Article

A Multi-Server Two-Factor Authentication Scheme with Un-Traceability Using Elliptic Curve Cryptography

Guosheng Xu 1, Shuming Qiu 1,2,*, Haseeb Ahmad 3, Guoai Xu 1, Yanhui Guo 1, Miao Zhang 1 and Hong Xu 4

1 School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; guoshengxu@bupt.edu.cn (G.X.); xga@bupt.edu.cn (G.X.); yhguo@bupt.edu.cn (Y.G.); zhangmiao@bupt.edu.cn (M.Z.)
2 Elementary Educational College, Jiangxi Normal University, Nanchang 330022, China
3 Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan; haseeb_ad@hotmail.com
4 High-Tech Research and Development Center, the Ministry of Science and Technology, Beijing 100044, China; xuhong@htrdc.com
* Correspondence: qiushuming2008@163.com or shumingqiu@bupt.edu.cn

Received: 2 July 2018; Accepted: 20 July 2018; Published: 23 July 2018

Abstract: To provide secure communication, the authentication-and-key-agreement scheme plays a vital role in multi-server environments, Internet of Things (IoT), wireless sensor networks (WSNs), etc. This scheme enables users and servers to negotiate for a common session initiation key. Our proposal first analyzes Amin et al.’s authentication scheme based on RSA and proves that it cannot provide perfect forward secrecy and user un-traceability, and is susceptible to offline password guessing attack and key-compromise user impersonation attack. Secondly, we provide that Srinivas et al.’s multi-server authentication scheme is not secured against offline password guessing attack and key-compromise user impersonation attack, and is unable to ensure user un-traceability. To remedy such limitations and improve computational efficiency, we present a multi-server two-factor authentication scheme using elliptic curve cryptography (ECC). Subsequently, employing heuristic analysis and Burrows–Abadi–Needham logic (BAN-Logic) proof, it is proven that the presented scheme provides security against all known attacks, and in particular provides user un-traceability and perfect forward security. Finally, appropriate comparisons with prevalent works demonstrate the robustness and feasibility of the presented solution in multi-server environments.

Keywords: multi-server; authentication; key agreement; elliptic curve cryptography (ECC); BAN-Logic; wireless sensor networks (WSNs)

1. Introduction

With the recent advancements in Internet and communication technology and the growing demand for sharing multiple data resources, secure and efficient communication between the involved stakeholders has become more essential in areas such as e-commerce, telecare medical information, distributed cloud storage systems, etc. Obviously, privacy protection has emerged as a vital issue for secure and trusted communication. For secure and effective communication over an insecure network, the involved parties are required to negotiate on a common session key beforehand. For such negotiations, authentication-and-key-agreement protocols serve as the only solution. The first password authentication with insecure communication was established by Lamport in 1981 [1]. Later, Frank et al. [2] presented an authentication protocol based on hypertext transport protocol in
1991. However, Yang et al. [3] identified that Frank’s proposal was insecure and provided an improved solution in 2005. In order to present a secure and efficient authentication and key agreement protocol, in the following decade, many single-, two-, and three-factor authentication protocols were constructed while employing RSA, discrete logarithm over general groups, elliptic curve cryptography (ECC), chaotic maps [4–22], etc. However, some security limitations are prevailing in these protocols. By analyzing a large number of authentication protocols, we found that such shortcomings are resulted due to either improper usage of the cryptographic primitives or design defects of the protocols.

In 2011, Awasthi et al. [23] showed that the protocol of Shen et al. [24] is prone to user impersonation attack. To remedy impersonation attack, Awasthi et al. put forward a refined time stamp-based authentication-and-key-agreement protocol. However, in that protocol, the adversary can easily obtain smart card and identity parameters through an open channel. In 2014, Huang et al. [25] pointed out that the scheme presented by Awasthi et al. is unable to resist against user impersonation attack, and overlooks the password updation stage. Moreover, we remark that Awasthi et al.’s scheme also fails to ensure user anonymity. Huang et al. proposed an enhanced time stamp-based two-factor remote user authentication protocol while incorporating RSA, and claimed that the scheme can resist various attacks. However, Amin et al. [26] proved that the proposal of Huang et al., is susceptible to impersonation, offline password guessing, and insider attacks, while also having an inefficient password updation stage. Keeping in view the limitations of Huang et al.’s proposal, Amin et al. presented an authentication-and-key-agreement mechanism based on RSA.

In a multi-server environment, users interact with multiple servers. To login with different identities and passwords in such an environment is troublesome for the users. To eliminate this problem, first, users and multiple servers are registered at the registration center (RC). Subsequently, users can make an authentication-and-key-agreement with multiple servers by utilizing the unique identity and password pair. A proposed architecture of the multi-server authentication system is depicted in Figure 1. In 2013, Pippel et al. [27] employed smart cards to present a robust multi-server authentication protocol and proved it to be resistant against various known attacks. In a subsequent work, Li et al. [28] identified that the protocol presented by Pippel et al. is unable to provide correct authentication. Moreover, it cannot withstand impersonation attack and insider attack. Afterwards, Li et al. designed an improved smart card authentication protocol and proved that it can withstand perfect forward secrecy, stolen smart-card attack, offline password guessing attack, and so on. Even so, Srinivas et al. [29] provided that Li et al.’s scheme is unable to resist insider attack, denial-of-service attack, and stolen smart-card attack, and cannot provide perfect forward secrecy. However, we remark that Li et al.’s scheme addresses perfect forward secrecy. As a solution, Srinivas et al. presented an improved two-factor authentication scheme for the same multi-server architecture with reduced computation and communication cost while claiming that their protocol is susceptible to various known attacks. To the best of our knowledge, most of the schemes cannot provide perfect forward secrecy and user un-traceability, and are susceptible to key-compromise user impersonation attack and offline password guessing attack. More precisely, once an authentication-and-key-agreement mechanism fails to ensure user un-traceability, the user’s entire whereabouts are exposed to the attacker. This provides a great deal of convenience for attackers to carry out more attacks. This proposal takes the schemes of Amin et al. and Srinivas et al. as examples to depict how an adversary traces the legal user, effectively guesses the correct password, or succeeds in obtaining the session key. These security flaws usually exist in wireless sensor networks (WSNs) as well [30–40]. Moreover, The methods of attacking and designing we use are very useful and effective in analyzing similar vulnerabilities and designing new protocols in WSNs, respectively.

1.1. Contributions

The key contributions of our proposal are listed as follows: (1) We prove that Amin et al.’s protocol fails to ensure perfect forward secrecy and user un-traceability, and is susceptible to key-compromise user impersonation attack and offline password guessing attack. (2) It is proven that Srinivas et al.’s
scheme fails to ensure user un-traceability, and is prone to key-compromise user impersonation attack and offline password guessing attack. (3) To overcome these limitations, we design a two-factor authentication-and-key-agreement scheme for multi-server architecture while incorporating ECC. (4) The presented scheme ensures perfect forward secrecy, user anonymity, and un-traceability. Moreover, it provides security against major attacks, including impersonation attack, offline password guessing attack, key-compromise user impersonation attack, etc. (5) The security analysis using Burrows–Abadi–Needham logic (BAN-Logic) provides that the proposed protocol ensures secured mutual authentication between a remote user and server.

Figure 1. The architecture of the multi-server authentication system.

1.2. Outline of This Paper

The remaining contents of the proposal are organized as follows: cryptographic primitive and attacker model are detailed in Section 2. The scheme of Amin et al., and its cryptanalysis are presented in Sections 3 and 4, respectively. Sections 5 and 6 provide the scheme of Srinivas et al., and its cryptanalysis, respectively. The improved version of the proposed scheme is provided in Section 7. The heuristic security analysis and BAN-Logic are presented in Sections 8 and 9, respectively. Section 10 details the security and performance comparisons. Finally, Section 11 contains the concluding remarks.

2. Preliminary

We take advantage of ECC to present a two-factor authentication scheme. The following section briefly introduces the collision-resistant cryptographic one-way hash function as well as some computationally infeasible problems, including the elliptic curve computational Diffie–Hellman Problem (ECCDHP) and the elliptic curve discrete-logarithm problem (ECDLP). Table 1 depicts some notations and descriptions that are used in the proposed scheme.

| Symbol | Description | Symbol | Description |
|--------|-------------|--------|-------------|
| RC     | Registration center | $S_j$  | Server      |
| $U_i$  | User        | $SC_i$ | Smart card of $U_i$ |
| $Id_i$ | Identification of user $U_i$ | $Pw_j$ | Password belonging to user $U_i$ |
| $r_i, a_i$ | Random numbers of $U_i$ | $p$    | Large prime |
| $Q_j = r_jP$ | Public key of $S_j$ | $r_j$  | Private key of $S_j$ |
| $c_{ij}, b_j$ | Random number of $S_j$ | $\oplus$ | The bitwise XOR operation |
| $\|$     | The string concatenation operation | $H(\cdot)$ | One-way hash function |
| $A$     | The malicious adversary | $SK_{ij}$ | Session key belonging to $U_i$ and $S_j$ |
2.1. Collision-Resistant One-Way Hash Function

Basically, the one-way hash function $H(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^n$ requires an input in the form of an arbitrary length binary string $x \in \{0, 1\}^*$, and yields a string in binary form $y = H(x) \in \{0, 1\}^n$. In brief terms, a cryptographic collision-resistant one-way hash function $H(\cdot)$ ensures the following:

1. Given $y \in \{0, 1\}^n$, it is difficult to determine the input $x \in \{0, 1\}^*$ within polynomial time.
2. It is difficult to determine $x' \in \{0, 1\}^*$ such that $H(x) = H(x')$, where $x' \neq x$.
3. It is difficult to uncover a pair $(x, x') \in \{0, 1\}^*$ such that $x' \neq x$ and $H(x) = H(x')$ could hold.

