = \{M_{A1}, A_{fy}, B_{fy}, T_2\}$ to $\mathcal{HA}_z$.

Step IA3: $\mathcal{HA}_z$ verifies freshness of T_2 after receiving message from $\mathcal{FA}_y$ and accepts the message as T_2 is fresh. $\mathcal{HA}_z$ based on PID_{mx} extracts K_{xz} and ID_{mx} from the verifier table and computes:

$$(ID_{mx}, T_{a1}, PID_{mx}) = D_{K_{xz}}(B_{ma}). \tag{6}$$

$\mathcal{HA}_z$ compares the decrypted ID_{mx} from Equation (6) with the one extracted from verifier table. The attacker $\mathcal{MU}_a$ will pass this test as both values are same. Now, $\mathcal{HA}_z$ computes:

$$N_{ma}P = A_{mx} - H(K_{xz}, ID_{mx}, ID_{hz})P. \tag{7}$$

$\mathcal{HA}_z$ checks:

$$C_{ma} \stackrel{?}{=} Mac_{K_{xz}}(N_{ma}P, ID_{mx}, T_{a1}). \tag{8}$$

$\mathcal{HA}_z$ authenticates $\mathcal{MU}_x$ on the basis of equality of Equation (8). $\mathcal{MU}_a$ will also pass this test, as all parameters in computation of C_{ma} were in access to $\mathcal{MU}_a$ and were correctly calculated at the time of computation of C_{ma} by $\mathcal{MU}_a$. Now, $\mathcal{HA}_z$ computes:

$$N_{fy}P = A_{fy} - H(K_{yz}, ID_{fy}, T_2)P. \tag{9}$$

$\mathcal{HA}_z$ then checks:

$$B_{fy} \stackrel{?}{=} Mac_{(N_{fy}P)_x}(ID_{hz}, T_{a1}). \tag{10}$$

As $\mathcal{FA}_y$ is legitimate; therefore, it will pass the check of Equation (10). Hence, $\mathcal{HA}_z$ computes:

$$A_{hz} = N_{mx}P + H(ID_{mx})P + H(K_{yz}, ID_{hz}, N_{fy}P), \tag{11}$$
$$B_{hz} = Mac_{K_{yz}}(N_{fy}P, N_{mx}P + H(ID_{mx}P, T_3)), \tag{12}$$
$$C_{hz} = N_{fy}P + H(K_{xz}, ID_{hz}, N_{mx}P)P, \tag{13}$$
$$D_{hz} = Mac_{K_{xz}}(ID_{fy}, N_{fy}P, T_3, PID_{mx}). \tag{14}$$

$\mathcal{HA}_z$ then updates:

$$K_{yz} = K_{yz} \oplus h(ID_{fy}, N_{fy}P, T_3), \tag{15}$$
$$K_{xz} = K_{xz} \oplus h(ID_{mx}, N_{ma}P, T_3). \tag{16}$$

Finally, HA sends $M_{hf3} = \{A_{hz}, B_{hz}, C_{hz}, D_{hz}, T_3\}$ to $\mathcal{FA}_y$ and increments $SNum_{mx}$.

Step IA4: $\mathcal{FA}_y$ checks freshness of T_3 and computes:

$$N_{mx}P + H(ID_{mx})P = A_{hz} - H(K_{yz}, ID_{hz}, N_{fy}P). \tag{17}$$

$\mathcal{FA}_y$ then verifies validity of B_{hz} and, on success, computes:

$$C_{fy} = Mac_{(N_{mx}P + H(ID_{mx}P))_x}(ID_{fy}, N_{fy}P, T_3, T_4, C_{mx}), \tag{18}$$
$$SK = h(N_{fy}(N_{mx}P + H(ID_{mx})P)). \tag{19}$$

Then, $\mathcal{FA}_y$ sends $M_{fu4} = \{C_{fy}, C_{hz}, D_{hz}, T_3, T_4\}$ to $\mathcal{MU}_x$.

Step IA5: $\mathcal{MU}_a$ intercepts the message and computes:

$$N_{fy}P = C_{hz} - H(K_{xz}, ID_{hz}, N_{ma}P)P, \tag{20}$$
$$SK = h((N_{ma} + H(ID_{mx}))N_{fy}P), \tag{21}$$
$$D_{ma} = Mac_{N_{ma} + H(ID_{mx}P)_x}(C_{fy}, N_{fy}P). \tag{22}$$

$\mathcal{MU}_a$ sends $M_{A5} = \{D_{ma}, T_{A5}\}$ to $\mathcal{FA}_y$.

Step IA6: $\mathcal{FA}_y$ verifies freshness of T_{A5} and checks validity of D_{ma}. As T_{A5} is freshly generated, so it will pass the test. Similarly, $\mathcal{MU}_a$ has access to all parameters used for computation of D_{ma}, so it will also pass the test. Therefore, $\mathcal{MU}_a$ has also deceived the $\mathcal{FA}_y$ and passed the authentication. Now, $\mathcal{MU}_a$ can easily communicate with $\mathcal{FA}_j$ on behalf of $\mathcal{MU}_x$ using the shared key $SK = h(N_{fy}(N_{ma}P + H(ID_{mx})P))$.

3.2. Traceability

Along with security, user anonymity/privacy is of vital interest, if compromised the attacker can foresee victim related important information, including his lifestyle, habits, shopping preferences, and sensitive location-related information of the mobile user. Ensuring (1) identity hiding and (2) untraceability are primary goals of privacy protection. Identity hiding refers to concealing original idntity of the user on public network, and untraceability ensures that no one can predict that two different sessions are requested by a single user. In the scheme of Lu et al., a static parameter PID_{mx} is used as pseudo identity of $\mathcal{MU}_x$, which remains the same for all sessions. Although it provides identity hiding, it lacks untraceability. Therefore, anyone just listening to the public channel can affirm whether or not different sessions are initiated by a single user.

3.3. Incorrectness

In Lu et al.'s scheme, the $\mathcal{HA}_z$ updates the pre-shared keys K_{xz} with $\mathcal{MU}_x$ and K_{yz} with $\mathcal{FA}_y$ during each session as shown in Equation (15) and (16), whereas these keys are not updated on other sides, i.e., $\mathcal{MU}_x$ and $\mathcal{FA}_y$. Hence, the subsequent authentication request will fail and the scheme can work for a single time authentication, which is not required in any scenario, especially in IoT-based systems.

3.4. Scalability Problem

Due to storage of verifier table on $\mathcal{HA}_z$, the scheme may suffer scalability issues. Moreover, finding corresponding entries from a large verifier table may cause delay in delay sensitive scenarios.

4. Proposed scheme

This section explains our improved authentication scheme for roaming user in IoT-based wireless networks, the reasons effecting Lu et al.'s security are considered in designing phase of our improved scheme. The storage of verifier table with entries consisting of tuple $\{ID_{mx}, PID_{mx}, K_{xz}\}$ is the hitch giving space to insecurities. Moreover, the verifier also results in delaying the authentication process. In Lu et al.'s scheme, $\mathcal{HA}_z$ updates the pre-shared keys K_{xz} with $\mathcal{MU}_x$ and K_{yz} with $\mathcal{FA}_y$ during each session, whereas these keys (K_{xz}, K_{yz}) are not updated on other sides, i.e., $\mathcal{MU}_x$ and $\mathcal{FA}_y$. Therefore, the authentication may fail in subsequent sessions. Proposed scheme handles this incorrectness by removing this step, as updation of these keys is an unnecessary step. The proposed scheme avoids usage of any verifier stored on $\mathcal{HA}_z$to provide scuffle-free security. Moreover, the proposed scheme modifies some steps in registration and login/authentication phases. The working of the proposed scheme is shown in Figure 2. Following subsections explain the phases of the scheme:

