Research Article

A Lightweight ID Based Authentication and Key Agreement Protocol for Multiserver Architecture

Yanrong Lu,1,2 Lixiang Li,1,2 Haipeng Peng,1,2 Xing Yang,1,2 and Yixian Yang1,2

1Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China

CorrespondenceshouldbeaddressedtolixiangLi;li_xiang2006@163.com

Received 26 October 2014; Accepted 5 January 2015

Academic Editor: Javier Bajo

Copyright © 2015 Yanrong Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There is an increasing demand of an anonymous authentication to secure communications between numerous different network members while preserving privacy for the members. In this study, we address this issue by using an ID based authenticated and key agreement protocol to improve the recent protocol proposed by Xue et al. They claimed that their protocol could resist masquerade and insider attacks. Unfortunately, we find that Xue et al.’s protocol is not only really insecure against masquerade and insider attacks but also vulnerable to off-line password guessing attack. Therefore, a slight modification to their protocol is proposed to improve their shortcomings. Moreover, our protocol does not use timestamps, so it is not required to synchronize the time. As a result, according to our performance and security analyses, we can prove that our proposed protocol can enhance efficiency and improve security in comparison to previous protocols.

1. Introduction

With the rapid growth of network technology, user authentication plays an important role in achieving the dependable network environments. When we enjoy online shopping, online game, on line documentation, data exchange, and so forth, identity authentication is a basic protection measure to authenticate the identity of remote users [1]. Since Lamport [2] first protocol was shown in 1981, numerous protocols have been proposed and used in many communication systems [3–7]. In 2000, Hwang and Li [8] proposed a new remote user authentication scheme using smart card based on ElGamal’s public key cryptosystem. However, their scheme is inefficient because of high communication and computation costs. In order to remedy the security problems and to reduce the communication and computation costs, a large number of smart cards based authentication schemes using one way hash function have been investigated [9–14].

Traditionally, password authentication is mostly considered in single server environment where it has not been efficiently solved in a multiserver based environment. In addition, not only does each user need to log into different remote servers repetitively but also it needs to remember many various sets of identities and passwords if he wants to access these services. In order to solve this problem, different protocols have been suggested to access the resources of multiserver environments. In 2009, Hsiang and Shih [15] proposed a one-way hash function based remote authentication protocol for multiserver environment. Later, Sood et al. [16] showed that Hsiang et al.’s protocol could not resist stolen smart card and replay and impersonation attacks and then they proposed an improved protocol. Unfortunately, Li et al. [17] pointed out that Sood et al.’s protocol was susceptible to stolen smart card and replay attacks and was not able to provide key agreement phase. Li et al. also proposed a modified version of Sood et al.’s protocol so as to remedy the security deficiencies. Recently, Xue et al. [18] showed that Li et al.’s protocol was still vulnerable to eavesdropping and replay and forgery attacks. To remedy the weaknesses of Li et al.’s protocol, they proposed a one-way hash based authentication and key agreement protocol for multiserver architecture, which was claimed to resist many kinds of attacks. However,
through careful analysis, we found that Xue et al.'s protocol had some critical security pitfalls and is insecure for practical applications.

In this paper, we analyze a novel multiserver authentication protocol proposed by Xue et al. We show that the protocol suffers from masquerade, off-line password guessing, and insider attacks. In order to overcome these security weaknesses, a slight modification to their protocol is proposed to improve their shortcomings. Moreover, our protocol employs random numbers instead of timestamps to avoid time synchronization. As a result, according to our performance and security analysis, we can prove that our proposed protocol is able to enhance efficiency and improve security in comparison to previous protocols.

The rest of this paper is organized as follows. In Section 2, we review Xue et al.'s protocol and Section 3 shows the security weaknesses of Xue et al.'s protocol. In Section 4, we propose a new enhancement authentication protocol for multiserver environment to overcome these security weaknesses. In Section 5, we present analysis of our protocol. Section 6 shows the performance and functionality comparison among the proposed protocol and other related ones. We conclude in Section 7.

2. Review of Xue et al.'s Protocol

In Xue et al.'s protocol, there are three participants, user $U_i$, service providing server $S_j$, and control server $CS$, and four phases, namely, registration, login and authentication, password updating, and identity updating. The notations used throughout this paper are summarized as follows.

- $U_i, S_j, CS$: user, service providing server, and control server;
- $ID_i, SID_j$: identity of $U_i$ and $S_j$;
- $h(\cdot)$: hash function;
- $P_i$: password of $U_i$;
- $x, y$: secret key selected by CS;
- $\oplus, \|$: exclusive-OR operation and concatenation operation;
- $N_{1i}, N_{2i}, N_{3i}$: random number selected by $U_i$, $S_j$, and $CS$.

The login and authentication phases are shown in Box 1.

2.1. Registration Phase. CS firstly chooses two security elements $x$ and $y$.

1. $U_i$ selects a password $P_i$ and a random number $b$. Then, $U_i$ computes $A_1 = h(b \| P_i)$ and sends $\{ID_i, b, A_1\}$ to CS through a secure channel.

2. CS computes $PID_i = h(ID_i \| b)$ and $B_i = h(PID_i \| x)$ and sends $B_i$ to $U_i$ via a secure channel.

3. $U_i$ computes $C_i = h(ID_i \| A_1)$ and $D_i = B_i \oplus h(PID_i \| A_1)$ and stores $\{C_i, D_i, h(\cdot), b\}$ into the smart card.

2.2. Login and Authentication Phases. (1) $U_i$ inserts his smart card into the terminal and inputs his identity $ID_i$ and password $P_i$.