2.2. Intractable Problems in ECC

The elliptic curve equation over a finite field $F_p$ in ECC takes the form $E_p(a, b) : y^2 = x^3 + ax + b \pmod{p}$, where $4a^3 + 27b \neq 0 \pmod{p}$ and $a, b \in F_p$ [41].

1. ECDLP: The elliptic curve discrete-logarithm problem over elliptic curve $E_p(a, b)$ refers to computing $m \in F_p^*$ from $Q = mP$ for given $P, Q \in E_p(a, b)$.
2. ECCDHP: The elliptic curve computational Diffie–Hellman problem over elliptic curve $E_p(a, b)$ refers to computing $mnP$, given points $mP, nP \in E_p(a, b)$.

2.3. Adversary Model

According to [18,42–47], the capacities of $\mathcal{A}$ in authentication and key agreement schemes, which are used in cryptanalysis of Amin et al.’s scheme, Srinivas et al.’s scheme, and our proposed scheme, are listed as follows:

1. $\mathcal{A}$ is able to intercept, block, delete, modify, and resend the message contents through an open channel.
2. Because identity and password have low entropy, $\mathcal{A}$ can enlist all pairs of $(Pw_i, Id_i)$ simultaneously from $(D_{Pw}, D_{Id})$ within polynomial time, where $D_{Pw}$ and $D_{Id}$ refer to the space of passwords and identities in $D_{Pw}$ and $D_{Id}$, respectively.
3. $\mathcal{A}$ can either acquire $Pw_i$ of the $U_i$ via malicious device or reveal the information from $SC$, but is not permitted to use both methods together.
4. $\mathcal{A}$ can acquire a server’s private key while evaluating forward secrecy or key-compromise user impersonation attack.
5. $\mathcal{A}$ has the ability to reveal all parameters of the smart card when assessing stolen smart-card attack, offline password guessing attack, impersonation attack, forward secrecy, etc.

3. Brief Review of Amin et al.’s Proposal

This section provides a brief review of Amin et al.’s [26] authentication scheme for Session Initiation Protocol (SIP). The scheme presented by the authors comprises four stages: initialization, registration, login and authentication, and password updation. We omit the description of the password updation stage.

3.1. Initialization

$S$ takes two large primes $p$ and $q$ as secret parameters to calculate $n = p \times q$ as a public parameter. Afterwards, $S$ chooses a prime $e$ to obtain $d$ by computing $e \times d \equiv 1 \pmod{(p - 1)(q - 1)}$, such that $1 < e < (p - 1)(q - 1)$.

3.2. Registration

1. $U_i$ enters an identity $Id_i$ and password $Pw_i$. Subsequently, $U_i$ randomly picks up a number $r$ and calculates $PWr_i = H(Pw_i||u)$. Afterwards, $U_i$ transmits the registration request message $\{Id_i, PWr_i\}$ to $S$ via secure medium.
2. Upon receiving the request message \(\{Id_i, PWr_i\}\) from the new user \(U_i\), \(S\) calculates \(CID_i = H(Id_i || d)\), \(Reg_i = H(CID_i || PWr_i || Id_i)\), and \(Y_i = CID_i \oplus H(PWr_i || Id_i)\). Afterwards, \(S\) stores the contents \(\{Reg_i, Y_i, n, e, H(\cdot)\}\) in a new card SC and sends SC to \(U_i\).

3. Once obtaining SC, \(U_i\) stores \(u\) into SC.

3.3. Login and Authentication

1. To start the session with the \(S\), \(U_i\) inserts SC into a card reader and inputs their login details, including \(Id_i\) and \(Pw_i\). Subsequently, SC calculates \(PWr_i = H(Pw_i || r), CID_i = Y_i \oplus H(PWr_i || Id_i)\), and \(Reg_i = H(D||PWr_i || Id_i)\). Afterwards, it verifies the value of \(Reg_i\). In case of invalid values, the session is ended. Otherwise, SC randomly chooses a number \(N_i\), the current time stamp \(T_{u}\), and calculates \(D_i = H(CID_i || H(PWr_i || Id_i) || T_u || N_i)\) and \(L_i = (Id_i || D_i || N_i)^e \mod n\). Next, SC transmits the login request message \(\{L_i, Y_i, T_u\}\) to \(S\).

2. Upon receiving the login request from \(U_i\), \(S\) verifies the time stamp \(T_u\) corresponding to the current time. In the case of valid time stamp \(T_u\), it continues to execute the following steps. Otherwise, \(S\) decrypts \(L_i\) to obtain \((Id_i^* || D_i^* || N_i^*)\) and then checks whether \(CID_i^* = H(Id_i^* || d), H(PWr_i || Id_i)^* = Y_i \oplus CID_i^*\) and \(D_i^* = H(CID_i || H(PWr_i || Id_i)^* || T_u || N_i^*)\). Afterwards, \(S\) checks \(D_i^* = ?D_i^*\). After finishing this verification, \(S\) randomly selects a number and computes \(X_i = H(N_i || CID_i), Z_i = N_i \oplus N_2\). Finally, \(S\) transmits the respond message \(\{L_i, Y_i, T_u\}\) to SC via public channel.

3. Once receiving the response message from \(S\), SC checks the validity of \(T_u\). After finishing the verification, SC checks whether \(N_2^* = N_1 \oplus Z_i, X_i^* = H(N_2^* || CID_i)\) and verifies \(X_i^* = ?X_i\). If it holds, \(U_i\) accepts the response message. Finally, \(S\) and \(U_i\) calculate the session key: \(SK = H(N_1 || CID_i || N_2^*) = H(N_2^* || CID_i^* || N_2)\).

4. Limitations of Amin et al.’s Scheme

According to the adversary model presented in Section 2.3, in the following, we prove that Amin et al.’s scheme is unable to provide user un-traceability and perfect forward secrecy, and is prone to key-compromise user impersonation attack and offline password guessing attack.

4.1. User Un-Traceability

Observing the protocol of Amin et al., it can be found that the password is transmitted during the login request message stage. However, \(Y_i = CID_i \oplus h(PWr_i || Id_i)\) is a fixed value in SC, unless \(U_i\) changes their password during the password updation stage. Usually, the user does not change their password after every session. Therefore, \(U_i\) can be traced by the adversary using \(Y_i\). Hence, Amin et al.‘s protocol does not ensure user un-traceability.

4.2. Offline Password Guessing Attack

Offline password guessing attack is the main limitation for most of the presented proposals addressing authentication. If \(A\) somehow steals the SC of \(U_i\) and embeds the data \(\{Reg_i, Y_i, r\}\) in it, then the adversary \(A\) can perform the following steps to obtain \(Id_i\) and \(Pw_i\) of \(U_i\).

1. From the password dictionary space \(D_{pw}\), the adversary \(A\) randomly chooses the password \(PW^*\), and picks up the identity \(ID^*\) from the identity dictionary space \(D_{id}\).

2. \(A\) calculates \(PWr_i^* = h(PW^* || r)\).

3. \(A\) calculates \(CID_i^* = Y_i \oplus h(PWr_i^* || ID_i^*)\).

4. \(A\) calculates \(Reg_i^* = h(CID_i^* || PWr_i^* || ID_i^*)\).

5. To check the correctness of \(PW^*\) and \(ID^*\), \(A\) examines whether \(Reg_i^* = Reg_i\), where \(Reg_i\) belongs to SC of \(U_i\).

6. If the aforementioned equality holds, \(A\)’s guess results as successful. Otherwise, \(A\) repeats Steps 1–5 until it obtains the correct password and identity of \(U_i\).
From the aforementioned procedure, we find that the computational time complexity of offline password guessing attack is \( O(|\mathcal{D}_{\text{PW}}| \cdot |\mathcal{D}_{\text{ID}}| \cdot 3T_h) \), where \( |\mathcal{D}_{\text{PW}}|, |\mathcal{D}_{\text{ID}}| \), and \( T_h \) refer to the number of \( \mathcal{D}_{\text{PW}} \), the number of \( \mathcal{D}_{\text{ID}} \), and the performing time of hash function \( h(\cdot) \), respectively. According to [48–50], usually, \( |\mathcal{D}_{\text{ID}}| < |\mathcal{D}_{\text{PW}}| < 10^6 \). Therefore, the aforementioned attack is very efficient. Hence, Amin et al.’s protocol is unable to resist key-compromise user impersonation attack. Actually, the verified data \( \mathcal{R}_{\text{Reg}} \) are stored in \( U_l \)’s smart card, which is the main reason for the success of the above attack. By computing \( \mathcal{R}_{\text{Reg}} \), the smart card is able to check the correct login of the legal user. Moreover, it also gives \( A \) the chance to guess password and identity. Since the identity and password have low entropy in such scenarios, \( A \) can guess them successfully within polynomial time.