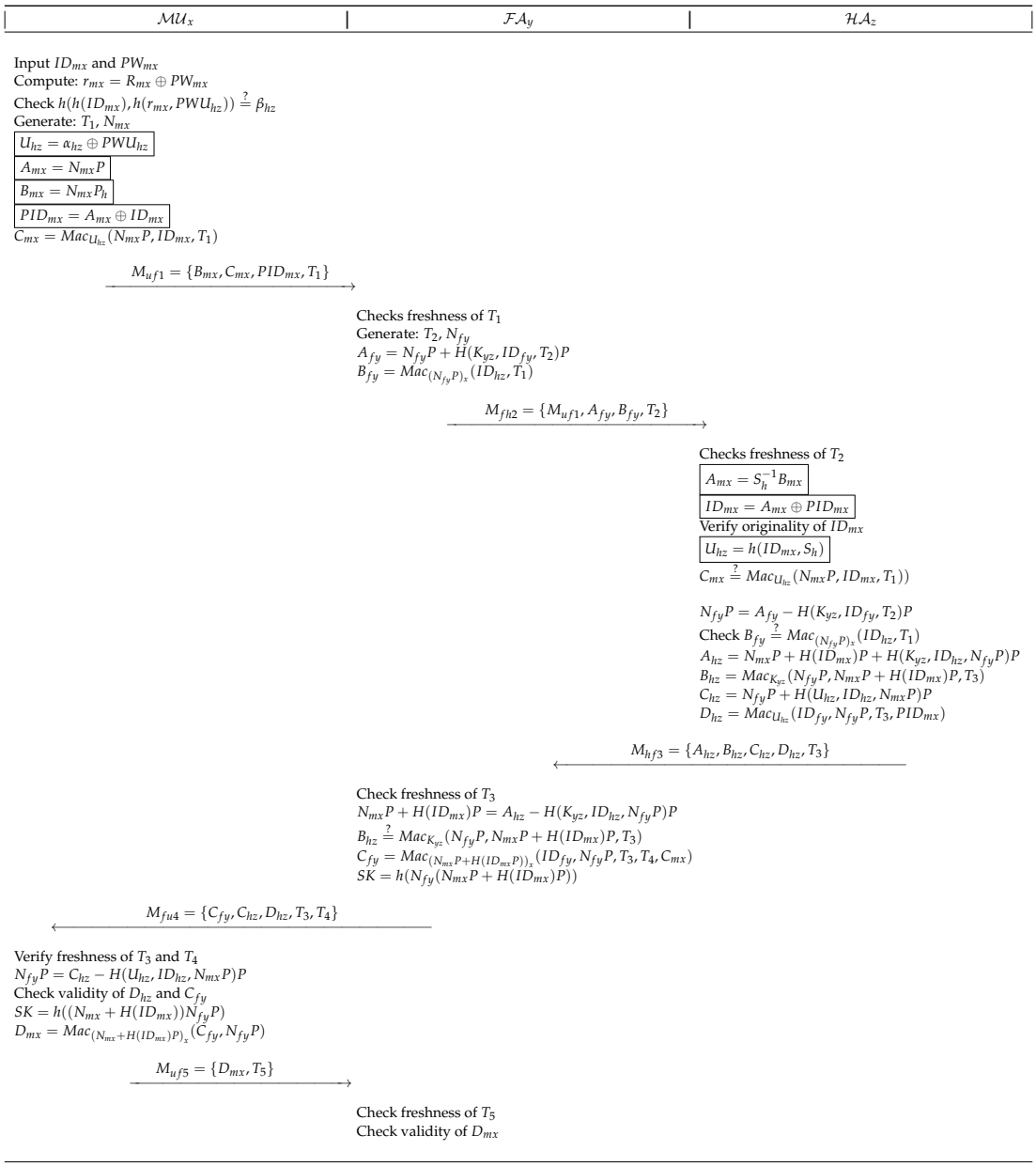

Figure 2. Proposed Scheme.

4.1. System Setup Phase

For system-setup purposes, Home Network Agent $\mathcal{HA}_z$ selects an Elliptic curve $E_p(a, b) : y^2 = x^3 + ax + b \mod p$, where $a,\ b \in F_p$ a finite field, such that $4a^3 + 27b^2 \neq 0$, along with an infinite point O. HA then selects a base point P over $E_p(a, b)$. $\mathcal{HA}_z$ selects a secret key S_h and computes public key $P_h = S_hP$. $\mathcal{HA}_z$ also selects two hash functions $h()$, $H()$, as well as a keyed MAC functions $Mac_k()$, along with symmetric encryption/decryption algorithms $E_k(), D_k()$.

Note: The details of cryptographic primitives, including Hash, keyed MAC, etc., can be found in Reference [32].

4.2. Proposed Registration Phase

Step PRP1: The mobile user $\mathcal{MU}_x$ selects identity/password pair $\{ID_{mx}, PW_{mx}\}$, along with r_{mx} (generated randomly), and computes $PWU_{hz} = h(PW_{mx}, r_{mx})$. $\mathcal{MU}_x$ sends the pair $\{ID_{mx}, PWU_{hz}\}$ to $\mathcal{HA}_z$.

Step PRP2: Upon reception of $\{ID_{mx}, PWU_{hz}\}$ to $\mathcal{HA}_z$ pair from $\mathcal{MU}_x$, $\mathcal{HA}_z$. $\mathcal{HA}_z$ then computes $U_{hz} = h(ID_{mx}, S_h)$, $\alpha_{hz} = U_{hz} \oplus PWU_{hz}$, and $\beta_{hz} = h(h(ID_{mx}), PWU_{hz})$. $\mathcal{HA}_z$ then sends a *smart-card* containing $\{\alpha_{hz}, \beta_{hz}, P_h = S_h P\}$ to $\mathcal{MU}_x$.

Step PRP3: Upon reception of *smart-card*, $\mathcal{MU}_x$ computes $R_{mx} = r_{mx} \oplus PW_{mx}$ inserts r_{mx}. Finally, the *smart-card* contains: $\{\alpha_{hz}, \beta_{hz}, r_{mx}, h(), H(), E_k, D_k, Mac_k, P_h = S_h, P\}$.

4.3. Login & Authentication Phase

Step PLA1: After inserting smart-card, $\mathcal{MU}_x$ inputs ID_{mx} and PW_{mx},the smart-card computes $r_{mx} = R_{mx} \oplus PW_{mx}$ and $PWU_{hz} = h(PW_{mx}, r_{mx})$. The smart-card then verifies $h(h(ID_{mx}), h(r_{mx}, PWU_{hz})) \stackrel{?}{=} \beta_{hz}$. Terminates the session if verification is unsuccessful. Otherwise, generates time-stamp T_1, random N_{mx} and computes $U_{hz} = \alpha_{hz} \oplus PWU_{hz}$, $A_{mx} = N_{mx}P$, $B_{mx} = N_{mx}P_h$, $PID_{mx} = A_{mx} \oplus ID_{mx}$ and $C_{mx} = Mac_{U_{hz}}(N_{mx}P, ID_{mx}, T_1)$. $\mathcal{MU}_x$ sends $M_{uf1} = \{B_{mx}, C_{mx}, PID_{mx}, T_1\}$ to $\mathcal{FA}_y$.

Step PLA2: $\mathcal{FA}_y$ upon reception of request, checks freshness of T_1 and generates fresh time-stamp T_2, random N_{fy}. $\mathcal{FA}_y$ then computes $A_{fy} = N_{fy}P + H(K_{yz}, ID_{fy}, T_2)P$, $B_{fy} = Mac_{(N_{fy}P)_x}(ID_{hz}, T_1)$ and sends $M_{fh2} = \{M_{uf1}, A_{fy}, B_{fy}, T_2\}$ to $\mathcal{HA}_z$.