(2) $U_i$ computes $A_i = h(b \| P_i)$ and $C_i^f = h(ID_i \| A_i)$ and checks whether $C_i^f = C_i$. If it is true, $U_i$ is viewed as a legitimate user. Otherwise, the terminal rejects the login request. Then, $U_i$ computes $B_i = D_i \oplus C_i$, $F_i = B_i \oplus N_{1i}$, $P_{ij} = h(B_i \oplus h(N_{1j} \| SID_j \| PID_j \| TS_j))$, $CID_j = 1D_j \oplus h(B_i \| N_{1j} \| TS_j \| "00")$, and $G_i = b \oplus h(B_i \| N_{1j} \| TS_j \| "11")$ and transmits $\{F_i, P_i, CID_j, G_i, PID_j, TS_j\}$ to $S_j$.

(3) $S_j$ generates a random number $d$ and sends $\{SID_j, b\}$ to CS. Finally, $S_j$ obtains $BS_j = h(PSID_j \| y)$ from CS and stores $\{BS_j, b\}$ into its database. When receiving the message from $U_i$, $S_j$ checks whether $TS_j - TS_j < \Delta T$ is valid and where $TS_j$ is the current time. If it is true, $S_j$ chooses a number $N_{2j}$ and computes $L_i = BS_j \oplus N_{2j}$, $K_i = h(N_{2j} \| BS_j \| P_{ij} \| TS_j)$, and $M_i = d \oplus h(BS_j \| N_{1j} \| TS_j \| "11")$ and transmits $\{F_i, P_{ij}, CID_j, G_i, PID_j, TS_j, L_i, K_i, M_i, PSID_j\}$ to CS.

(4) CS checks whether $TS_{CS} - TS_j < \Delta T$ is valid, where $TS_{CS}$ is the current time. If it is true, $S_j$ chooses a random number $N_{12j}$, computes $BS_j = h(PSID_j \| y)$, $N_{12} = I_j \oplus BS_j$, and $K' = h(N_{12j} \| BS_j \| P_{ij} \| TS_j)$ and checks whether $K' = K$. If it is true, CS continues to calculate $B_i = h(PID_i \| x)$, $N_{1j} = F_i \oplus B_i$, $SID_j = L_i \oplus h(BS_j \| N_{12j} \| TS_j \| "00")$, and $M_i = d \oplus h(BS_j \| N_{12j} \| TS_j \| "11")$, and transmits $\{F_i, P_{ij}, CID_j, G_i, PID_j, TS_j, L_i, K_i, M_i, PSID_j\}$ to CS.

(5) $S_j$ computes $N_{1j} \oplus N_{12j} = P_i \oplus h(SID_j \| N_{12j} \| BS_j)$ and $Q_j = h(N_{12j} \| N_{13j}, R_j = N_{12j} \| h(ID_i \| N_{13j} \| B_j)$, and $V_i = h(N_{12j} \| N_{13j}$). Then, CS delivers $\{P_i, Q_i, R_i, V_i\}$ to $U_i$.

(6) $U_i$ computes $N_{12} \oplus N_{13j} = R_i \oplus h(ID_i \| N_{13j} \| B_j)$ and $V_i = h(N_{12j} \| N_{13j})$ and checks whether $V_i = V_i$. If it follows, $S_j$, $CS$ and $S_j$ are authenticated by $U_i$. Finally, the common session key SK = $h(N_{13j} \| N_{12j} \| N_{13j} \| TS_j)$ can be shared among $U_i$, $S_j$, and $CS$.

2.3. Password Updating Phase. (1) $U_i$ computes $C_i^f = h(ID_i \| A_i^f)$ and $D_i^f = B_i \oplus h(PID_i \| A_i^f)$ and updates corresponding value in the smart card.

(2) $U_i$ submits $\{ID_i, A_i^f\}$ with a new password $P_i$ to CS. Then, CS updates user's $P_i$ stored in its verification table.

2.4. Identity Updating Phase. (1) $U_i$ chooses a random number $b'$ and computes $A_i^f = h(b' \| P_i)$ and submits $\{ID_i, b', A_i^f\}$ to CS.

(2) CS computes $PID_i = h(ID_i \| b')$ and $B_i^f = h(PID_i \| x)$ and submits $B_i^f$ to $U_i$.

(3) $U_i$ computes $C_i^f = h(ID_i \| A_i^f)$ and $D_i^f = B_i' \oplus h(PID_i \| A_i^f)$. Finally, the smart card is updated to $\{C_i^f, D_i^f, h(\cdot), b'\}$.

(4) $S_j$ selects a random number $d'$ and submits $\{SID_j, d'\}$ to CS.

(5) CS computes $PSID_i^f = h(SID_j \| d')$ and $BS_i^f = h(PSID_i^f \| y)$. Then, CS sends $BS_i^f$ to $S_j$. Then, $S_j$ updates $\{BS_j, d'\}$ in its database.
calculates and off-line password guessing attacks. The detailed analyses protocol is vulnerable to two kinds of masquerade, insider many types of attacks, the actual situation is not the case. In Although Xue et al. claimed that their protocol can resist assume that a malicious attacker A can totally control his smart card, he can masquerade as G and transmits the message
\[ S_{i3}^{'}, P_{i3}^{'}, C_{ID}, G_{i}, P_{ID}, T_{Si3} \] to CS.