4.3. Lacks of Perfect Forward Secrecy

Assume that the \( A \) obtains the long term private key \( d \) of \( S \) and eavesdrops the transmitted message \( \{L_{i}, Y_{i}, T_{u}\}, \{X_{i}, Z_{i}, T_{s}\} \). Having that information, \( A \) can easily calculate two random numbers \( \{N_1, N_2\} \). \( A \) undergoes the following procedure to compute \( SK \) between \( U_l \) and \( S \).

1. The adversary \( A \) computes \( (L_i)^d \mod n = (ID_{i}||D_{i}||N_1) \) to obtain \( \{ID_{i}, N_1\} \).
2. \( A \) computes \( CID_{i} = h(ID_{i}||d) \).
3. \( A \) computes \( N_2 = Z_{i} \oplus N_1 \).
4. \( A \) computes \( SK = h(N_1||CID_{i}||N_2) \).

The computational time overhead of the aforementioned attack is \( O(2T_h + T_c + T_{\text{eor}}) \), where \( T_c \) and \( T_{\text{eor}} \) are the running time of modular exponentiation and exclusive-or operation, respectively. Therefore, the protocol of Amin et al. does not ensure perfect forward secrecy. This problem can be solved by adding an operation of public key cryptography, which slightly increases the computation load. However, it is a feasible approach in terms of the trade-off between security and practicality.

4.4. Key-Compromise User Impersonation Attack

If the long-term private key \( d \) of \( S \) is revealed to the adversary \( A \) in Amin et al.’s protocol, \( A \) can impersonate the legitimate user \( U_l \) to \( S \) as follows:

1. \( A \) computes \( (L_i)^d \mod n = (ID_{i}||D_{i}||N_1) \), and subsequently calculates \( CID_{i} = h(ID_{i}||d) \) and \( A = H(\mathcal{PWR}_{i}||ID_{i}) = Y_{i} \oplus CID_{i} \).
2. \( A \) obtains the login request message \( \{L_{i}, Y_{i}, T_{u}\} \) of \( U_l \), randomly selects a number \( N_{a} \), and computes \( D'_{i} = H(CID_{i}||A||T'_{u}||N_{a}) \); \( L'_{i} = (ID_{i}||D'_{i}||N_{a})^e \mod n \). Afterwards, \( A \) transmits the forged request message \( \{L'_{i}, Y_{i}, T'_{u}\} \) to \( S \).
3. Upon receiving the forged message, obviously \( S \) can verify it successfully. Thus, \( S \) randomly provokes a number \( N_{a}' \), and computes \( X'_{i} = H(N_{a}'||CID_{i}) \) and \( Z'_{i} = N_{a} \oplus N_{a}' \). Finally, \( S \) sends \( \{X'_{i}, Z'_{i}, T_{s}\} \) to \( A \).
4. Upon receiving the response from \( S \), \( A \) calculates \( N_{a}'' = N_{a} \oplus Z'_{i} \). Finally, the server \( S \) believes that \( SK = H(N_{a}'||CID_{i}||N_2) \) is the common session key between a legitimate user and itself. However, in actual terms, \( A \) acts as \( U_l \).

Therefore, Amin et al.’s protocol is unable to resist key-compromise user impersonation attack.

5. Review of Srinivas et al.’s Scheme

The following section reviews Srinivas et al.’s protocol [29] comprising four steps: initialization, registration, login and authentication, and password updation stage.

5.1. Initialization

The trusted registration center \( RC \) during this stage selects a 1024-bit large prime \( p \), generates \( g \in Z_{p}^{*} \), chooses a one-way hash function \( H(\cdot) : \{0,1\}^n \rightarrow Z_{p}^{*} \), and randomly picks a number \( mk \) as the master secret key.
5.2. Registration Process

5.2.1. Server Registration

$S_j(1 \leq j \leq k)$ chooses a unique identity $SId_j$ and sends $SId_j$ to $RC$ through a secure-medium. Upon receiving $SId_j$, $RC$ calculates $r_j = H(SId_j || mk)$, and sends $\{r_j, p, g, H(\cdot)\}$ to $S_j$ through a secure medium.

5.2.2. User Registration

First, a new user $U_i$ selects $Id_i$, $PW_i$, and randomly chooses a number $r_i$. Subsequently, the user calculates $UId_i = H(Id_i || r_i), R PW_i = H(PW_i || r_i)$ and sends $\{UId_i, R PW_i\}$ to $RC$. Upon receiving the registration request, $RC$ calculates $v_{ij} = H(r_j || UId_i), s_{ij} = v_{ij} \oplus R PW_i$. Afterwards, $RC$ sends $U_i$ a new smart card $SC_i$ containing $\{s_{i1}, s_{i2}, \ldots, s_{ik}, p, g, H(\cdot)\}$ through a secure medium. Finally, upon receiving $SC_i$ from $RC$, $U_i$ inputs $B_i = r_i \oplus H(Id_i || PW_i)$ to $SC_i$.

5.3. Login and Authentication

1. $U_i$ inserts $SC_i$ into a card reader and inputs $Id_i$ and $PW_i$. $SC_i$ checks $r_i = B_i \oplus H(Id_i || PW_i)$, $UId_i = H(Id_i || n)$ and $R PW_i = H(PW_i || r_i)$. Afterwards, $SC_i$ randomly generates a number $a$, chooses the current time stamp $T_j$, and calculates $X_i = g^a \mod p, v_{ij} = s_{ij} \oplus H(UId_i || R PW_i)$ and $h_{ij} = H(v_{ij} || UId_i || SId_j || T_i || X_i)$. Subsequently, $SC_i$ transmits the login request message $\{UId_i, X_i, h_{ij}, T_j\}$ to $S_j$.

2. $S_j$ receives the request message from $U_i$, figures out $h_{ij}^* = H(H(r_j || UId_i) || UId_i || SId_j || X_i || T_i)$, and checks $h_{ij} = h_{ij}^*$. $S_j$ terminates the login request if the expression does not hold. Apart from that, $S_j$ picks a random number $b$ and calculates $Y_j = g^b \mod p, z_{ij} = (X_i)^b \mod p$. Afterwards, $S_j$ picks the current time stamp $T_j$ and computes $SK_{ji} = H(UId_i || SId_j || T_i || h_{ij}^* || T_j || z_{ij})$ and $R_j = H(UId_i || T_i || H(r_j || UId_i) || T_j || SK_{ji} || Y_j)$. Finally, $S_j$ sends the response message $\{Y_j, R_j, T_j\}$ to $S_i$.

3. On receiving the response message, $S_i$ figures out $z_{ij} = (Y_j)^a \mod p, SK_{ij} = H(UId_i || SId_j || T_i || h_{ij} || T_j || z_{ij})$, and $R_j^* = H(UId_i || T_i || v_{ij} || T_j || SK_{ji} || Y_j)$. Subsequently, $SC_i$ checks $R_j^* = R_j$ and terminates this login request if the expression does not hold. Otherwise, $SC_i$ calculates $R_j = H(UId_i || X_i || Y_j || SK_{ji} || H(r_j || UId_i))$ and checks $R_j^* = R_j$. After successful accomplishment of all steps, $S_j$ and $U_i$ believe that they have the common session key $SK_{ji} = SK_{ij}$.

5.4. Password Update Stage

After the authentication session between $SC_i$ and targeted server $S_j$, $U_i$ inputs $Id_i, PW_i$, and a new password $PW_{\text{new}}$. Subsequently, $SC_i$ calculates $B_{\text{new}} = r_i \oplus H(Id_i || PW_{\text{new}})$ and $s^*_{ij} = s_{ij} \oplus H(UId_i || H(PW_i || r_i)) \oplus H(UId_i || H(PW_{\text{new}} || r_i))$, where $1 \leq j \leq k$. Afterwards, $SC_i$ replaces $\{s_{i1}, s_{i2}, \ldots, s_{ik}, B_i\}$ with $\{s^*_{i1}, s^*_{i2}, \ldots, s^*_{ik}, B_{\text{new}}\}$.

6. Limitations of Srinivas et al.’s Protocol

According to the adversary model presented in Section 2.3, we present some possible attacks for Srinivas et al.’s protocol, including key-compromise user impersonation attack, offline password guessing attack, and lack of user un-traceability. The details are described in the following sections.