Step PLA3: $\mathcal{HA}_z$ verifies freshness of T_2 after receiving message from $\mathcal{FA}_y$. Rejects the message, if T_2 is not fresh. Otherwise, $\mathcal{HA}_z$ computes $A_{mx} = S_h^{-1}B_{mx}$ and $ID_{mx} = A_{mx} \oplus PID_{mx}$. $\mathcal{HA}_z$ verifies originality of ID_{mx} stored in subscribers identity table. Upon successful verification, $\mathcal{HA}_z$ computes $U_{hz} = h(ID_{mx}, S_h)$ and verifies $C_{mx} \stackrel{?}{=} Mac_{U_{hz}}(N_{mx}P, ID_{mx}, T_1))$. Upon successful verification, $\mathcal{HA}_z$ computes $N_{fy}P = A_{fy} - H(K_{yz}, ID_{fy}, T_2)P$ and then checks $B_{fy} \stackrel{?}{=} Mac_{(N_{fy}P)_x}(ID_{hz}, T_1)$. On success, $\mathcal{HA}_z$ computes $A_{hz} = N_{mx}P + H(ID_{mx})P + H(K_{yz}, ID_{hz}, N_{fy}P)P$, $B_{hz} = Mac_{K_{yz}}(N_{fy}P, N_{mx}P + H(ID_{mx}P, T_3))$. $\mathcal{HA}_z$ computes $C_{hz} = N_{fy}P + H(U_{hz}, ID_{hz}, N_{mx}P)P$, $D_{hz} = Mac_{U_{hz}}(ID_{fy}, N_{fy}P, T_3, PID_{mx})$. HA then sends $M_{hf3} = \{A_{hz}, B_{hz}, C_{hz}, D_{hz}, T_3\}$ to $\mathcal{FA}_y$.

Step PLA4: $\mathcal{FA}_y$ checks freshness of T_3 after receiving response of $\mathcal{HA}_z$. On success, $\mathcal{FA}_y$ computes $N_{mx}P + H(ID_{mx})P = A_{hz} - H(K_{yz}, ID_{hz}, N_{fy}P)P$. $\mathcal{FA}_y$ then verifies validity of B_{hz} and on success, computes $C_{fy} = Mac_{(N_{mx}P+H(ID_{mx}P))_x}(ID_{fy}, N_{fy}P, T_3, T_4, C_{mx})$.The session key is computed as $SK = h(N_{fy}(N_{mx}P + H(ID_{mx})P))$. Then, $\mathcal{FA}_y$ sends $M_{fu4} = \{C_{fy}, C_{hz}, D_{hz}, T_3, T_4\}$ to $\mathcal{MU}_x$.

Step PLA5: Upon reception, $\mathcal{MU}_x$ verifies freshness of T_3 and T_4 and on success, computes $N_{fy}P = C_{hz} - H(U_{hz}, ID_{hz}, N_{mx}P)P$. $\mathcal{MU}_x$ further checks validity of D_{hz} and C_{fy}, if both holds, $\mathcal{MU}_x$ computes session key $SK = h((N_{mx} + H(ID_{mx}))N_{fy}P)$, $D_{mx} = Mac_{(N_{mx}+H(ID_{mx})P)_x}(C_{fy}, N_{fy}P)$ and sends $M_{uf5} = \{D_{mx}, T_5\}$ to $\mathcal{FA}_y$.

Step PLA6: $\mathcal{FA}_y$ verifies freshness of T_5 and checks validity of D_{mx}. If it holds, $\mathcal{FA}_y$ treats $\mathcal{MU}_x$ as legitimate user and now further communication between $\mathcal{FA}_y$ and $\mathcal{MU}_x$ may be carried out using the shared key $SK = h(N_{fy}(N_{mx}P + H(ID_{mx})P))$.

5. Security Analysis

This section explains the automated formal security validation of the proposed algorithm using popular tool ProVerif, as well as under the hardness assumptions of ECDLP, collision resistant property of one-way hash, and hardness of symmetric encryption algorithm. The section then solicits the informal discussion on required security, supplemented by the security features comparisons with existing related schemes.

5.1. Formal Security Analysis

For the purpose of formal security analysis of our protocol, we define formal interpretations of repetition and chose the cipher-text attack (IDN-CCA) of the symmetric cryptographic algorithm, secure hash collision-resistant function, and ECDLP as follows:

Definition 1. *Given* (Σ, Ω, Φ) *is the algorithm of symmetric key and cipher-text* $CP = ENC_{key}(k)$*, the IDN-CCA's definition is considered as hard problem if* $ADV_{\mathcal{A}}^{IDN-CCA}(t_{a1}) \leq \epsilon_{a1}$*, in which* $ADV_{\mathcal{A}}^{IDN-CCA}(t_{a1})$ *describes an* $\mathcal{A}$*'s benefit in finding the string* $p \in \Omega$ *(the set of plain-texts) of antecedent messages from the given* $CP \in \Sigma$ *(the set of cipher-texts) also algorithm of symmetric key with key* $k \in \Phi$ *(the set of enc/dec keys) which is unknown, for any small enough* $\epsilon_{a1} > 0$ *[32].*

Definition 2. *Given an elliptic curve based point* $G = yP$ *over* $E_p(x, y)$*, the interpretation of the ECDLP is considered as hard problem if* $ADV_{\mathcal{C}}^{ECDLP}(t_{a2}) \leq \epsilon_{a2}$*, in which* $ADV_{\mathcal{C}}^{ECDLP}(t_{a2})$ *describes the benefit of an* $\mathcal{A}$ *in discovering the integer* $y \in \mathcal{Z}_q^*$ *from G and P which are given, for any small enough* $\epsilon_{a2} > 0$ *[32].*

Definition 3. *Given the output* $O = H(y)$*, the interpretation of the function of hash is considered as hard problem if* $ADV_{\mathcal{A}}^{H}(t_{a3}) \leq \epsilon_{a3}$*, in which* $ADV_{\mathcal{A}}^{H}(t_{a3})$ *describes the benefit of an* $\mathcal{A}$ *in extracting the input* $y \in \{0,1\}^*$ *from* $H(y)$ *which is given, for any small enough* $\epsilon_{a3} > 0$ *[32].*
For the formal analysis of security, we have defined random oracles [33] which are as follows:
Reveal 1*: This oracle will output plain-text k unconditionally from cipher-text* $CP = ENC_{key}(k)$ *that is given.*
Reveal 2*: This oracle will output integer y unconditionally from yP and P that are publicly given values.*
Reveal 3*: This oracle will output the input y from O that is the corresponding value of hash.*

Theorem 1. *On the basis of supposition* $IND - CCA$ *Security of Symmetric Cryptography algorithm, the enhanced protocol is provably protected in the arbitrary oracle model across a probabilistic polynomial time restricted attacker for extracting mobile user.*