(4) When receiving the message from S_{i3}, CS checks whether T_{SCS} - T_{Si3} < \Delta T. If it is true, CS calculates B_{S_i} = h(PSID_{j} \parallel y), N_{i2} = J_{i} \oplus B_{S_i}, K_{i}' = h(N_{i2} \parallel B_{S_i} \parallel P_{ij} \parallel T_{Si3}'), B_{i} = h(PID_{j} \parallel x), N_{i1}' = F_{i}^{'} \oplus B_{i}, ID_{i} = C_{ID} \oplus h(B_{i} \parallel N_{i1}' \parallel T_{Si3}' \parallel "00"), SID_{i} = L_{i}' \oplus h(B_{S_i} \parallel N_{i2} \parallel T_{Si3}' \parallel "11"), and P_{ij}' = h(B_{i} \oplus h(N_{i1}' \parallel SID_{i} \parallel PID_{i} \parallel T_{Si3}')) and checks whether P_{ij}' = P_{ij}. If it is true, CS continues to compute
\[ b = G_{i}' \oplus h(B_{i} \parallel N_{i1}' \parallel T_{Si3}' \parallel "11"), d = M_{i}' \oplus h(B_{S_i} \parallel N_{i2} \parallel T_{Si3}' \parallel "11"), PID_{i} = h(ID_{i} \parallel b), \] and PSID_{i} = h(SID_{i} \parallel d) and verifies whether PID_{i} and PSID_{i} are equal with the received corresponding values. If they follow, CS selects a random number N_{i3} and computes
\[ P_{i3} = N_{i3}' \oplus h(SID_{i} \parallel N_{i2} \parallel B_{i}) \] and
\[ Q_{i} = h(N_{i1}' \oplus N_{i3}), R_{i} = N_{i2} \oplus N_{i3} \oplus h(ID_{i} \parallel N_{i1}' \parallel B_{i}), \] and
\[ V_{i} = h(N_{i2} \oplus N_{i3}). \] Then CS sends \[ P_{i3}', Q_{i}', V_{i} \] to S_{i3}. (5) After receiving the message from CS, S_{i3} computes
\[ N_{i3}' \oplus N_{i3} = P_{i3}' \oplus h(SID_{i} \parallel N_{i2} \parallel B_{i}) \] and
\[ Q_{i}' = h(N_{i1}' \oplus N_{i3}) \] and checks whether Q_{i}' = Q_{i}. Then, S_{i3} directly transmits \[ R_{i}', V_{i} \] to \( \mathcal{A} \) who is masquerading as U_{i}. (6) The masquerading user \( \mathcal{A} \) can verify the received value of
\[ V_{i}' = h(N_{i2} \oplus N_{i3}) \] by \[ N_{i2} \oplus N_{i3} = R_{i}' \oplus h(ID_{i} \parallel N_{i1}' \parallel B_{i}). \] Finally, \( \mathcal{A} \) is masquerading as U_{i}, S_{i3}, and CS agree on the common session key \[ SK = h(N_{i1}' \parallel N_{i2} \parallel N_{i3}) \parallel TS_{i3}' \] and access the services provided by S_{i3}. 3.2 Masquerade Attack against a Legitimate Service Providing Server. Assume a malicious attacker \( \mathcal{A} \) has broken S_{i}. Then,
\(\mathcal{A}\) can get the secret number \(BS_j\) and perform the following masquerade attack.

1. \(\mathcal{A}\) has intercepted a valid request message \(\{F_i, P_{ij}, CID_i, G_i, PID_i, TS_j\}\) sent from \(U_i\) to \(S_j\) in the public communication. Then, \(\mathcal{A}\) computes \(I_1 = BS_j \oplus N_i^{12}, K_i = h(N_i^{12} \parallel BS_j \parallel P_{ij} \parallel TS_j), L_i = SID_j \oplus h(BS_j \parallel N_i^{12} \parallel TS_j, \text{“00”}), \) and \(M_i = d \oplus h(BS_j \parallel N_i^{12} \parallel TS_j, \text{“11”})\), where \(N_i^{12}\) is a random number generated by \(\mathcal{A}\). Then \(\mathcal{A}\) transmits the message \(\{F_i, P_{ij}, CID_i, G_i, PID_i, TS_j, I_1, K_i, L_i, M_i, PSID_j\}\) to \(CS\).

2. Upon receiving the message, \(CS\) carries out a series of computations and verifications according to its original protocol without being detected since \(CS\) has no operation to validate the correctness of \(BS_j\). Finally, \(CS\) sends \(\{P_i, Q_i, R_i, V_i\}\) to \(\mathcal{A}\) who is masquerading as \(S_j\).

3. The masquerading sever \(\mathcal{A}\) can verify the received value of \(Q_i = h(N_i^{12} \oplus N_i^{13})\) through \(N_i^{12} \oplus N_i^{13} = P_i \oplus h(SID_j \parallel N_i^{12} \parallel BS_j, TS_j)\). Then the masquerading sever \(\mathcal{A}\) delivers the message \(\{R_i, V_i\}\) to \(U_i\).

4. When receiving the message from \(\mathcal{A}\) who is masquerading as \(S_j\), \(U_i\) computes \(N_i^{13} = R_i \oplus h(ID_1 \parallel N_i^{12} \parallel B_j)\) and \(V_i^* = h(N_i^{12} \oplus N_i^{13})\) and verifies \(V_i^*\) with the received value of \(V_i\). \(U_i\) will notify that the attacker who is masquerading as the server \(S_j\) is the service providing server. Therefore, \(\mathcal{A}\) can further establish a session key \(SK = h(N_i^{12} \parallel N_i^{13} \parallel TS_j)\) with \(U_i\) and \(CS\).

3.3. Off-Line Password Guessing Attack. A malicious attacker \(\mathcal{A}\) stealing user's smart card can gather information \(\{CID_i, D_i, h(\cdot), b\}\) from the memory of the stolen smart card [19, 21].

1. \(\mathcal{A}\) intercepts a request message \(\{F_i, P_{ij}, CID_i, G_i, PID_i, TS_j\}\) delivered from \(U_i\) to \(S_j\) in the public communication channel.

2. \(\mathcal{A}\) guesses a password \(P_i^*\) to compute \(A_i^* = h(b \parallel P_i^*)\) and checks whether \(h(PID_j \parallel A_i^*) = C_i\). If it is true, \(\mathcal{A}\) has guessed the correct password. Otherwise, \(\mathcal{A}\) repeatedly guesses a new password until he succeeds.