6.1. Offline Password Guessing Attack

Assume that $A$ extracts the information $\{s_{i1}, s_{i2}, \ldots, s_{ik}, B_i, p, g, H(\cdot)\}$ of $SC_i$ by side-channel attack. Now, $A$ can execute the following steps to get the correct identity $Id_i$ and password $PW_i$ of user $U_i$ in polynomial time.
1. From the password dictionary space $\mathcal{D}_{Pw}$, the adversary $A$ chooses the password $PW^*$, and picks up the identity $Id^*$ from the identity dictionary space $\mathcal{D}_{Id}$.
2. $A$ computes $n^* = B_i \oplus H(Id||PW_i)$.
3. $A$ computes $RPw_i = H(Pw_i||n^*)$.
4. $A$ computes $v_{ij}^* = s_{ij} \oplus H(UId_i||RPw_i)$.
5. $A$ computes $h_{ij}^* = H(v_{ij}^*||UId_i||SId_i||X_i||T_i)$.
6. $A$ verifies whether $h_{ij}^* = h_{ij}$, where $h_{ij}$ is acquired from smart card of $U_i$.
7. If it holds, then $PW^*$ and $Id^*$ is the correct identity and password pair. Otherwise, $A$ repeats Steps 1–6 until it obtains the correct identity and password of $U_i$.

We determine the computational time complexity of the aforementioned attack algorithm. That is, 

$$\mathcal{O}(|\mathcal{D}_{Pw}| \times |\mathcal{D}_{Id}| \times 4T_h),$$

where $|\mathcal{D}_{Pw}|$, $|\mathcal{D}_{Id}|$, and $T_h$ are the number of $\mathcal{D}_{Pw}$, the number of $\mathcal{D}_{Id}$, and the time to compute hash function $h(\cdot)$, respectively. According to [48–50], usually, $|\mathcal{D}_{Id}| < |\mathcal{D}_{Pw}| < 10^6$. Therefore, the offline password guessing attack is very efficient. Thus, Srinivas et al.’s protocol is not resistant against offline password guessing attack.

6.2. Lack of User Un-Traceability

It can be observed from Srinivas et al.’s protocol that the attacker can get $UId_i$ transmitted within the login request message. Since $UId_i = H(Id_i||r_i)$ is a fixed value, where $Id_i$ and $r_i$ are invariable, unless the user $U_i$ changes their password during the password updation stage, any adversary can trace the user $U_i$ by using $UId_i$. Therefore, Srinivas et al.’s protocol cannot provide user un-traceability.

6.3. Key-Compromise User Impersonation Attack

If the long-term private key $r_j$ of $S_j$ is revealed to $A$ in Srinivas et al.’s protocol, then $A$ can adopt the following actions to impersonate the legitimate $U_i$ to $S_j$.

1. $A$ intercepts the login request message \{UId$_i$, X$_i$, h$_{ij}$, T$_i$\} of $U_i$, and calculates $v_{ij} = H(r_j||UId_i)$.
2. $A$ randomly selects a number $a'$ to compute $X'_i = g^{a'} \mod p$, $h'_{ij} = H(v_{ij}||UId_i||SId_i||X'_i||T'_i)$. Afterwards, $A$ sends the forged login request message \{UId$_i$, X’$_i$, h’$_{ij}$, T’$_i$\} to $S_j$.
3. Obviously, the forged message can pass the verification of $S_j$. Thus, $S_j$ randomly chooses a number $b'$ to compute $Y'_j = g^{b'} \mod p$, $z'_{ij} = (X'_i)^{b'} \mod p$. Subsequently, $S_j$ chooses the current time stamp $T'_j$ to compute $SK'_{ij} = H(UId_i||SId_i||T'_j||h'_{ij}||T'_i||z'_{ij})$ and $R'_{ij} = H(UId_i||T'_i||H(r_j||UId_i)||T'_j||SK'_{ij}||Y'_j)$. Finally, $S_j$ sends the response message \{Y’$_j$, R’$_{ij}$, T’$_j$\} to $A$.
4. On receiving the response message, $A$ figures out $z'_{ij} = (Y'_j)^{a'} \mod p$, $SK'_{ij} = H(UId_i||SId_i||T'_j||h'_{ij}||T'_i||z'_{ij})$. Subsequently, $A$ calculates $R''_{ij} = H(UId_i||X'_i||Y'_j||SK'_{ij}||v_{ij})$ and transmits it to $S_j$ through a public channel.
5. $S_j$ receives $R''_{ij}$, computes $R''_{ij} = H(UId_i||X'_i||Y'_j||SK'_{ij}||H(r_j||UId_i))$, and checks whether $R''_{ij} = ?R'_{ij}$.

After finishing all steps successfully, $S_j$ believes that it holds the common session key $SK'_{ij} = SK'_{ij}$ with $U_i$. Actually, however, $A$ plays as $U_i$. Thus, $A$ successfully impersonated $U_i$ to $S_j$ under the condition that the long-term private key of the server was leaked.

Therefore, Srinivas et al.’s protocol is prone to key-compromise user impersonation attack.

7. The Improved Scheme

The following section presents an improved mutual authentication protocol that gets motivation from Srinivas et al.’s [29] scheme to incorporate ECC. The presented solution not only remedies the limitations of Amin et al.’s [26] and Srinivas et al.’s [29] schemes, but also ensures mutual authentication and is resistant to many known attacks. The presented scheme comprises five stages: initialization,
server registration, user registration, authentication-and-key-agreement, and password updating. The notations of the presented scheme are listed in Table 1. Figures 2–4 depict the registration and authentication process of the proposed protocol.

**Figure 2.** Server registration.

| Server ($S_j$) | Secure Channel | Registration Center (RC) |
|-----------------|----------------|--------------------------|
| **Registration Phase:** | | |
| Inputs $SId_j$ | | Generates $s_j \in Z_p^*$ |
| | | Computes $r_j = H(SId_j || mk || s_j)$ |
| | | and selects a random number $c_j \in Z_p^*$ |
| Stores $\{r_j, Q_j, P, c_j, H(\cdot)\}$ | $r_j, Q_j, P, c_j, H(\cdot)$ | |

**Figure 3.** User registration.

| User ($U_i$) | Secure Channel | Registration Center (RC) |
|---------------|----------------|--------------------------|
| **Registration Phase:** | | |
| Inputs $Id_i, P wi$ | | Chooses an integer $2^4 \leq n_0 \leq 2^8$ |
| | | Computes $A_i = H((H(Id_i) \oplus RP wi) \mod n_0)$ |
| | | and generates a random number $c_j \in Z_p^*$ |
| | | Computes $v_{ij} = H(r_j || Id_i || c_j)$ |
| | | and selects a random number $c_j \in Z_p^*$ |
| | | $s_{ij} = v_{ij} \oplus H(Id_i || RP wi)$ |
| | | New smart card: $SC_i = \{A_i, s_{ij} (1 \leq j \leq k), n_0, Q_j, P, H(\cdot)\}$ |

**Figure 4.** Login and authentication.

| User ($U_i$) | Public Channel | Server ($S_j$) |
|---------------|----------------|----------------|
| **Login and Authentication Phase:** | | |
| Inserts $SC_i$, $Id_i, P wi$ | $P Id_i = H(Id_i || P wi || ri)$ | Computes $X_i^* = r_j X_i$ |
| | Computes $A_i = H((H(Id_i) \oplus RP wi) \mod n_0)$ | $Id_i^* = P Id_i \oplus H(X_i^*)$ |
| | Checks $A'_i = A_i$ | $v_{ij}^* = H(r_j || Id_i^* || c_j)$ |
| | Computes $v_{ij} = s_{ij} \oplus H(Id_i || RP wi)$ | Checks $h_{ij}^* = h_{ij}$ |
| | Chooses a random number $a_i \in Z_p^*$ | Generates $b_j \in Z_p^*$ |
| | Computes $X_i = a_i P$ | Computes $Y_j = b_j P$ |
| | $X_{ij} = a_i Q_j$ | $z_{ij} = b_j X_i$ |
| | $P Id_i = H(\{X_{ij}, IId_i\} || Id_i^* || X_i^* || X_i)$ | $SK_{ij} = H(Id_i || SId_i || v_{ij} || z_{ij} || X_{ij}^*)$ |
| | | $R_{ij} = H(P Id_i || v_{ij} || SK_{ij} || X_i || Y_j)$ |

The common session key: $SK = SK_{ij} = SK_{ij}^*$
7.1. Initialization

RC chooses an elliptic curve $E_p(a, b)$ from $F_p$, where $p$ is a 160-bit-long prime number. Afterwards, RC selects a fixed point $P \neq \infty \in E_p(a, b)$, and one-way hash function $H() : \{0,1\}^* \rightarrow Z_p^*$, and randomly picks a number as $mk$.

7.2. Server Registration

1. $S_i$ chooses an identity $SId_i$ and transmits it to RC via a secure-medium.
2. RC receives the registration message, randomly generates a number $s_i \in Z_p^*$, and computes $r_i = H(SId_i || mk || s_i), Q_i = r_iP$. Subsequently, RC randomly generates a number $c_i$ for $S_i$. Finally, RC sends $\{r_i, Q_i, c_i, P, H(\cdot)\}$ to $S_i$ through secure-medium.
3. $S_i$ stores $\{r_i, Q_i, c_i, P, H(\cdot)\}$ in its database.

7.3. User Registration

After the successful registration of $U_l$ with RC, $U_l$ can communicate with any server $S_j (1 \leq j \leq k)$.