Proof. Assume that experiment $EXPE1_{\mathcal{A}}^{IND-CCA}$ for the attacker $\mathcal{A}$ who has capability to extract the user's ID, $\mathcal{A}$ be a probabilistic polynomial time restricted attacker. We determine success probability for $EXPE1_{\mathcal{A}}^{IND-CCA}$ like $Succ1_{\mathcal{A}}^{IND-CCA} = 2Pr[EXPE1_{\mathcal{A}}^{IND-CCA} = 1] - 1$. Then, the benefit of $EXPE1_{\mathcal{A}}^{IND-CCA}$ is examined as $Adv_{\mathcal{A}}^{IND-CCA}(t_1, q_{R1}) = max_{\mathcal{A}} Succ1_{\mathcal{A}}^{IND-CCA}$, whereas the maximal is taken overall attacker $\mathcal{A}$ with number of query q_{R1} and time of execution t_1 made the *Reveal1* oracle. the enhanced protocol is provably protected in the arbitrary oracle model across attacker $\mathcal{A}$ for extract the ID of mobile user MU_a if $Adv_{\mathcal{A}}^{IND-CCA}(et_1; q_{R1}) \leq \in_1$, for any appropriately small $\in_1 > 0$. Examine the experiment $EXPE1_{\mathcal{A}}^{IND-CCA}$ as described in *Algorithm* 1, $\mathcal{A}$ can successfully extract the ID of mobile user MU_a if he is able to break $IND - CCA$ security of symmetric encryption description algorithm. Nevertheless, according to Definition 1, we could have $Adv_{\mathcal{A}}^{IND-CCA}(t_1) \leq \in_1$, for any appropriately small $\in_2 > 0$. Thus, we get $Adv_{\mathcal{A}}^{IND-CCA}(t_1; q_{R1}) \leq \in_1$ since $Adv_{\mathcal{A}}^{IND-CCA}(t_1; q_{R1})$ depends on $Adv_{\mathcal{A}}^{IND-CCA}(t_1)$. So, concluded that the enhanced protocol is protected against an $\mathcal{A}$ for extracting the ID of mobile user MU_a. □

Algorithm 1 $EXPR1_{\mathcal{A}}^{CCA-IND}$

1: Intercept the authentication request message
   $M_{uf1} = \{B_{mx}, C_{mx}, PID_{mx}, T_1\}$
   $B_{mx} = N_{mx}P_h$,
   $C_{mx} = M_{ac}U_{h2}(N_{mx}P, ID_{mx}, T_1)$.
2: Call *Reveal3* oracle
   Let $(N_{mx}.P) \leftarrow Reveal\ (B_m)$
3: **if** $(T_1 = T_1)$ **then**
4: Accept ID_{mx} as the true identity of MU_x
5: **return** 1
6: **else**
7: **return** 0
8: **end if**

Theorem 2. *Under the consideration that a hash function intently behaves as an arbitrary oracle model adjacent to a probabilistic polynomial time restricted attacker for extracting session key SK between user and foreign agent.*

Proof. Assume that experiment $EXPE2_{\mathcal{A}}^{Hash,ECDLP}$ for the attacker $\mathcal{A}$ who has capability to extract the arbitrary numbers in calculated the SK between user and foreign agent, $\mathcal{A}$ be a probabilistic

polynomial time restricted attacker. We determine success probability for $EXPE2_{\mathcal{A}}^{Hash,ECDLP}$ as $Succ2_{\mathcal{A}}^{Hash,ECDLP} = 2Pr[EXPE2_{\mathcal{A}}^{Hash,ECDLP} = 1] - 1$. After that, the benefit of $EXPE2_{\mathcal{A}}^{Hash,ECDLP}$ is considered as $Adv_{\mathcal{A}}^{Hash,ECDLP}(t_2; q_{R2}; q_{R3}) = max_{\mathcal{A}} Succ2_{\mathcal{A}}^{Hash,ECDLP}$, whereas the maximal is taken overall attacker $\mathcal{A}$ with time of execution t_2 and number of queries q_{R2} made to *Reveal2* and q_{R3} made to *Reveal3* oracles. The enhanced protocol is provably protected in the random oracle model across $\mathcal{A}$ for the values of hash of session key SK if $Adv_{\mathcal{A}}^{Hash,ECDLP}(t_2; q_{R2}; q_{R3}) \leq \in_2$,for any appropriately small $\in_2 > 0$. Examine the experiment $EXPE2_{\mathcal{A}}^{Hash,ECDLP}$ shown in *Algorithm* 2, $\mathcal{A}$ can successfully extract the values of hash of session key SK if he has the capability to convert the hash function and solve the $ECDLP$. Though, as by the Definition 2 and Definition 3, $Adv_{\mathcal{A}}^{ECDLP}(t_2) \leq \in_3$, $Adv_{\mathcal{A}}^{Hash}(t_3) \leq \in_4$, for any appropriately small $\in_3 > 0$, $\in_4 > 0$. Thus, we get $Adv_{\mathcal{A}}^{Hash,ECDLP}(t_2; q_{R2}; q_{R3}) \leq \in_2$ since $Adv_{\mathcal{A}}^{Hash,ECDLP}(t_2; q_{R2}; q_{R3})$ depends on $Adv_{\mathcal{A}}^{ECDLP}(t_2) \leq \in_3$ and $Adv_{\mathcal{A}}^{Hash}(t_3) \leq \in_4$. So, concluded that the enhanced protocol is provably protected against an attacker for extracting session key SK and foreign agent. □

Algorithm 2 $EXPR2_{\mathcal{A}}^{ECDLP,HASH}$

Intercept the authentication message
$M_{fh2} = \{M_{uf1}, A_{fy}, B_{fy}, T_2\}$
$A_{fy} = N_{fy}P + H(K_{yz}, ID_{fy}, T_2)P$,
$B_{fy} = Mac_{(N_{fy}P)_x}(ID_{h2}, T_1)$.
Intercept the authentication message
$M_{hf3} = \{A_{hz}, B_{hz}, D_{hz}, O_{hz}, T_3\}$,
$A_{hz} = N_{mx}P + H(ID_{mx})P + H(k_{yz}, ID_{hz}, N_{fy}P)P$,
$B_{hz} = Mac_{K_{yz}}(N_{fy}P, N_{mx}P + H(ID_{mx})P, T_3)$,
$C_{hz} = N_{fy}P + H(U_{hz}, ID_{hz}, N_{mx}P)P$,
$D_{hz} = Mac_{U_{hz}}(ID_{fy}, N_{fy}P, T_3, PID_{mx})$.
Intercept the authentication message
$M_{fu4} = \{C_{fy}, C_{hz}, D_{hz}, T_3, T_4\}$
$C_{fy} = Mac_{(N_{mx}P+H(ID_{mx}P))_x}(ID_{fy}, N_{fy}P, T_3, T_4, C_{mx})$.
Call *Reveal2* oracle
Let $(N_{fy}, H(K_{yz}, ID_{fy}, T_2)) \leftarrow$ *Reveal2* (A_{fy}).
Call *Reveal3* oracle
Let $(K\acute{}_{yz}, ID\acute{}_{fy}, T\acute{}_2) \leftarrow$ *Reveal3* $(H(K_{yz}, ID_{fy}, T_2))$
Call *Reveal2* oracle
Let $(N_{mx}, H(ID_{mx}), H(K_{yz}, ID_{hz}, N_{fy})) \leftarrow$ *Reveal* (A_{hz})
Call *Reveal3* oracle
Let $(K^*_{yz}, ID_{hz}, N_{fy})) \leftarrow$ *Reveal2* $(H(K_{yz}, ID_{hz}, N_{fy}))$
if $(T_2 = T\acute{}_1)$ **then**
Accept $N\acute{}_{fy}$ as an arbitrary number of FA_y
if $(K^*_{yz} = K\acute{}_{yz})$ **then**
Calculates $SK = h(N_{mx} + H(ID_{mx})N_{fy}P)$
$C_{fy} = Mac_{(N_{mx}P+H(ID_{mx})P)_x(ID_{fy}, N_{fy}P, T_3, T_4, C_{mx})}$
if $(C_{fy} = C\acute{}_{fy})$ **then**
SK is accepted betweeen MU_x and FA_y
return 1
else
return 0
end if
else
return 0
end if
else
return 0
end if