3. \(\mathcal{A}\) can also launch an off-line guessing attack on \(PID_j^* = h(ID_1 \parallel b)\) to obtain the identity \(ID_1\) of \(U_i\) since \(\mathcal{A}\) knows the value of \(b\) from the stolen smart card of \(U_i\).

4. \(\mathcal{A}\) possesses the valid smart card of \(U_i\) and knows the identity \(ID_1\) and the password \(P_i^*\) corresponding to \(U_i\) and hence can login to any service server.

3.4. Insider Attack. In general, the password is human memorable short strings. That is, password is not high-entropy keys [20]. Therefore, the following attack is feasible in practice.

1. In the registration phase, \(U_i\) sends \(\{ID_1, b, A_i\}\) to \(CS\), where \(A_i = h(b \parallel P_i)\). \(P_i\) is the password of \(U_i\). Then, a malicious insider attacker \(\mathcal{A}\) can guess a password \(P_i\) and therefore it is not difficult for \(\mathcal{A}\) to find out user's exact password \(P_i\) from \(A_i\) by performing an off-line password guessing attack.

2. \(\mathcal{A}\) tries to use identity-password pair \((ID_1, P_i)\) of \(U_i\), following the password authentication of Xue et al.'s protocol and can successfully login to the other servers.

4. Our Improved Protocol

In this section, we propose an enhanced and simple ID based authentication protocol to remedy the weaknesses of Xue et al.'s protocol. Our protocol has three phases; that is, registration, login, and authentication are shown in Box 2, and password update.

4.1. Registration Phase. The registration phase of \(U_i\) is as follows.

1. \(U_i\) generates a random number \(b\) and computes \(A_i = h(b \parallel P_i)\). Then, \(U_i\) submits \(\{ID_1, A_i\}\) to control sever \(CS\).

2. Upon receiving message from \(U_i\), \(CS\) first generates a random number \(d\) and computes \(B_i = h(A_i \parallel d)\) and \(PID_i = h(ID_1 \parallel h(y))\). Then, \(CS\) stores \(\{B_i, PID_i\}\) into a smart card and returns it to \(U_i\).

3. \(U_i\) computes \(C_i = h(ID_1 \parallel h(y) \parallel A_i)\) and stores the information \(\{B_i, C_i, h(y)\}\) into the smart card.

The registration phase of \(S_j\) is as follows.

1. \(S_j\) submits his identity \(SID_j\) to \(CS\).

2. When \(CS\) receives a registration request from \(S_j\), \(CS\) generates a random number \(e\) and computes \(PSID_j = h(SID_j \parallel e)\). Then, \(CS\) sends \(PSID_j\) to \(S_j\).

3. \(S_j\) stores \(PSID_j\) by computing \(BS_j = PSID_j \parallel z\), where \(z\) is the secret key of \(S_j\).

4.2. Login and Authentication Phases. \(U_i\) inserts his smart card into device and enters his identity \(ID_1\) and password \(P_i\). Then, the smart card validates the entered \(ID_1\) and \(P_i\) by checking whether \(C_i = h(ID_1 \parallel h(b \parallel P_i) \parallel h(y))\) is equal to the stored \(C_i\). If it holds, the smart card generates a random number \(N_i^{13}\) and computes \(P_j = h(PID_j \parallel A_j \parallel h(y)), F_j = h(SID_j \parallel h(y) \parallel P_j) \parallel ID_1\), and \(G_j = h(A_j \parallel PID_j \parallel SID_j \parallel h(y) \parallel h(y)) \parallel N_i^{13}\). Finally, \(U_i\) submits \(\{F_i, G_i\}\) to \(S_j\).

2. Upon receiving the message from \(U_i, S_j\) first extracts \(PSID_j\) from \(BS_j\) by using his secret key \(z\) and generates a random number \(N_i^{13}\). Then, \(S_j\) computes \(P_j = PSID_j \parallel F_i, K_j = PSID \parallel N_i^{13}, M_i = P_j \parallel N_i^{13} \parallel PSID\), and then transmits \(\{G_j, K_j, M_j\}\) to \(CS\).

3. (CS first checks whether \(P_j = h(PID_j \parallel A_j \parallel h(y)) \parallel P_j\). If it is true, \(CS\) generates a random number \(N_i^{13}\) and computes \(R_i = h(N_i^{13} \parallel A_j), V_i = h(N_i^{13} \parallel PSID_j), N_i^{13} = G_i \parallel h(A_j \parallel PID_j \parallel SID_j \parallel h(y)), N_i^{13} = P_i \parallel M_i, P_i = N_i^{13} \parallel N_i^{13} \parallel h(ID_1 \parallel N_i^{13} \parallel P_j)\), and \(Q_i = N_i^{13} \parallel N_i^{13} \parallel h(PSID_j \parallel N_i^{13} \parallel P_i)\). Finally, \(CS\) delivers \(\{R_i, V_i, P_i, Q_i\}\) to \(S_j\).

4. \(S_j\) directly verifies \(V_i^* = h(N_i^{13} \parallel PSID_j) \parallel V_i, Q_i\). If it holds, \(S_j\) calculates \(N_i^{13} = h(PSID_j \parallel N_i^{13} \parallel P_j) \parallel Q_i\), and sends \(\{R_i, P_i\}\) to \(U_i\).

5. Upon receiving the message from \(S_j, U_i\) first checks whether \(R_i = h(N_i^{13} \parallel A_j) \parallel R_i\) and then computes \(N_i^{13} = h(ID_1 \parallel N_i^{13} \parallel P_i)\). Finally, a session key \(SK = h(N_i^{13} \parallel N_i^{13} \parallel h(PSID_j))\) is established among \(U_i, S_j,\) and \(CS\).