1. $U_l$ selects $Id_l, Pw_l$, and randomly generates a number $r_i \in Z_p^*$ to compute $RPw_l = H(Id_l || Pw_l || r_i)$. Afterwards, $U_l$ transmits the registration request message $\{Id_l, RPw_l\}$ to RC through a secure medium.
2. Upon receiving the registration message, RC randomly generates numbers $r_s \in Z_p^*, 2^4 \leq n_0 \leq 2^8$, and computes the following: $A_i = H((H(Id_i) \oplus RPw_l) \mod n_0), v_{ij} = H(r_i || Id_i || c_i), s_{ij} = v_{ij} \oplus H(Id_i || RPw_l)$, where $(1 \leq j \leq k)$. Afterwards, RC inserts $\{A_i, s_{ij}(1 \leq j \leq k), n_0, Q_i, P, H(\cdot)\}$ into a new $SC_i$, and sends it to $U_l$ through secure-medium.
3. $U_l$ stores $r_s$ in $SC_i$.

7.4. Login and Mutual Authentication

$U_l$ initiates the login and authentication request for sending to $S_j$ by performing the following steps.

1. $U_l$ inserts $SC_i$ into a card reader and inputs $Id_l, Pw_l$. $SC_i$ computes $RPw_l = H(Id_l || Pw_l || r_i)$, and subsequently calculates $A_i^* = H((H(Id_i) \oplus RPw_l) \mod n_0)$. Afterwards, $SC_i$ inspects the correctness of $A_i^*$ while comparing it with the value of $A_i$ sorted in $SC_i$. If $A_i^* = A_i$, $Id_l$ and $Pw_l$ are validated. Otherwise, the session is expired. $SC_i$ continues to compute $v_{ij} = s_{ij} \oplus H(Id_i || RPw_l)$, and randomly selects a number $a_i \in Z_p^*$ to calculate the following: $X_i = a_iP, X_{ij} = a_iQ_j, PId_i = H(X_i) \oplus Id_l, h_l = h(v_{ij} || Id_l || SId_l || X_{ij} || X_i)$. Finally, $U_l$ transmits the request $\{PId_i, X_i, h_l\}$ to $S_j$ via an open channel.
2. After receiving $\{PId_i, X_i, h_l\}$, $S_j$ calculates $X_{ij}^* = r_iX_i, Id_j^* = PId_i \oplus H(X_{ij}^*)$ and $v_{ij}^* = H(r_j || 1d_i^* || c_i)$. Afterwards, $S_j$ computes $h_j^* = h(v_{ij}^* || Id_j^* || SId_j || X_{ij} || X_i)$. Then, $S_j$ verifies $h_j^* = h_i$. In the case of invalidation, $S_j$ terminates the session and sets the counter $N = 1$. $S_j$ keeps suspending the card until $U_l$ registers again if $N$ surpasses some threshold mark (e.g., 8). Otherwise, $S_j$ randomly selects a number $b_j$ to compute $Y_j = b_jP, z_{ij} = b_jX_i$, $SK_{ij} = H(Id_i || SId_l || v_{ij} || z_{ij} || X_{ij}^*)$, and $R_j = H(PId_i || v_{ij} || SK_{ij} || X_i || Y_j)$. Finally, $S_j$ sends the response message $\{Y_j, R_j\}$ to $U_l$ via open channel.
3. Upon receiving the respond message $\{Y_j, R_j\}$, $U_l$ computes $z_{ij}^* = a_iY_j, SK_{ij}^* = H(Id_l || SId_l || v_{ij} || z_{ij}^* || X_{ij}),$ and $R_j^* = H(PId_i || v_{ij} || SK_{ij}^* || X_i || Y_j)$. Subsequently, $U_l$ checks whether $R_j^* \neq R_j$. The session is aborted if these are not equal. Otherwise, $S_j$ is authenticated by $U_l$, and $U_l$ accepts $SK_{ij}$. Afterwards, $U_l$ computes $R_i = H(v_{ij} || X_i || Y_j || SK_{ij}^* || Id_l)$. Finally, $U_l$ transmits the challenge message $R_i$ to $S_j$ through an open channel.
4. Upon receiving the challenge message from $U_l$, $S_j$ computes $R_i^* = H(v_{ij} || X_i || Y_j || SK_{ij}^* || Id_l)$ and verifies whether $R_i^* \neq R_i$. If these are equal, then $U_l$ is authenticated successfully.

Finally, both $U_l$ and $S_j$ share the common session key $SK = SK_{ij}^* = SK_{ij}$. 


7.5. Password Updation

$U_i$ is able to change their password whenever they want, for which $U_i$ and $SC_i$ have to undergo the following procedure:

1. $U_i$ inserts the $SC_i$ into a card reader and inputs $Id_i$, current password $Pw_i$, and password to be updated $Pw_i^\ast$.
2. $SC_i$ computes $RPw_i = H(Id_i||Pw_i||r_i)$, and $A_i^\prime = H((H(Id_i) \oplus RPw_i) \mod n_0)$. Afterwards, $SC_i$ checks whether $A_i^\prime \neq A_i$. In case of inequality, $SC_i$ refuses $U_i$ to update the password.
3. Apart from that, $SC_i$ randomly selects a number $r_i^\ast$ to compute $RPw_i^\ast = H(Id_i||Pw_i^\ast||r_i^\ast)$, $s_i^\ast = s_{ij} \oplus H(Id_i||RPw_i^\ast) \oplus H(Id_i||RPw_i^\ast)$. Subsequently, $SC_i$ computes $A_i^\ast = H((H(Id_i) \oplus RPw_i^\ast) \mod n_0)$. Finally, $SC_i$ replaces $r_i, A_i, s_{ij}$ with $r_i^\ast, A_i^\ast, s_i^\ast$, respectively.

Remark: As Amin et al.’s scheme and Srinivas et al.’s scheme are vulnerable to offline password guessing attack and key-compromise user impersonation attack and cannot provide user un-traceability, and because Amin et al.’s scheme cannot provide perfect forward secrecy, in the proposed scheme: (1) we employ “honey words” + “fuzzy-verifiers” to resist against offline password guessing attack [42]; (2) according to [47], to provide perfect forward secrecy, we use public key cryptosystems (e.g., ECC); (3) we store a secret parameter $c_i$ in the server database which cannot be compromised by the adversary in order to resist key-compromise user impersonation attack; and (4) to provide user un-traceability, we deploy a dynamic identity technique via a public key algorithm, that is, $PId_i$.

8. Security Inspection

This section provides the details of how the presented protocol ensures the security against all known attacks, including key-compromise user impersonation attack and offline password guessing attack. Further, it also offers more comprehensive security features, in particular, user un-traceability and perfect forward secrecy under the capabilities of the adversary that were introduced in Section 2.3.

8.1. User Un-Traceability and Anonymity

During the login authentication stage, $Id_i$ is not sent through the public channel. Even if $A$ intercepts the login request messages $\{PID_i, X_i, h_{ij}\}$ from the public channel, $A$ still cannot extract $Id_i$ from $PId_i$, because $PId_i$ is protected by $H(X_{ij})$ and is a dynamic identity. Thus, the proposed scheme provides the user un-traceability and anonymity.

8.2. Stolen Smart-Card Attack

In the proposed scheme, even if $A$ steals $SC_i$ of $U_i$, then $A$ can extract the parameters $\{A_i, r_i, s_{ij}(1 \leq j \leq k), n_0, Q_j, P, H(\cdot)\}$ stored in $SC_i$ utilizing power analysis technology, and captures the transmitted message over a public channel. However, as per the following details, $A$ cannot execute any attack. Thus, the presented protocol is secured against stolen smart-card attack.

8.3. Offline Password Guessing Attack

Assuming that $A$ steals $SC_i$ and extracts $\{A_i, r_i, s_{ij}(1 \leq j \leq k), n_0, Q_j, P, H(\cdot)\}$ stored in it. $A$ intercepts all messages $\{PId_i, X_i, h_{ij}\}, \{Y_j, R_j\}, \{R_i\}$ over a public channel. If $A$ guesses an ID $Id_i^\prime$ and a password $Pw_i^\prime$, $A$ can calculate $RPw_i^\prime = H(Id_i^\prime||Pw_i^\prime||r_i)$, and then figures out $A_i^\prime = H((H(Id_i) \oplus RPw_i) \mod n_0)$. Afterwards, $A$ examines whether $A_i^\prime \neq A_i$. According to [42], $A$ can obtain the reduced password guessing space of size $|D|/n_0$, where $D$ is the space of passwords. Further, $A$ can guess the correct password only by online password guessing. However, $S_j$ prevents this guessing by using a login request threshold value (e.g., 8). Once the number of online guesses exceeds the threshold value, $S_j$ will terminate communication and suspend $SC_i$ until $U_i$ registers again. Therefore, the presented scheme offers resistance against offline password guessing attack.
8.4. Privileged Insider Attack

If an internal attacker eavesdrops the registration information \{Id_i, Rp_i\} during user registration, \(A\) is unable to get \(Pw_i\), because it is secured by one-way hash function \(H(\cdot)\) as well as with random number \(r_i\). Thus, the presented scheme is immune to the privileged insider attack.