5.2. Automated Security Analysis with ProVerif

We chose the prevailing software tool ProVerif [34,35] for performing an automated security perusal. The ProVerif is developed over the concept of applied π calculus [36]. It is able to test and simulate many cryptographic operations, such as encryption/decryption, symmetric/asymmetric cryptosystems, hashes, signatures, etc. It can substantiate the characteristics of secrecy and authenticity. Complete protocol as given in Figure 2 is implemented and verified in ProVerif. Three channels as shown in Figure 3a are introduced in the implementation. The secure channel sch1 is dedicated for facilitating registration between mobile user and home agent, whereas two public channels pch2 and pch3 have been introduced for commencing communication between mobile user and home agent

with foreign agent. Subsequently, variables and constants are also defined in Figure 3a. To keep the mobile user anonymous, its identity IDmx is kept private, whereas identities of home and foreign agents, i.e., IDhz and IDfy, respectively, are public. Mobile user's password PWmx, shared keys Kxz, Kyz between mobile user-home agent and foreign agent-home agent, respectively, are assumed as private. Sh and Ph are considered as the private public key pairs of home agent. The Constructors are specified to simulate cryptographic operations and functions. Thereafter, destructor and equation are specified to simulate inverse and decryption.

```
(**.....**Channels**.....**)
free sch1:channel [private].      (*  MU<...>HA )
free pch2:channel.                (*  MU<...>FA  *)
free pch3:channel.                (*  HA<...>FA  *)
(**.....**Constants*Variables**.....**)
const P:bstr.
free IDmx:bstr.   [private].
free IDhz:bstr.
free IDfy:bstr.
free PWmx:bstr.   [private].
free Kxz:bstr.    [private].
free Kyz:bstr.    [private].
free Sh:bstr.     [private].
free Ph:bstr.
(**.....**Constructor**.....**)
fun Con(bstr,bstr):bstr.
fun Add(bstr,bstr):bstr.
fun Sub(bstr,bstr):bstr.
fun XoR(bstr,bstr):bstr.
fun OR(bstr,bstr):bstr.
fun Mul(bstr,bstr):bstr.
fun Inv(bstr):bstr.
fun H(bstr):bstr.
fun Enc(bstr,bstr):bstr [private].
fun Mac(bstr):bstr.
(**.....**Destructors * Equations**.....**)
reduc forall m:bstr,key:bstr; Dec(Enc(m,key),key)=m.
equation forall a:bstr; Inv(Inv(a))=a.
```

```
(**.....*Events*.....**)
event beginMUser(bstr).
event endMUser(bstr).
event beginHAgt(bstr).
event endHAgt(bstr).
event beginFAgt(bstr).
event endFAgt(bstr).
(**.....*Process Replication*.....**)
process ( (!pMuser) | (!pFAgt) | (!pHAgt) )
(**.....*Queries*.....**)
free SK:bstr [private].
query attacker(SK).
query id:bstr; inj event(endMuser(id)) ==> inj event(
    beginMuser(id)).
query id:bstr; inj event(endFAgt(id)) ==> inj event(
    beginFAgt(id)).
query id:bstr; inj event(endHAgt(id)) ==> inj event(
    beginHAgt(id)).
```

```
(**.....*Results*.....**)
1 Result inj event(endMuser(id_2301)) ==> inj event(
    beginMuser(id_2301)) is True.
2 Result inj event(endFAgt(id_4321)) ==> inj event(
    beginFAgt(id_4321)) is True.
3 Result inj event(endHAgt(id_6435)) ==> inj event(
    beginHAgt(id_6435)) is True.
4 Result not attacker(SK()) is True
```

```
(**.....*Mobile Node Process*.....**)
let pMuser=
new rmx:bstr;
let PWU = H(Con(PWmx,rmx)) in
out(sch1, (IDmx, PWU));
in(sch1, (xahz:bstr, xbhz:bstr, xPh:bstr ));
event beginMUser(IDmx);
let rmx = XoR(Rmx, PWmx) in
if (H(Con(H(IDmx),H(Con(rmx, PWU)))) = xbhz) then
new Nmx:bstr;
new T1:bstr;
let Uhz = XoR(xahz, PWU) in
let Amx = Mul(Nmx, P) in
let Bmx = Mul(Nmx, xPh) in
let PIDmx = XoR(Amx, IDmx) in
let Cmx = Mac(Con(Mul(Nmx, P),T1,IDmx), Uhz) in
out(pch2:, Muf1=(Bmx, Cmx, PIDmx, T1));
in(pch2, Mfu4=(xCfy:bstr, xChz:bstr, xDhz:bstr, xT3:bstr,
    xT4:bstr));
let Mul(Nfy, P) = Sub(xChz, Mul(H(Con(Uhz,IDhz, Mul(Nmx,
    P))),P)) in
if(Cfy' = xCfy) then
if(Dhz' = xDhz) then
let SK = H(OR(Nmx, Mul(H(IDmx),Mul(Nfy, P)))) in
let Dmx = Mac(Con(xCfy,Mul(Nfy, P)), OR(Nmx, Mul(H(IDmx),
    Mul(Nfy, P)))x) in
out(pch2:, Muf5=(Dmx, T5));
event endMUser(IDmx).
(**.....*Forign Agent Process*.....**)
let pFAgt=
in(pch2, xMuf1:bstr=(xBmx:bstr, xCmx:bstr, xPIDmx:bstr,
    xT1:bstr));
event beginFAgt(IDfy);
new Nfy:bstr;
new T2:bstr;
let Afy = OR(Mul(Nfy, P),Mul(H(Con(Kyz,IDfy,T2), P))) in
let Bfy = Mac(Con(IDhz,xT1), Mul(Nfy, P)x) in
out(pch3:, Mfh2=(Muf1, Afy, Bfy, T2));
in(pch3, xMhf3:bstr=(xAhz:bstr, xBhz:bstr, xChz:bstr,
    xDhz:bstr, xT3:bstr));
let OR(Mul(Nmx, P),Mul(H(IDmx), P)) = Sub(xAhz, Mul(H(Con
    (Kyz,IDhz,Mul(Nfy,P)),P)))in
if(Bhz' = xBhz) then
let Cfy = Mac(Con(IDfy,Mul(Nfy, P),xT3, T4, Cmx), OR(Nmx,
    Mul(H(IDmx), P)x)) in
let SK = H(Mul(Nfy, OR(Mul(Nmx, P),Mul(H(IDmx), P)))) in
event endFAgt(IDfy).
(**.....*Home Agent Process*.....**)
let pHAgt=
in(pch3, xMfh2:bstr = (xMuf1:bstr, xAfy:bstr, xBfy:bstr,
    xT2:bstr));
event beginHAgt(IDhz);
let Amx = Mul(Inv(Sh),Bmx) in
let IDmx = XoR(Amx, PIDmx) in
if(IDmx' = IDmx) then
let Uhz = h(Con(IDmx, Sh)) in
let Cmx' = Mac(Con(Mul(Nmx, P),T1,IDmx), Uhz) in
if(Cmx' = Cmx) then
let Mul(Nfy, P) = Sub(xAfy, Mul(H(Con(Kyz,IDfy, xT2)),P))
    in
let Bfy' = Mac(Con(IDhz,xT1), Mul(Nfy, P)x) in
if(Bfy' = Bfy) then
let Ahz = OR(Mul(Nmx, P),OR(Mul(H(IDmx),P),Mul(H(Con(Kyz,
    IDfy,T2)),P))) in
let Bhz = Mac(Con(Mul(Nfy, P),XoR(Mul(Nmx, P),Mul(H(IDmx)
    , P)),T3), Kyz) in
let Chz = XoR(Mul(Nfy, P), Mul(H(Con(Uhz,IDhz,Mul(Nmx,P))
    ),P)) in
let Dhz = Mac(Con(IDfy, Mul(Nfy,P), T3, PIDmx),Uhz) in
event endHAgt(IDhz).
```

Figure 3. ProVerif Simulation.