4.3. Password Updating Phase. When \(U_i\) changes original password by simply inserting the smart card into a device and he can finish this process without any assistance from \(CS\).
Box 2: Login and authentication phases of our improved protocol.

\[ U_i \text{ generates a random } b' \text{ and a new password } P_i'; \text{ then } U_i \text{ computes } A'_i = h(b' \parallel P_i'). \text{ Then, the smart card will compute } C'_i = h(A'_i \parallel ID_i \parallel b') \text{ and replace } C_i \text{ with } C'_i. \]

5. Security Analysis of Our Improved Protocol

In this section, we first adopt Burrows-Abadi-Needham (BAN) logic [22] to prove that a session key between communicating parties can be correctly generated within authentication process. Then, we conduct a security analysis of the improved protocol to show that the improved protocol can withstand all possible security attacks. The following attacks are based on the assumptions that a malicious attacker \( \mathcal{A} \) has completely monitored the communication channel in login and authentication phases. So \( \mathcal{A} \) can eavesdrop, modify, insert, or delete any messages transmitted via public channel [2].

5.1. Verifying Authentication with BAN Logic

BAN logic has been highly successful in analyzing the security of authentication schemes [23]. We introduce some notations of BAN logic as follows:

- \( A \parallel X \): A believes a statement \( X \);
- \( A \leftarrow K \leftarrow B \): share a key \( K \) between \( A \) and \( B \);
- \#X: \( X \) is fresh;
- \( A \triangleleft X \): \( A \) sees \( X \);
- \( A \Rightarrow X \): \( A \) controls \( X \);
- \( A|\equiv X \): \( A \) said \( X \);
- \( (X,Y)_K \): \( X \) and \( Y \) are hashed with the key \( K \);
- \( (X)_K \): \( X \) is XORed with the key \( K \).

We introduce logical postulates of BAN logic that we used into our protocol as follows.

1. BAN logical postulates are as follows.

   a. Message-meaning rule: Consider \( (A|\equiv A \leftarrow K \leftarrow B, A \triangleleft (X)_K / (A|\equiv B \leftarrow X) \); if \( A \) believes that the key \( K \) is shared by \( A \) and \( B \) and sees \( X \) encrypted with \( K \), then \( A \) believes that \( B \) once said \( X \).

   b. Nonce-verification rule: Consider \( (A|\equiv X, A|\equiv Y) / (A|\equiv X, Y) \); if \( A \) believes that \( X \) could have been uttered only recently and that \( B \) once said \( X \), then \( A \) believes that \( B \) believes \( X \).

   c. The belief rule: Consider \( (A|\equiv X, A|\equiv Y) / (A|\equiv X, Y) \); if \( A \) believes \( X \) and \( Y \), then \( A \) believes \( X, Y \).

   d. Fresh conjuncturation rule: Consider \( (A|\equiv X), A|\equiv #(X)/ (A|\equiv #(X, Y)) \); if \( A \) believes freshness of \( X \), then \( B \) believes freshness of \( X, Y \).

   e. Jurisdiction rule: Consider \( (A|\equiv B \Rightarrow X, A|\equiv B|\equiv X) / (A|\equiv X) \); if \( A \) believes that \( B \) has jurisdiction over \( X \) and \( A \) trusts \( B \) on the truth of \( X \), then \( A \) believes \( X \).

   f. Introduction of the session key: Consider \( (A|\equiv SK, A|\equiv B|\equiv SK) / (A|\equiv A \leftarrow B) \); if \( A \) believes that the session key \( SK \) is fresh and \( B \) believes \( X \), which are the necessary elements for a key, then \( A \) believes that he/she shares the session key \( SK \) with \( B \).

2. Establishment of security goals:

   \( (g_1) U_i|\equiv U_i \leftarrow S_j \).
(3) Idealized protocol:

\[
\begin{align*}
U_i \overset{h(y)}{\rightarrow} (P_j)_{h(SID_i, h(y))} \land (ID_j \parallel h(b \parallel P_j) \parallel SID_j \parallel h(y))_{N_{i2}}, \\
S_j \overset{h(y)}{\rightarrow} (N_{i2})_{h(SID_j,e)} \land (P_{ij})_{N_{i2}}, \\
CS \overset{(P_{ij}, ID_j)}{\rightarrow} (N_{i2}, N_{i2}, N_{12}), \\
(PSID_j, P_{ij})_{(N_{i2}, N_{i2}, N_{12})} \land (N_{i2})_{h(SID_j,e)}, \\
\end{align*}
\]

(4) Initiative premises:

\[
\begin{align*}
(p_1) & \ U_i \equiv \#N_{i1}, \\
(p_2) & \ U_i \equiv \#N_{i2}, \\
(p_3) & \ U_i \equiv \#N_{i3}, \\
(p_4) & \ S_j \equiv \#N_{i2}; \\
(p_5) & \ S_j \equiv \#N_{i2}; \\
(p_6) & \ S_j \equiv \#N_{i2}, \\
(p_7) & \ CS \equiv \#N_{i2}, \\
(p_8) & \ CS \equiv \#N_{i2}, \\
(p_9) & \ CS \equiv \#N_{i2}, \\
(p_{10}) & \ S_j \equiv S_j \overset{h(y)}{\rightarrow} CS, \\
(p_{11}) & \ U_i \equiv ID_j, \\
(p_{12}) & \ U_i \equiv (P_{ij}, b), \\
(p_{13}) & \ S_j \equiv (SID_j, e), \\
(p_{14}) & \ CS \equiv U_i \Rightarrow (P_{ij}, b), \\
(p_{15}) & \ CS \equiv U_i \Rightarrow P_{ij}. \\
\end{align*}
\]