8.5. Key-Compromise User Impersonation Attack

If the adversary steals the long-term private key of the server, it is still unable to impersonate the user to the server. This kind of attack is referred to as a key-compromise user impersonation attack. In the presented protocol, even if \(r_i\) of \(S_i\) is revealed to \(A\), still \(A\) cannot determine \(v_{ij} = H(r_j||Id_i||c_j)\), because \(A\) is unable to obtain the random number \(c_j\). Therefore, \(A\) cannot forge the login request message \{\(h_{ij}\)\}, and therefore cannot be authenticated by \(S_j\). That is, \(A\) cannot impersonate \(U_i\). Thus, the presented protocol is unsusceptible to key-compromise user impersonation attack. Further, it implies that the presented scheme ensures resistance against user impersonation attack.

8.6. Server Impersonation Attack

\(A\) intercepts the response message \{\(Y_j, R_j\)\} if \(A\) tries to make a server impersonation attack. \(A\) randomly generates a number \(b'_j\) to compute \(Y'_j = b'_jP, z'_j = b'_jX_j\). Afterwards, \(A\) tries to compute \(SK'_{ij}\) and \(R'_j\). Since \(A\) does not know \(v_{ij}\) and \(X'_j\) computed by the secret key \{\(r_j, c_j\)\} of \(S_j\), \(A\) is unable to calculate \(SK'_{ij}\) and cannot forge \(R_j\). Thus, \(A\) cannot carry out the server impersonation attack.

8.7. Replay Attack

If \(A\) intercepts the login message \{\(Pl_{di}, X_i, h_{ij}\)\} from \(U_i\), and wants to replay this message to \(S_j\). This replay attack is easily captured by inspecting the freshness of \(X_i\) in the presented scheme, where \(X_i = a_iP, and a_i\) is a random number. Similarly, replaying the challenging message and response message is detected by either \(U_i\) or \(S_j\). Thereupon, it is inferred that the presented protocol is immune to replay attack.

8.8. Known Key Security

Suppose that \(A\) compromises the previous session key \(SK_{ij} = H(Id_i||SId_j||v_{ij}||z_{ij}||X_{ij})\) between \(U_i\) and \(S_j\). However, the next session key \(SK'_{ij}\) will be computed by new random numbers \(a'_j\) and \(b'_j\). That is, \(SK'_{ij} = H(Id_i||SId_j||v_{ij}||z'_{ij}||X'_{ij})\). To calculate the new session key, \(A\) has to compute \(a'_j, b'_j, a'_j b'_j P, a'_j Y_j, b'_j X_j\) from \(X'_j, Y'_j\). However, this is computationally infeasible for \(A\) because of ECDLP and ECCDHP. Therefore, the presented scheme offers known key security.

8.9. Mutual Authentication

In the proposed scheme, only the legitimate \(h_{ij}\) and \(R_i\) can be verified by \(S_j\), and only the legitimate \(R_j\) can be verified as the user \(U_i\). That is, the proposed scheme allows \(S_j\) and \(U_i\) to authenticate each other. Thus, the presented protocol ensures mutual authentication between a legitimate \(U_i\) and \(S_j\).

8.10. Man-in-the-Middle Attack

It is impossible for \(A\) in the proposed scheme to compute the correct login request and challenge message. Therefore, \(A\) cannot be authenticated by the server. Moreover, \(A\) is unable to calculate the correct response message, and thus \(A\) cannot pass the user verification. It is therefore inferred that the proposed scheme is immune to man-in-the-middle attack.

8.11. Denial-of-Service Attack

If \(U_i\) wants the login authentication in the proposed scheme, it must input the correct \(Id_i\) and \(Pw_i\) to pass the verification of SC. If \(A\) inputs wrong \(Id_i\) and \(Pw_i\) into SC, \(A\) is unable to compute the correct
login request message. Moreover, if $U_i$ wants to update the password, it has to pass the verification of SC. An incorrect or previous password cannot pass the verification. Therefore, the proposed scheme ensures resistance against denial-of-service attack.

8.12. Perfect Forward Secrecy

Suppose that $r_j$ of $S_j$ is compromised and $A$ acquires $r_i$, $Id_i$, and $Pw_i$. To calculate the correct $SK_{ij} = H(Id_i||SId_j||v_{ij}||z_{ij}||X_{ij})$, $A$ is required to calculate $z_{ij}, X_{ij}$. However, it is impossible for $A$ to compute $z_{ij}, X_{ij}$ because of ECDLP and ECCDHP. Thus, $A$ is not capable of figuring out $SK_{ij}$. Therefore, the presented protocol ensures perfect forward secrecy.

9. BAN-Logic Proof

BAN is a logic of belief. The intended use of BAN is to analyze authentication protocols by deriving the beliefs that honest principals correctly executing a protocol can come to as a result of the protocol execution. For example, a user might come to believe that a session key they have negotiated with a server is a good key for a future session [51]. This section incorporates the BAN-Logic [52] to prove the session key agreement between user $U_i$ and server $S_j$ after the execution of the improved scheme. BAN-Logic notations and Basic BAN-Logic postulates are described in Tables 2 and 3.

Table 2. Burrows–Abadi–Needham logic (BAN-Logic) notations.

| Symbol | Description |
|--------|-------------|
| $A|\equiv X$ | $A$ has trust on $X$ |
| $A<? X$ | $A$ acquires/observes $X$ |
| $A|\sim X$ | $A$ sends $X$ $X$ (or $A$ once called) |
| $A|\Rightarrow X$ | $A$ regulates $X$ |
| $\#(X)$ | $X$ is fresh |
| $A \leftrightarrow K \rightarrow B$ | $A$ and $B$ utilize shared key $K$ for communication |
| $(X,Y)_K$ | use $K$ as key to compute hash values of $X$ and $Y$ |
| $<X>_K$ | $X$ is exclusive or-ed with $K$ |

9.1. Idealized Scheme

The ideal form of the presented protocol is derived as follows:

**Message 1.** $U_i \to S_j$: $X_i, Id_i < X_i(1)\overset{u_{ij}}{\leftarrow_{S_j}} (Id_i || SId_j || X_{ij} || X_i) \overset{v_{ij}}{\leftarrow_{S_j}} (X_i, Y_j, U_i \leftrightarrow_{SK}^{S_j} S_j, Id_i $) $U_i \leftrightarrow_{SK}^{S_j} S_j$.

**Message 2.** $S_j \to U_i$: $Y_j, (PId_i, U_i \leftrightarrow_{SK}^{S_j} S_j, X_i, Y_j) \overset{v_{ij}}{\leftarrow_{S_j}}$.

9.2. Security Objectives

We prove that the improved scheme can satisfy the following objective:

**Objective 1.** $U_i \equiv S_j$.

**Objective 2.** $U_i \equiv (U_i \leftrightarrow_{SK}^{S_j} S_j)$.

**Objective 3.** $S_j \equiv U_i \equiv (U_i \leftrightarrow_{SK}^{S_j} S_j)$.

**Objective 4.** $S_j \equiv (U_i \leftrightarrow_{SK}^{S_j} S_j)$.

9.3. Initiative Premises

For the initial status of the proposed scheme, the following assumptions are made.

**IP 1.** $U_i \equiv \#(a_i)$. 
9.4. Proof Procedure

The main proof steps of the proposed scheme are presented below.

Step 1. From Message 2, it shows the following:

\[ U_i \leftarrow (P Id_i, U_i \xleftarrow{SK} S_j, X_i, Y_j) \] \( U_i \xleftarrow{v_j} S_j \).  

Step 2. From Step 1, IP 5, and the message-meaning rule, it illustrates the following:

\[ U_i \equiv S_j \mid \sim (P Id_i, U_i \xleftarrow{SK} S_j, X_i, Y_j). \]  

Step 3. From IP 1 and the freshness conjunctenation rule, the following can be inferred:

\[ U_i \equiv \exists (P Id_i, U_i \xleftarrow{SK} S_j, X_i, Y_j). \]  

Step 4. From Steps 2 and 3, the freshness rule, and the nonce verification rule, we obtain the following:

\[ U_i \equiv S_j \equiv (P Id_i, U_i \xleftarrow{SK} S_j, X_i, Y_j). \]  

Step 5. From Step 4 and the believe rule, we deduce the first objective as follows:

\[ U_i \equiv S_j \equiv (U_i \xleftarrow{SK} S_j) \quad \text{(Objective 1)}. \]  

Step 6. From Objective 1, IP 7, and the jurisdiction rule, we accomplish the second objective as follows:

\[ U_i \equiv (U_i \xleftarrow{SK} S_j) \quad \text{(Objective 2)}. \]  

Step 7. From Message 1, it indicates the following:

\[ S_j \leftarrow (X_i, Y_j, U_i \xleftarrow{SK} S_j, Id_i) \] \( U_i \xleftarrow{v_j} S_j \).
Step 8. From Step 7, IP 6, and the message meaning rule, the following can be inferred:

\[ S_j | \equiv U_i | \sim (X_i, Y_j, U_i \leftarrow SK \rightarrow S_j, Id_i) \].