Every participant can be described through two events a begin and an end event. The protocol authenticity is realized through exposing the respective relationship between begin and end interval of the related event initiated by the specific participant. If end event is not reached it simply means the protocol terminated unsuccessfully and scheme is incorrect. In Figure 3b, three distinct processes are implemented and simulated on behalf of three participants. These participants includes pMuser, pHagt, and pFagt, which are defined and implemented as shown in Figure 2 and described in Section 4. The proposed scheme is simulated as an unbounded parallel execution of user, home and foreign networks processes.

The subsequent four queries are defined in Figure 3c to substantiate the security and correctness of our protocol. The query attacker simulates an actual attack to expose the session key, whereas another 3 queries inj-event corresponds to begin and end event of 3 processes, i.e., user, home, and foreign networks. If any of these queries results false, it implies the scheme is incorrect. The abilities of an attacker are evaluated by executing the Not-attacker (SK) predicate, where *SK* is private. It is

assumed that public parameters are accessible to the attacker. The Not-attacker is also applied over SK. Moreover, three successive queries on inj-event affirms the association between initiation and termination of events corresponding to each of these processes, i.e., user, home, and foreign networks. The outcome of the discussed queries are shown in Figure 3d.

It is observed through results 1, 2, and 3 in Figure 3d that each process initiated and terminated successfully, which substantiates the correctness of our scheme, whereas result 4 Not-attacker (SK) affirms that session key is secure against security threats. Hence, our protocol maintains authenticity and secrecy during its execution.

5.3. Security Requirements

The security requirement of the proposed scheme and a comparison of the proposed scheme with related competing schemes [9,12,14,25,26] is detailed in following subsections. Table 2 also illustrates the comparisons and confirms that only the proposed scheme provides all the required features and resists known attacks, whereas competing schemes lacks either some features or ensuring against some known attack.

Table 2. Comparison of functional security.

↓ **Features/Scheme** →	[9]	[12]	[14]	[25]	[26]	**Our**
Mutual Authentication	✓	✓	✓	✓	✓	✓
Correctness	✓	✓	✓	✓	✗	✓
User Anonymity/Untraceability	✗	✓	✓	✓	✗	✓
Perfect Forward Secrecy	✓	✓	✓	✗	✓	✓
Resists User Forgery	✓	✓	✗	✓	✓	✓
Resists Stolen Verifier	✓	✓	✓	✓	✗	✓
Resists Insiders	✓	✓	✓	✓	✗	✓
Resists Stolen Smart-Card	✓	✓	✗	✓	✓	✓
Resists Known Session parameters	✓	✓	✓	✗	✓	✓

Provides: ✓, Not-Provides: ✗.

5.3.1. Mutual Authentication

The proposed scheme, through $\mathcal{HA}_z$ (the home agent) provides mutual authentication between $\mathcal{MN}_x$ (the mobile node) and $\mathcal{FA}_y$ (the foreign agent). $\mathcal{HA}_z$ authenticates $\mathcal{MN}_x$ by validating $C_{mx} \stackrel{?}{=} Mac_{U_{hz}}(N_{mx}P, ID_{mx}, T_1))$, computation of valid/legal C_{mx} requires an adversary to have access to the secret parameter of $\mathcal{MN}_x$, i.e., $U_{hz} = h(ID_{mx}, S_h)$, as well as valid/legal $N_{mx}P$, which can only be extracted though A_{mx} by the use of secret key (S_h) of $\mathcal{HA}_z$. Neither U_{hz} nor $N_{mx}P$ can be computed by any adversary, which implies that only valid $\mathcal{MN}_x$ can pass this test. Moreover, $\mathcal{HA}_z$ authenticates $\mathcal{FA}_y$ by validating $B_{fy} \stackrel{?}{=} Mac_{(N_{fy}P)_x}(ID_{hz}, T_1)$. The computation of valid/legal B_{fy} requires an adversary to extract $N_{fy}P$, which can by computed by public parameter $A_{fy} = N_{fy}P + H(K_{yz}, ID_{fy}, T_2)P$ sent by $\mathcal{FA}_y$. The computation of A_{fy} requires an adversary to have access to the pre-shared secret key K_{yz} among $\mathcal{HA}_z$ and $\mathcal{FA}_y$. No adversary, insider/outsider can have access to the pre-shared secret key. Therefore, only legal/valid $\mathcal{FA}_y$ can pass this test. Similarly, $\mathcal{FA}_y$ authenticates $\mathcal{HA}_z$ validating $B_{hz} \stackrel{?}{=} Mac_{K_{yz}}(N_{fy}P, N_{mx}P + H(ID_{mx})P, T_3)$, the computation of valid B_{hz} requires an adversary to have access to pre-shared secret key K_{yz} between $\mathcal{HA}_z$ and $\mathcal{FA}_y$. Moreover, the adversary also needs to compute the valid/legal, corresponding $N_{fy}P$ against the parameter $A_{fy} = N_{fy}P + H(K_{yz}, ID_{fy}, T_2)P$ sent on public channel earlier by $\mathcal{FA}_y$ to $\mathcal{HA}_z$, the computation of A_{fy} again requires the use of pre-shared secret key K_{yz}. Therefore, only valid $\mathcal{HA}_z$ can pass this test. likewise, $\mathcal{MN}_x$ authenticates: 1) $\mathcal{HA}_z$ by validating $D_{hz} \stackrel{?}{=} Mac_{U_{hz}}(ID_{fy}, N_{fy}P, T_3, PID_{mx})$ and 2) $\mathcal{FA}_y$ by verifying $C_{fy} \stackrel{?}{=} Mac_{(N_{mx}P+H(ID_{mx}P))_x}(ID_{fy}, N_{fy}P, T_3, T_4, C_{mx})$. To generate a valid/legal D_{hz}, an adversary requires having access to secret parameter U_{hz} of $\mathcal{MN}_x$, as well as computation of valid/legal $N_{fy}P$, both of which can be performed only by legal $\mathcal{HA}_z$. Likewise, to generate valid C_{fy},

an adversary requires to compute valid/legal $N_{mx}P + H(ID_{mx}P$, $N_{fy}P$ and C_{mx}. All the mentioned parameters can only be computed by legal $\mathcal{FA}_y$. Hence, mutual authentication among $\mathcal{MN}x$ and $\mathcal{FA}_y$ through $\mathcal{HA}_z$ is essential trait of the proposed scheme.

5.3.2. Correctness

The proposed scheme correctly accomplishes the process of authentication between $\mathcal{MN}x$ and $\mathcal{FA}_y$ through $\mathcal{HA}_z$. Unlike Lu et al.'s scheme, in the proposed scheme, $\mathcal{HA}_z$ does not unnecessarily updates (K_{xz}, K_{yz}) after each successful login. More precisely, the proposed schemes does not require any verifier table for any user; therefore, no entry can be modified by $\mathcal{HA}_z$. Due to non-usage of verifier table by $\mathcal{HA}_z$, the user request does not involve fining and comparing with verifier entries, which helps in minimizing the delay. Hence, the proposed scheme provides correct and secure authentication process.

5.3.3. User Anonymity/Untraceability

Unfortunately and despite their claim, in the scheme of Lu et al. the pseudo identity PID_{mx} remains same not only for multiple but for all sessions. In the proposed scheme, on every login/authentication request $\mathcal{MN}x$ selects a new random variable N_{mx} and computes the dynamic pseudo identity $PID_{mx} = N_{mx}P \oplus ID_{mx}$. Therefore, the proposed scheme not only provides identity hiding but also untraceability/unlinkability.

5.3.4. Perfect Forward Secrecy:

The session key $SK = h(N_{fy}(N_{mx}P + H(ID_{mx})P))$ computed after successful authentication among $\mathcal{MN}_x$ or $\mathcal{FA}_y$ contains the share from both, i.e., N_{mx} from $\mathcal{MN}_x$ and N_{fy} from $\mathcal{FA}_y$. Both N_{mx} and N_{fy} are generated freshly for each session. Moreover, neither $\mathcal{MN}_x$ nor $\mathcal{FA}_y$ having full control on key generation. Even if one or more shared keys from previous session/s are compromised, the adversary may not be able to compute any future session key. Hence, the proposed scheme provides perfect forward secrecy.