(5) Protocol analysis:

\[
\begin{align*}
(a_1) & \text{ By } (p_1)-(p_3) \text{ and } U_i \leftarrow (P_{ij}, ID_j)_{(N_{i2}, N_{i2}, N_{i2})}, \text{ we obtain } U_i \equiv (N_{i1}, N_{i2}, N_{i3}); \\
(a_2) & \text{ By } U_i \leftarrow (P_{ij}, ID_j)_{(N_{i2}, N_{i2}, N_{i2})} \text{ and } (a_1), \text{ we apply the message-meaning rule to derive } U_i \equiv S_j \Rightarrow P_{ij}; \\
(a_3) & \text{ By } (a_2), \text{ we apply the freshness concatenation rule and the nonce-verification rule to derive } U_i \equiv S_j \equiv P_{ij}; \\
(g_1) & \text{ By } (a_3), (p_1)-(p_3), \text{ and introduction of the session keys, we get } U_i \equiv U_i \overset{SK}{\rightarrow} S_j; \\
(g_2) & \text{ By } (g_1) \text{ and } (p_1)-(p_3), \text{ we apply the nonce-verification rule to derive } U_i \equiv U_i \overset{SK}{\rightarrow} S_j; \\
(a_4) & \text{ By } (p_{i2}) \text{ and } S_j \leftarrow \langle (P_{ij})_{h(SID_j, h(y))} \rangle, \text{ we obtain } S_j \equiv h(SID_j \parallel h(y)); \\
(a_5) & \text{ By } S_j \leftarrow \langle (P_{ij})_{h(SID_j, h(y))} \rangle \text{ and } (a_1), \text{ we apply the message-meaning rule to derive } S_j \equiv U_i \sim P_{ij}; \\
(a_6) & \text{ By } (a_5), \text{ we apply the fresh concatenation rule and the nonce-verification rule to derive } S_j \equiv U_i \overset{SK}{\rightarrow} P_{ij}; \\
(g_3) & \text{ By } (a_6), (p_4)-(p_6), \text{ and introduction of the session keys, we get } S_j \equiv U_i \equiv U_i \overset{SK}{\rightarrow} S_j; \\
(g_4) & \text{ By } (g_3) \text{ and } (p_4)-(p_6), \text{ we apply the nonce-verification rule to derive } S_j \equiv U_i \overset{SK}{\rightarrow} S_j; \\
(a_7) & \text{ By } (p_{i6}) \text{ and } CS \leftarrow (P_{ij})_{N_{i2}}, \text{ we apply the message-meaning rule, the fresh concatenation rule, and the nonce-verification rule to derive } CS \equiv U_i \overset{SK}{\rightarrow} P_{ij}; \\
(a_8) & \text{ By } (a_7), (p_{i4})-(p_{i5}), \text{ and } (a_1), \text{ we apply the jurisdiction rule and the belief rule to derive } CS \equiv P_{ij}; \\
(a_9) & \text{ By } (a_8), (p_{i6}) \text{ and } CS \leftarrow (P_{ij})_{N_{i2}}, \text{ we apply the fresh concatenation rule and the nonce-verification rule to derive } CS \equiv S_j \overset{SK}{\rightarrow} S_j; \\
(g_5) & \text{ By } (a_9), (p_{i4})-(p_{i5}), \text{ and introduction of the session keys, we get } CS \equiv S_j \overset{SK}{\rightarrow} CS; \\
(g_6) & \text{ By } (g_5) \text{ and } (p_{i4})-(p_{i5}), \text{ we apply the nonce-verification rule to derive } CS \equiv S_j \overset{SK}{\rightarrow} CS; \\
(a_{10}) & \text{ By } (a_9), (p_{i6}) \text{ and } S_j \leftarrow (PSID_j, P_{ij})_{(N_{i2}, N_{i2}, N_{12})}, \text{ we apply the message-meaning rule, the fresh concatenation rule, and the nonce-verification rule to derive } S_j \equiv CS \equiv P_{ij}; \\
(g_7) & \text{ By } (a_{10}), (p_{i4})-(p_{i5}), \text{ and introduction of the session keys, we get } S_j \equiv S_j \overset{SK}{\rightarrow} CS; \\
(g_8) & \text{ By } (g_7) \text{ and } (p_{i4})-(p_{i5}), \text{ we apply the nonce-verification rule to derive } S_j \equiv CS \equiv S_j \overset{SK}{\rightarrow} CS; \\
(a_{11}) & \text{ By } (p_1)-(p_3), (p_{i1}) \text{ and } U_i \leftarrow (ID_j, P_{ij})_{(N_{i2}, N_{i2}, N_{i2})}, \text{ we apply the belief rule to derive } U_i \equiv P_{ij}; \\
(a_{12}) & \text{ By } (a_1) \text{ and } U_i \leftarrow (ID_j, P_{ij})_{(N_{i2}, N_{i2}, N_{i2})}, \text{ we apply the message-meaning rule, the fresh concatenation rule, and the nonce-verification rule to derive } U_i \equiv CS \equiv P_{ij}; \\
(g_9) & \text{ By } (a_{12}), (p_1)-(p_3), \text{ and introduction of the session keys, we get } U_i \equiv U_i \overset{SK}{\rightarrow} CS; \\
(g_{10}) & \text{ By } (g_9) \text{ and } (p_1)-(p_3), \text{ we apply the nonce-verification rule to derive } U_i \equiv CS \equiv U_i \overset{SK}{\rightarrow} CS; \\
\end{align*}
\]
(a_{13}) by \((p_2)\) and CS \(\in (ID, h(b \parallel P_j) \parallel SID_j \parallel h(y))_N_{i_j}\), we apply the message-meaning rule, the fresh conjuncturation rule, and the nonce-verification rule to derive CS\(\equiv U_j \equiv (ID, h(b \parallel P_j) \parallel SID_j \parallel h(y))\);

(a_{14}) by \((a_{13})\) and \((p_{10})-(p_{12})\), we apply the belief rule to derive CS\(\equiv U_j \equiv P_i\);

(g_{11}) by \((a_{13})\), \((p_8)\)-\((p_8)\), and introduction of the session keys, we get CS\(\equiv U_j \equiv P_i \leftarrow SK\) CS;

(g_{12}) by \((g_{11})\) and \((p_8)-(p_8)\), we apply the nonce-verification rule to derive CS\(\equiv U_j \equiv U_i \leftarrow SK\) CS;

As a result, analyzing the security of our protocol with BAN logic, we can now be sure that the proposed protocol is truly capable of achieving the goals.