Step 9. From IP 2 and the freshness conjunction rule, the following can be obtained:

\[ S_j | \equiv \#(X_i, Y_j, U_i \leftarrow SK \rightarrow S_j, Id_i) \].

Step 10. From Steps 8 and 9, the freshness rule, and the nonce-verification rule, we determine the following:

\[ S_j | \equiv U_i | \equiv (X_i, Y_j, U_i \leftarrow SK \rightarrow S_j, Id_i) \].

Step 11. From Step 10 and the believe rule, the third objective can be achieved as follows:

\[ S_j | \equiv U_i | \equiv U_i \leftarrow SK \rightarrow S_j \quad \text{(Objective 3)} \].

Step 12. From Objective 3, IP 8, and the jurisdiction rule, the fourth objective is accomplished as follows:

\[ S_j | \equiv (U_i \leftarrow SK \rightarrow S_j) \quad \text{(Objective 4)} \].

By accomplishing Objectives 1–4, both \( U_i \) and \( S_j \) believe that the \( SK \) is settled between them. Therefore, the proposed scheme ensures mutual authentication along with key agreement.

10. Performance Comparison

This section analyzes the computational and security performance of the presented scheme while comparing it with multiple schemes, including those of Awasthi et al. [23], Huang et al. [25], Amin et al. [26], Pippal et al. [27], Li et al. [28], and Srinivas et al. [29]. The exclusive-OR operation and string concatenation are usually neglected when comparing the computational cost. However, the following operations are considered: \( T_{me} \), the execution time of point multiplication operation; \( T_e \), the time for execution of modular exponentiation operation; \( T_h \), the running time of a hash operation; and \( T_{mm} \), the running time for modular multiplication operation. More precisely, we compare the experimental results of the aforementioned operations as performed by [53,54], where \( T_e, T_{me}, T_h, \) and \( T_{mm} \) take 3.85 ms, 2.226 ms, 0.0023 ms, and 0.001855 ms, respectively (Table 4). Following [53,54], the aforementioned operations were executed on a computing platform having Intel Pentium Dual Core E2200 2.20 GHz processor, the Ubuntu 12.04.1 LTS 32-bits operating system, and 2048 MB of RAM.

| Symbol | \( T_e \) | \( T_{me} \) | \( T_h \) | \( T_{mm} \) |
|--------|-----------|-------------|-----------|-------------|
| Time   | 3.85 ms   | 2.226 ms    | 0.0023 ms | 0.001855 ms |

In Table 5, we compare the schemes of [23,25–29] with the presented protocol in terms of security. In Table 5, we observe that [23,25–29] cannot provide \([C_1 - C_3, C_5]\) features. The scheme in [26] is still unable to provide perfect forward secrecy \([C_{12}]\), although the authors used RSA-based public cryptography. The proposed scheme fulfills all known security features \([C_1 - C_{12}]\). Thus, the presented scheme surpasses [23,25–29] in terms of security.
we remark that the presented solution is more feasible for practical multi-server environments in terms of usability and security. Afterwards, to address the limitations of prevalent works, we put forward an enhanced multi-server authentication-and-key-agreement scheme while proving that it cannot resist offline password guessing attack and key-compromised user impersonation attack. This paper first analyzes Amin et al.’s [26] scheme and proves that the considered scheme cannot provide perfect forward secrecy and user un-traceability, and is susceptible to offline password guessing attack.

Table 5. Comparison of security features.

| Features          | Awasthi et al. [23] | Huang et al. [25] | Amin et al. [26] | Pippal et al. [27] | Li et al. [28] | Srinivas et al. [29] | Proposed Scheme |
|-------------------|---------------------|-------------------|------------------|-------------------|----------------|---------------------|-----------------|
| C1                | No                  | No                | No               | No                | No             | No                  | Yes             |
| C2                | No                  | No                | No               | No                | No             | Yes                 | No              |
| C3                | No                  | No                | No               | No                | No             | Yes                 | Yes             |
| C4                | No                  | No                | Yes              | No                | No             | Yes                 | Yes             |
| C5                | No                  | No                | No               | No                | No             | Yes                 | Yes             |
| C6                | No                  | Yes               | No               | No                | Yes            | Yes                 | Yes             |
| C7                | Yes                 | Yes               | Yes              | Yes               | Yes            | Yes                 | Yes             |
| C8                | N/A                 | N/A               | Yes              | Yes               | Yes            | Yes                 | Yes             |
| C9                | Yes                 | Yes               | Yes              | Yes               | Yes            | Yes                 | Yes             |
| C10               | No                  | Yes               | Yes              | Yes               | No             | Yes                 | Yes             |
| C11               | No                  | No                | Yes              | No                | No             | Yes                 | Yes             |
| C12               | N/A                 | N/A               | No               | Yes               | Yes            | Yes                 | Yes             |

C1 provides user anonymity and un-traceability. C2 resists stolen smart-card attack. C3 resists offline password guessing attack. C4 resists privileged insider attack. C5 resists (key-compromised) user impersonation attack. C6 resists server impersonation attack. C7 resists replay attack. C8 provides known key security. C9 provides mutual authentication. C10 resists man-in-the-middle attack. C11 resists denial-of-service attack. C12 provides perfect forward secrecy.

Table 6. Comparison of computational complexity.

| Schemes           | Cost               | User Computation | Server Computation | Total               |
|-------------------|--------------------|------------------|--------------------|---------------------|
| Awasthi et al. [23]| 3T_c + 3T_mm + 2T_h| 3T_c + T_mm + 3T_h| 6T_c + 4T_mm + 5T_h| 23.1189 ms          |
| Huang et al. [25] | 2T_c + 2T_h        | 3T_c + 3T_h      | 5T_c + 5T_h        | 19.2615 ms          |
| Amin et al. [26]  | T_c + 6T_h         | T_c + 4T_h       | 2T_c + 10T_h       | 7.723 ms            |
| Pippal et al. [27]| 3T_c + T_mm + 4T_h| 4T_c + T_mm + 3T_h| 7T_c + 2T_mm + 7T_h| 26.9698 ms          |
| Li et al. [28]    | T_c + 5T_h         | 3T_c + 8T_h      | 4T_c + 13T_h       | 15.4299 ms          |
| Srinivas et al. [29]| 2T_c + 8T_h    | 2T_c + 4T_h      | 4T_c + 12T_h       | 15.676 ms           |
| Proposed scheme   | 3T_mc + 9T_h       | 3T_mc + 6T_h     | 6T_mc + 15T_h      | 13.3905 ms          |

Table 6 presents the computational cost of the schemes [23,25–29] and the proposed scheme for login and authentication. The computational cost of the proposed protocol is comparatively lower than the schemes in [23,25,27–29], but slightly higher than the scheme in [26]. However, according to Table 5, the scheme in [26] cannot address [C1−C3,C7,C12] security features. Thus, combining Tables 5 and 6, we remark that the presented solution is more feasible for practical multi-server environments in terms of the trade-off between usability and security.

11. Conclusions

This paper first analyzes Amin et al.’s [26] scheme and proves that the considered scheme cannot provide perfect forward secrecy and user un-traceability, and is susceptible to offline password guessing attack and key-compromised user impersonation attack. Second, we review Srinivas et al.’s [29] multi-server authentication scheme while proving that it cannot resist offline password guessing attack and key-compromised user impersonation attack, and is unable to ensure user un-traceability. Afterwards, to address the limitations of prevalent works, we put forward an enhanced multi-server two-factor authentication scheme. Heuristic analysis and BAN-Logic proof ensure that the presented scheme includes various known security features. The security and efficiency analyses display the robustness and efficiency of the presented scheme. Overall, the presented scheme is proven to be more feasible for multi-server authentication-and-key-agreement scenarios in various low-power networks. Moreover, the design and analysis methods in this paper can also be used for authentication protocols in IoT, WSNs, etc.
Author Contributions: Writing—original draft, G.X. (Guosheng Xu), S.Q., H.A., G.X. (Guoai Xu), Y.G., M.Z. and H.X.; and Writing—review and editing, G.X. (Guosheng Xu), S.Q., H.A. and G.X. (Guoai Xu).

Funding: This work was funded by the National Key Research and Development Program of China (No. 2018YFB0803600 and No. 2017YFB0801901) to Guoai Xu, and BUPT Excellent Ph.D. Students Foundation (No. CX2018312) to Shuming Qiu.

Acknowledgments: The authors are thankful to the Editor and anonymous reviewers for the generous feedback and constructive comments.

Conflicts of Interest: There are no conflicts of interest regarding the publication of this paper.