5.3.5. User Forgery Attack

As described in Section 5.3.1, the $\mathcal{HA}_z$ authenticates the user by validating C_{mx} and valid/legal C_{mx} can only be computed by legal $\mathcal{MN}_x$. Moreover, $\mathcal{FA}_y$ authenticates $\mathcal{MN}_x$ by validating $D_{mx} \stackrel{?}{=} Mac_{(N_{mx}+H(ID_{mx})P)_x}(C_{fy}, N_{fy}P)$, an adversary requires to compute $N_{mx}P$, as well as $N_{fy}P$. Only legal $\mathcal{MN}_x$ can compute it's own secretly generated parameter $N_{mx}P$ and extract $N_{fy}P$ out of $N_{fy}P = C_{hz} - H(U_{hz}, ID_{hz}, N_{mx}P)P$, which requires the usage of secret parameter U_{hz} of $\mathcal{MN}_x$. Therefore, the proposed scheme strongly resists user forgery attack.

5.3.6. Stolen Verifier and Insider Attack

The home agent $\mathcal{HA}_z$, in the proposed scheme does not store any information relating to the credentials of, including password, $\mathcal{MN}_x$; rather, $\mathcal{HA}_z$ is free of any verifier table. The only information stored is the public identities of the users. Moreover, during registration process, $\mathcal{MN}_x$ sends $PWU_{hz} = h(PW_{mx}, r_{mx})$, along with ID_{mx}, to $\mathcal{HA}_z$. The password is concealed in one-way hash function, along with a random number. Therefore, no deceitful insider gets any information relating to password and is having no advantage. Hence, the proposed scheme resists insider attacks. Moreover, without verifier table, the stolen verifier is impossible in the proposed scheme.

5.3.7. Stolen Smart-Card Attack

In the proposed scheme, the smart-card contains $\{\alpha_{hz}, \beta_{hz}, r_{mx}, h(), H(), E_k, D_k, Mac_k, P_h = S_h, P\}$, where, the user related information is stored in α_{hz}, β_{hz} and r_{mx} parameters, where $\alpha_{hz} = U_{hz} \oplus PWU_{hz}$, and $\beta_{hz} = h(h(ID_{mx}), PWU_{hz})$. Extracting password information from α or β requires inverse to hash

function, which by definition is a hard problem. Moreover, user secret parameter U_{hz} is also concealed with PWU_{hz}, and without password information, it is computationally infeasible to compute U_{hz}. Therefore, the proposed scheme resists stolen smart-card attacks.

5.3.8. Known Session-Specific Parameters Attack

The adversary in the proposed scheme may not able to compute session key even if, he gets the session parameters N_{mx} and N_{fy}, as the session key also requires the hashed identity concealed in an elliptic curve point $H(ID_{mx})P$. Computation of ID_{mx} needs to break on way property of hash, as well as elliptic curve discrete logarithm problem. Therefore, the proposed scheme resists known session-specific parameters attack.

6. Performance Comparisons

This section illustrates the performance comparisons of the proposed with competing schemes. For performance comparison purposes, following notations are used:

- T_{hm}: Computation time for hash/mac operations
- T_{ed}: Computation time for Symmetric Enc/Dec
- T_{pme}: Computation time for scalar multiplication of point over $E_p(a, b)$
- T_{pae}: Computation time for addition of points over $E_p(a, b)$
- T_{me}: Computation time for modular exponentiation
- T_{pb}: Computation time for bilinear pairing
- T_{mtp}: Computation time for map to point hash

Referring the results of Kilinic and Yanik [37], the experiment time computed over Ubuntu 12.04.1 LTS 32bit Operating system with version (0.5.12) of PBC library structured on the version (5.0.5) of the GMP Library on an Intel PC with Dual CPU E2200 2.20GHz and with memory of 2048 MB, the execution time for $T_{hm} \approx 0.0023$ ms, $T_{ed} \approx 0.0046$ ms, $T_{pme} \approx 2.226$ ms, $T_{pae} \approx 0.0288$ ms, $T_{me} \approx 3.85$ ms, $T_{pb} \approx 5.811$ ms, and $T_{mph} \approx 0.947$ ms, respectively. The computation costs of each scheme is presented in Table 3. The scheme of Reddy et al. completes the authentication by computing $18T_{hm} + 4T_{pme}$, the scheme of Li et al. requires $10T_{pme} + 1T_{pae} + 17T_{hm} + 2T_{pb} + 1T_{mtp}$ operations for a successful authentication procedure, the scheme of Jiang et al. computes $12T_{hm} + 2T_{me}$ to accomplish the authentication process, and the scheme of Gope-Hwang performs $21T_{hm}$ during authentication, whereas Lu et al.'s scheme completes a round of authentication procedure with computation cost $25T_{hm} + 15T_{pme} + 10T_{pae} + 3T_{ed}$. The computation cost of the proposed scheme is $23T_{hm} + 14T_{pme} + 7T_{pae}$, although the computation cost of the proposed scheme is bit higher than some competing schemes. However, while providing all security features, the proposed scheme reduced $2T_{hm}$, $1T_{pme}$, $3T_{pae}$, and $3T_{ed}$ as compared with seminal Lu et al.'s scheme. Table 3 also shows execution time of all competing schemes; it is shown that proposed scheme completes roaming authentication in 31.8946 ms and reduced approximately 1.8547 ms as compared with Lu et al.'s scheme.

Table 3. Comparison of computation cost.

Entity → Scheme ↓	$\mathcal{MU}_x$	$\mathcal{FA}_y$	$\mathcal{HA}_k$	Total	Time (ms)
[9]	$10T_{hm} + 2T_{pme}$	$4T_{hm} + 2T_{pme}$	$4T_{hm}$	$18T_{hm} + 4T_{pme}$	8.9454
[12]	$5T_{pme} + 1T_{pae} + 7T_{hm} + 1T_{mtp} + 1T_{pb}$	$3T_{pme} + 1T_{pb} + 5T_{hm}$	$2T_{pme} + 5T_h$	$10T_{pme} + 1T_{pae} + 17T_{hm} + 2T_{pb} + 1T_{mtp}$	34.936
[14]	$3T_{hm} + 1T_{me}$	$4T_{hm}$	$5T_{hm} + 1T_{me}$	$12T_{hm} + 2T_{me}$	7.7276
[25]	$6T_{hm}$	$5T_{hm}$	$10T_{hm}$	$21T_{hm}$	0.0483
[26]	$10T_{hm} + 5T_{pme} + 3T_{pae} + 2T_{ed}$	$6T_{hm} + 4T_{pme} + 2T_{pae}$	$9T_{hm} + 6T_{pme} + 5T_{pae} + 1T_{ed}$	$25T_{hm} + 15T_{pme} + 10T_{pae} + 3T_{ed}$	33.7493
our	$9T_{hm} + 5T_{pme} + 2T_{pae}$	$6T_{hm} + 4T_{pme} + 2T_{pae}$	$8T_{hm} + 5T_{pme} + 3T_{pae}$	$23T_{hm} + 14T_{pme} + 7T_{pae}$	31.8946

7. Conclusions

In this paper, we identified weaknesses of Lu et al.' scheme against stolen verifier and traceability attacks. We also identified that their scheme has correctness issues besides scalability. To combat the weaknesses, we proposed an improved scheme for IoT-based wireless networks. The formal, informal, and automated security analysis has proven that our scheme with stands the known attacks, whereas the performance analysis has shown that our scheme is more efficient and practical as compared with Lu et al.'s scheme. The proposed scheme is more practical in roaming scenarios.