5.2. Masquerade Attack. Assume a malicious attacker \(\mathcal{A}\) has extracted [19, 21] the information \(\{B_j, C_j, h(y)\}\) stored in the smart card. Furthermore, \(\mathcal{A}\) intercepts a request message \(\{F_i, G_i\}\) and tries to masquerade the legal user to compute the session key SK. However, it is impossible for \(\mathcal{A}\) to forge a valid login request \(\{F_i, G_i\}\) because \(\mathcal{A}\) does not know the identity ID\(_j\) and the random number \(b\) of \(U_j\), which will result in \(\mathcal{A}\) incorrectly computing the value of \(A_{i_j} = h(P_i \parallel b)\).
The identity ID\(_j\) and A\(_j\) of \(U_j\) are all protected by the one-way hash function, and thus it is computationally infeasible to derive A\(_i\) and ID\(_j\) from the values B\(_j\) = h(A\(_j\) \parallel d) and C\(_j\) = h(ID\(_j\) \parallel A_{i_j} \parallel h(y)), respectively. Thus, masquerade attack as U\(_j\) is infeasible to the proposed protocol. On the other hand, suppose \(\mathcal{A}\) intercepts a message \(\{F_i, G_i\}\) and \(\{K_i, M_i, G_i\}\) and tries to masquerade as S\(_j\) to authenticate by CS, he will fail because he cannot compute the correct P\(_{ij}\). Besides, it is also impossible to forge \(K_i\) without the knowledge of secret key \(z\) of S\(_j\). Thus, masquerade attack as S\(_j\) is also infeasible to the proposed protocol.

5.3. Insider Attack with Smart Card. Our proposed protocol provides user registration using cipher code \(A_i = h(P_i \parallel b)\) over a secret channel. Even if a malicious attacker \(\mathcal{A}\) has gotten [19, 21] the information \(\{B_j, C_j, h(y)\}\) stored in the smart card, he cannot guess the parameter \(b\) which avoids the inherent risk of stolen passwords. Thus, our protocol resists insider attack.

5.4. Replay Attack. Replay attack means a malicious attacker \(\mathcal{A}\) must not obtain sensitive information by replaying previously transmitted messages [24]. If a malicious attacker \(\mathcal{A}\) wants to replay the same messages of the sender or the receiver, it is clear that user cannot succeed because \(U_j, S_j\), and CS chooses different random numbers \((N_{i1}, N_{i2}, N_{i3})\) in each new session. Besides, \(\mathcal{A}\) cannot compute the session key SK = h\((N_{i1} \oplus N_{i2} \oplus N_{i3} \parallel h(P_{ij}))\) correctly since the parameter P\(_{ij}\) is not directly exposed in public channel. Thus, \(\mathcal{A}\) has no opportunity to successfully replay used messages.

5.5. Mutual Authentication. Our protocol can provide mutual authentication among \(U_i, S_j\), and CS.

(1) CS authenticates \(S_j\) by computing the message \(P_{ij} = h(PID, A_i \parallel h(y))\) with its own memory comparing with the receiving message \(P'_{ij} = K_j \ominus PSID \ominus M_i\), where both of \(K_j\) and \(M_i\) come from \(S_j\). Furthermore, the authentication of CS to \(U_j\) is completely dependent on the authentication of CS to \(S_j\) since obtained \(P_{ij}\) of \(S_j\) is directly derived from \(U_i\).

(2) CS is authenticated by verifying the computed \(R'_i = h(N_{i1} \parallel ID_i)\) and \(V'_i = h(N_{i2} \parallel PSID_i)\) with the received \(R_i\) and \(V_i\), respectively. At the same time, the authentication of \(U_i\) to \(S_j\) is completely dependent on the authentication of \(U_i\) to CS since \(V_i = h(ID_i \parallel N_{i3})\) transmitted by \(S_j\) is headed from CS.

5.6. Off-Line Password Guessing Attack. Assume a malicious attacker \(\mathcal{A}\) has stolen the smart card and extracted [19, 21] the information \(\{B_j, C_j, h(y)\}\) from it. Moreover, \(\mathcal{A}\) has eavesdropped the request message \(\{F_i, G_i\}\). If \(\mathcal{A}\) tries to obtain the identity ID\(_j\) and password P\(_j\) correctly at the same time, \(\mathcal{A}\) first should obtain \(A_j = h(b \parallel P_j)\). It is obviously impossible to get \(A_j\) from \(B_j = h(A_j \parallel d)\) since it is protected by a one-way hash function and a random number \(d\). Thus, the proposed protocol is secure against the off-line password guessing attack.

5.7. The Session Key Perfect Forward Secrecy. Even if a malicious attacker \(\mathcal{A}\) obtains all of participants’ secret keys and previous session keys, he still cannot compromise session key SK = h\((N_{i1} \oplus N_{i2} \oplus N_{i3} \parallel h(P_{ij}))\). Since in each session a fresh session key is generated depending on \((N_{i1}, N_{i2}, N_{i3}, P_{ij})\) and the secret differs in every session. Thus, the proposed protocol can provide the session key perfect forward secrecy.

5.8. Stolen Smart Card Attack. Even though \(\mathcal{A}\) has read [19, 21] the information \(\{B_j, C_j, h(y)\}\) from the stolen smart card, \(\mathcal{A}\) cannot get real identity ID\(_j\) and the password P\(_j\) correctly at the same time since they are protected by a one-way hash function and two random numbers \((b, d)\). Thus, it is not possible to guess these two parameters correctly at the same time in polynomial time. Therefore, the proposed protocol is secure against the stolen smart card attack.