References

1. Lamport, L. Password authentication with insecure communication. Commun. ACM 1981, 24, 770–772. [CrossRef]
2. Franks, J.; Hallam-Baker, P.; Hostetler, J.; Lawrence, S.; Leach, P.; Luotonen, A. HTTP Authentication: Basic and Digest Access Authentication. IETF RFC 1999, 2617, 1–34.
3. Yang, C.; Wang, R.; Liu, W. Secure authentication scheme for session initiation protocol. Comput. Secur. 2005, 24, 381–386. [CrossRef]
4. Khan, M.K. Fingerprint Biometric-based Self-Authentication and Deniable Authentication Schemes for the Electronic World. IETE Tech. Rev. 2009, 26, 191–195. [CrossRef]
5. Farash, M.S.; Chaudhry, S.A.; Heydari, M.; Sadough, S.M.S.; Kumari, S.; Khan, M.K. A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security. Int. J. Commun. Syst. 2017, 30, e3019. [CrossRef]
6. Arkko, J.; Torvinen, V.; Camarillo, G.; Niemi, A.; Haukka, T. Security Mechanism Agreement for SIP Sessions; IETF Internet Draft: Fremont, CA, USA, 2002.
7. Arshad, R.; Ikram, N. Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed. Tools Appl. 2013, 66, 165–178. [CrossRef]
8. Chaudhry, S.A.; Khan, I.; Irshad, A.; Ashraf, M.U.; Khan, M.K.; Ahmad, H.F. A provably secure anonymous authentication scheme for session initiation protocol. Secur. Commun. Netw. 2016. [CrossRef]
9. Chaudhry, S.A.; Naqvi, H.; Shon, T.; Sher, M.; Farash, M.S. Cryptanalysis and Improvement of an Improved Two Factor Authentication Protocol for Telecare Medical Information Systems. J. Med. Syst. 2015, 39, 66. [CrossRef][PubMed]
10. Farash, M.S.; Attari, M.A. An Enhanced authenticated key agreement for session initiation protocol. Inf. Technol. Control 2013, 42, 333–342. [CrossRef]
11. He, D. An efficient remote user authentication and key agreement protocol for mobile client-server environment from pairings. Ad Hoc Netw. 2012, 10, 1009–1016. [CrossRef]
12. He, D.; Chen, J.; Chen, Y. A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. Secur. Commun. Netw. 2012, 5, 1423–1429. [CrossRef]
13. Islam, S.; Khan, M. Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. J. Med. Syst. 2014, 38. [CrossRef][PubMed]
14. Kumari, S.; Karuppiah, M.; Das, A.K.; Li, X.; Wu, F.; Gupta, V. Design of a secure anonymity-preserving authentication scheme for session initiation protocol using elliptic curve cryptography. J. Ambient Intell. Hum. Comput. 2017. [CrossRef]
15. Qiu, S.; Xu, G.; Ahmad, H.; Wang, L. A Robust Mutual Authentication Scheme Based on Elliptic Curve Cryptography for Telecare Medical Information Systems. IEEE Access 2018, 6, 7452–7463. [CrossRef]
16. Shen, C.; Nahum, E.; Schulzrinne, H.; Wright, C.P. The impact of TLS on SIP server performance: Measurement and modeling. IEEE/ACM Trans. Netw. 2012, 20, 1217–1230. [CrossRef]
17. Thomas, M. SIP Security Requirements; Work In Progress; IETF Internet Draft: Fremont, CA, USA, 2001.
18. Wang, D.; He, D.; Wang, P.; Chu, C. Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. IEEE Trans. Depend. Secur. Comput. 2015, 12, 428–442. [CrossRef]
19. Xie, Q. A new authenticated key agreement for session initiation protocol. Int. J. Commun. Syst. 2012, 25, 47–54. [CrossRef]
20. Zhang, Z.; Qi, Q.; Kumar, N.; Chilamkurti, N.; Jeong, H.J. A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. Multimedia Tools Appl. 2015, 74, 3477–3488. [CrossRef]
21. Qiu, S.; Xu, G.; Ahmad, H.; Guo, Y. An enhanced password authentication scheme for session initiation protocol with perfect-forward-secrecy. PLoS ONE 2018, 13, e0194072. [CrossRef] [PubMed]
22. Qiu, S.; Xu, G.; Guo, Y.; Zhang, M. Cryptanalysis and improvement of 2 mutual authentication schemes for Session Initiation Protocol. Int. J. Commun. Syst. 2018, 31, e3568. [CrossRef]
23. Awasthi, A.K.; Srivastava, K.; Mittal, R.C. An improved timestamp-based remote user authentication scheme. Comput. Electr. Eng. 2011, 37, 869–874. [CrossRef]
24. Jau-Ji, S.; Lin, C.-W.; Hwang, M.-S. Security enhancement for the timestamp-based password authentication scheme using smart cards. Comput. Secur. 2003, 22, 591–595.
25. Huang, H.-F.; Chang, H.-W.; Yu, P.-K. Enhancement of Timestamp-based User Authentication Scheme with Smart Card. Int. J. Netw. Secur. 2014, 16, 463–467.
26. Amin, R.; Maitra, T.; Giri, D.; Srivastava, P.D. Cryptanalysis and Improvement of an RSA Based Remote User Authentication Scheme Using Smart Card. Wirel. Pers. Commun. 2017, 96, 4629–4659. [CrossRef]
27. Pippal, R.S.; Jaidhar, C.D.; Tapaswi, S. Robust Smart Card Authentication Scheme for Multi-server Architecture. Wirel. Pers. Commun. 2013, 72, 729–745. [CrossRef]
28. Li, X.; Niu, J.; Kumari, S.; Liao, J.; Liang, W. An Enhancement of a Smart Card Authentication Scheme for Multi-server Architecture. Wirel. Pers. Commun. 2015, 80, 175–192. [CrossRef]
29. Srinivas, J.; Mukhopadhyay, S.; Mishra, D. A Self-Verifiable Password Based Authentication Scheme for Multi-Server Architecture Using Smart Card. Wirel. Pers. Commun. 2017, 96, 6273–6297. [CrossRef]
30. He, D.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Inf. Sci. Int. J. 2015, 321, 263–277. [CrossRef]
31. Chang, I.; Lee, T.; Lin, T.; Liu, C. Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks. Sensors 2015, 15, 29841–29854. [CrossRef] [PubMed]
32. Hsiu-Lien, Y.; Chen, T.H.; Liu, P.C.; Tai-Hoo, K.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2011, 11, 4767–4779.
33. Choi, Y.; Lee, D.; Kim, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2014, 14, 10081–10106. [CrossRef] [PubMed]
34. Shi, W.; Gong, P. A new user authentication protocol for wireless sensor networks using elliptic curves cryptography. Int. J. Distrib. Sens. Netw. 2013, 2013, 51–59. [CrossRef]
35. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. Peer-to-Peer Netw. Appl. 2015, 8, 1070–1081. [CrossRef]
36. Jung, J.; Moon, J.; Lee, D.; Won, D. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. Sensors 2017, 17, 644. [CrossRef] [PubMed]
37. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors 2016, 16, 2123. [CrossRef] [PubMed]
38. Wang, D.; Li, W.; Wang, P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. IEEE Trans. Ind. Inform. 2018, doi:10.1109/TII.2018.2834351. [CrossRef]
39. 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]
40. Wang, D.; Wang, P. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. Ad Hoc Netw. 2014, 20, 1–15. [CrossRef]
41. Menezes, A.J. Elliptic Curve Public Key Cryptosystems; Kluwer Academic Publishers: Boston, MA, USA, 1993.
42. Wang, D.; Wang, P. Two birds with one stone: Two-factor authentication with security beyond conventional bound. IEEE Trans. Depend. Secur. Comput. 2016. [CrossRef]
43. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. Adv. Cryptol. 1999, 1666, 388–397.
44. Eisenbarth, T.; Kasper, T.; Moradi, A.; Paar, C.; Salmazadeh, M.; Shalmani, M.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; Lecture Notes in Computer Science; Springer: Berlin, Germany, 2008; Volume 5157, pp. 203–220.
45. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* 2002, 51, 541–552. [CrossRef]
46. Castiglione, A.; De Santis, A.; Castiglione, A.; Palmieri, F. An Efficient and Transparent One-Time Authentication Protocol with Non-interactive Key Scheduling and Update. *AINA* 2014, 351–358. [CrossRef]
47. Wang, D.; Wang, N.; Wang, P.; Qing, S. Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity. *Inf. Sci.* 2015, 321, 162–178. [CrossRef]
48. Wang, D.; Zhang, Z.; Wang, P. Targeted online password guessing: An underestimated threat. *Proc. ACM CCS* 2016, 16, 1242–1254.
49. Wang, D.; Wang, P. On the implications of Zipf’s law in passwords. In Proceedings of the 21st European Symposium on Research in Computer Security, Heraklion, Greece, 26–30 September 2016; pp. 11–131.
50. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf’s Law in Passwords. *IEEE Trans. Inform. Forensics Secur.* 2017, 12, 2776–2791. [CrossRef]
51. Syverson, P.F.; Cervesato, I. *The Logic of Authentication Protocols*; FOSAD: Bertinoro, Italy, 2000; pp. 63–136.
52. Burrow, M.; Abadi, M.; Needham, R.M. A logic of authentication. *ACM Trans. Comput. Syst.* 1990, 8, 18–36. [CrossRef]
53. Arshad, H.; Nikooghadam, M. An efficient and secure authentication and key agreement scheme for session initiation protocol using ECC. *Multimed. Tools Appl.* 2014, 75, 1–17. [CrossRef]
54. Kilinc, H.; Yanik, T. A survey of SIP authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* 2013. [CrossRef]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).