Author Contributions: B.A.A. wrote the initial draft, revision and was involved in ProVerif Simulation. S.A.C. conceptualized the idea and performed cryptanalysis and designed the new scheme. A.B., and A.A.-B. performed security and efficiency analysis. M.H.A. performed formal analysis and supervised the whole process. All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This Project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant No. (RG-7-611-40). The author, therefore, gratefully acknowledge the DSR for technical and financial support.

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

References

1. He, D.; Kumar, N.; Khan, M.K.; Lee, J. Anonymous two-factor authentication for consumer roaming service in global mobility networks. *IEEE Trans. Consum. Electron.* **2013**, *59*, 811–817. [CrossRef]
2. Li, X.; Liu, S.; Wu, F.; Kumari, S.; Rodrigues, J.J.P.C. Privacy Preserving Data Aggregation Scheme for Mobile Edge Computing Assisted IoT Applications. *IEEE Internet Things J.* **2019**, *6*, 4755–4763. [CrossRef]
3. Wei, F.; Vijayakumar, P.; Jiang, Q.; Zhang, R. A Mobile Intelligent Terminal Based Anonymous Authenticated Key Exchange Protocol for Roaming Service in Global Mobility Networks. *IEEE Trans. Sustain. Comput.* **2018**, 1-1. [CrossRef]
4. Jiang, Y.; Lin, C.; Shen, X.; Shi, M. Mutual Authentication and Key Exchange Protocols for Roaming Services in Wireless Mobile Networks. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 2569–2577. [CrossRef]
5. Jo, H.J.; Paik, J.H.; Lee, D.H. Efficient Privacy-Preserving Authentication in Wireless Mobile Networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 1469–1481. [CrossRef]
6. Hsu, R.; Lee, J.; Quek, T.Q.S.; Chen, J. GRAAD: Group Anonymous and Accountable D2D Communication in Mobile Networks. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 449–464. [CrossRef]
7. Alezabi, K.A.; Hashim, F.; Hashim, S.J.; Ali, B.M. An efficient authentication and key agreement protocol for 4G (LTE) networks. In Proceedings of the 2014 IEEE REGION 10 SYMPOSIUM, Kuala Lumpur, Malaysia, 14–16 April 2014; pp. 502–507.
8. 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. *Math. Comput. Model.* **2012**, *55*, 214–222. [CrossRef]
9. Goutham Reddy, A.; Yoon, E.; Das, A.K.; Yoo, K. Lightweight authentication with key-agreement protocol for mobile network environment using smart cards. *IET Inf. Secur.* **2016**, *10*, 272–282. [CrossRef]

10. El Idrissi, Y.E.H.; Zahid, N.; Jedra, M. An Efficient Authentication Protocol for 5G Heterogeneous Networks. In *Ubiquitous Networking*; Sabir, E., García Armada, A., Ghogho, M., Debbah, M., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 496–508.
11. Su, C.; Santoso, B.; Li, Y.; Deng, R.H.; Huang, X. Universally Composable RFID Mutual Authentication. *IEEE Trans. Dependable Secur. Comput.* **2017**, *14*, 83–94. [CrossRef]
12. 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]
13. He, D.; Chen, C.; Chan, S.; Bu, J. Secure and Efficient Handover Authentication Based on Bilinear Pairing Functions. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 48–53. [CrossRef]
14. 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]
15. Zhu, J.; Ma, J. A new authentication scheme with anonymity for wireless environments. *IEEE Trans. Consum. Electron.* **2004**, *50*, 231–235.
16. Tsai, J.L.; Lo, N.W.; Wu, T.C. Secure Handover Authentication Protocol Based on Bilinear Pairings. *Wirel. Pers. Commun.* **2013**, *73*, 1037–1047. [CrossRef]
17. Chang, C.C.; Lee, C.Y.; Chiu, Y.C. Enhanced authentication scheme with anonymity for roaming service in global mobility networks. *Comput. Commun.* **2009**, *32*, 611–618. [CrossRef]
18. Chaudhry, S.A.; Albeshri, A.; Xiong, N.; Lee, C.; Shon, T. A privacy preserving authentication scheme for roaming in ubiquitous networks. *Clust. Comput.* **2017**, *20*, 1223–1236. [CrossRef]
19. Chen, C.M.; Xiang, B.; Liu, Y.; Wang, K.H. A secure authentication protocol for internet of vehicles. *IEEE Access* **2019**, *7*, 12047–12057. [CrossRef]
20. Chen, C.M.; Wang, K.H.; Yeh, K.H.; Xiang, B.; Wu, T.Y. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3133–3142. [CrossRef]
21. Wang, D.; Wang, P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Comput. Netw.* **2014**, *73*, 41–57. [CrossRef]
22. Youn, T.; Park, Y.; Lim, J. Weaknesses in an Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks. *IEEE Commun. Lett.* **2009**, *13*, 471–473. [CrossRef]
23. Kim, J.S.; Kwak, J. Improved secure anonymous authentication scheme for roaming service in global mobility networks. *Int. J. Secur. Its Appl.* **2012**, *6*, 45–54.
24. Lee, H.; Lee, D.; Moon, J.; Jung, J.; Kang, D.; Kim, H.; Won, D. An improved anonymous authentication scheme for roaming in ubiquitous networks. *PLoS ONE* **2018**, *13*, e0193366. [CrossRef] [PubMed]
25. Gope, P.; Hwang, T. Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks. *IEEE Syst. J.* **2015**, *10*, 1370–1379. [CrossRef]
26. 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**, 1–12. [CrossRef]
27. Eisenbarth, T.; Kasper, T.; Moradi, A.; Paar, C.; Salmasizadeh, M.; Shalmani, M. 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*; Wagner, D., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volune 5157, pp. 203–220.
28. Dolev, D.; Yao, A.C. On the security of public key protocols. *Inf. Theory, IEEE Trans.* **1983**, *29*, 198–208. [CrossRef]
29. He, D.; Zeadally, S.; Kumar, N.; Lee, J.H. Anonymous Authentication for Wireless Body Area Networks With Provable Security. *IEEE Syst. J.* **2016**, *11*, 2590–2601. [CrossRef]
30. He, D.; Kumar, N.; Shen, H.; Lee, J.H. One-to-many authentication for access control in mobile pay-TV systems. *Sci. China Inf. Sci.* **2016**, *59*, 052108. [CrossRef]
31. Kumari, S.; Li, X.; Wu, F.; Das, A.K.; Arshad, H.; Khan, M.K. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. *Future Gener. Comput. Syst.* **2016**, *63*, 56–75. [CrossRef]
32. Hoffstein, J. An introduction to cryptography. In *An Introduction to Mathematical Cryptography*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–523.

33. Bellare, M.; Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security, CCS93, Fairfax, VA, USA, 3–5 November 1993; pp. 62–73.
34. Xie, Q.; Hwang, L. Security enhancement of an anonymous roaming authentication scheme with two-factor security in smart city. *Neurocomputing* **2019**, *347*, 131–138. [CrossRef]
35. Mansoor, K.; Ghani, A.; Chaudhry, S.A.; Shamshirband, S.; Ghayyur, S.A.K.; Mosavi, A. Securing IoT-Based RFID Systems: A Robust Authentication Protocol Using Symmetric Cryptography. *Sensors* **2019**, *19*, 4752. [CrossRef]
36. Ghani, A.; Mansoor, K.; Mehmood, S.; Chaudhry, S.A.; Rahman, A.U.; Najmus Saqib, M. Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key. *Int. J. Commun. Syst.* **2019**, *32*, e4139. [CrossRef]
37. Kilinc, H.; Yanik, T. A Survey of SIP Authentication and Key Agreement Schemes. *Commun. Surv. Tutorials IEEE* **2014**, *16*, 1005–1023. [CrossRef]