5.9. Not Requiring Clock Synchronization. In time stamps authentication protocols, the clocks of all devices must be synchronized [25]. In our protocol, we provide random numbers based authentication mechanism, instead of the timestamps that cause serious time synchronization problems.

6. Performance and Functionality Analysis

In this section, we compare our protocol with other related protocols regarding performance and security. It is crucial for smart card based schemes to provide low computation cost due to the smart card possesses the power constraints and small flash memory [26]. We take the login phase and authentication phase into consideration since these two are
Table 1: Functionality comparison.

|                        | Ours | Xue et al. [18] | Li et al. [17] | Sood et al. [16] |
|------------------------|------|-----------------|----------------|------------------|
| Provide mutual        | Yes  | Yes             | Yes            | Yes              |
| authentication         |      |                 |                |                  |
| Provide perfect forward secrecy | Yes  | Yes             | Yes            | Yes              |
| Resist insider         | Yes  | No              | No             | No               |
| attack                 |      |                 |                |                  |
| Resist masquerade      | Yes  | No              | No             | No               |
| attack                 |      |                 |                |                  |
| Resist replay          | Yes  | Yes             | No             | No               |
| attack                 |      |                 |                |                  |
| No time                | Yes  | No              | Yes            | Yes              |
| synchronization        |      |                 |                |                  |

Figure 1: Performance comparison.

The principal part of an authentication protocol. To analyze the computational complexity of the protocols, we use hashing operation as the time complexity since xor operations require very little computations. Figure 1 shows comparison regarding the performance. From this comparison, we can see that our proposed protocol has almost the least computation costs compared with other’s protocols. Hence, our proposed protocol is very useful in environments of limited computation and communication resources to access remote information systems.

Table 1 lists the functionality comparisons of our proposed protocol with Sood et al’s protocol [16], Li et al’s protocol [17], and Xue et al’s protocol [18]. We can see that the proposed protocol not only provides proper mutual authentication and perfect forward secrecy but also can prevent masquerade attack and other attacks. As a result, the proposed protocol is more secure and has many functionalities compared with these related protocols.

7. Conclusion

In this paper, we have shown that Xue et al’s protocol cannot really protect against masquerade attack, off-line password guessing, and insider attacks. In order to avoid these security weaknesses, a slight modification without using timestamps to their protocol is proposed to improve their shortcomings. Moreover, we discussed the security of the proposed protocol and showed that it conforms to all desirable security attributes. Finally, we compared the proposed protocol and existing competitive protocols regarding efficiency and security and showed that the proposed protocol is more secure and has the least computation costs. Therefore, our protocol is able to satisfy all of the essential requirements for multiserver environments. In the future, we will propose a cryptanalysis scheme [27] to prove that our authentication mechanism is secure. Moreover, we will evaluate our scheme for the energy and communication overheads using some network simulator for practical implementation. In addition, we will continue to extend our study to combine a user’s biometrics [28] and discuss the biometrics matching issue in detail.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper. This paper is supported by the National Natural Science Foundation of China (Grant nos. 61472045 and 61121061), the Beijing Higher Education Young Elite Teacher Project (Grant no. YETP0449), the Asia Foresight Program under NSFC Grant (Grant no. 61411146001), and the Beijing Natural Science Foundation (Grant no. 4142016).

References

[1] S. U. Khan, L. Lavagno, C. Pastrone, and M. A. Spirito, “Online authentication and key establishment scheme for heterogeneous sensor networks,” International Journal of Distributed Sensor Networks, vol. 2014, Article ID 718286, 11 pages, 2014.
[2] L. Lamport, “Password authentication with insecure communication,” Communications of the ACM, vol. 24, no. 11, pp. 770–772, 1981.
[3] A. K. Awasthi and S. Lal, “An enhanced remote user authentication scheme using smart cards,” Communications of the ACM, vol. 24, no. 11, pp. 770–772, 1981.
[4] J. Xu, W.-T. Zhu, and D.-G. Feng, “An improved smart card based password authentication scheme with provable security,” Computer Standards and Interfaces, vol. 31, no. 4, pp. 723–728, 2009.
[5] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, “Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards,” IEEE Transactions on Industrial Electronics, vol. 57, no. 2, pp. 793–800, 2010.
[6] D.-W. Zhao, H.-P. Peng, L.-X. Li, and Y.-Y. Yang, “A secure and effective anonymous authentication scheme for roaming service in global mobility networks,” Wireless Personal Communications, vol. 78, no. 1, pp. 247–269, 2014.
[7] M. K. Khan and S. Kumari, “An improved user authentication protocol for healthcare services via wireless medical sensor
networks,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 347169, 10 pages, 2014.

[8] M.-S. Hwang and L.-H. Li, “A new remote user authentication scheme using smart cards,” *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[9] C.-I. Fan, Y.-C. Chan, and Z.-K. Zhang, “Robust remote authentication scheme with smart cards,” *Computers & Security*, vol. 24, no. 8, pp. 619–628, 2005.

[10] M.-S. Hwang, S.-K. Chong, and T.-Y. Chen, “DoS-resistant ID-based password authentication scheme using smart cards,” *Journal of Systems and Software*, vol. 83, no. 1, pp. 163–172, 2010.

[11] S.-W. Lee, H.-S. Kim, and K.-Y. Yoo, “Efficient nonce-based remote user authentication scheme using smart cards,” *Applied Mathematics and Computation*, vol. 167, no. 1, pp. 355–361, 2005.

[12] J.-Y. Liu, A.-M. Zhou, and M.-X. Gao, “A new mutual authentication scheme based on nonce and smart cards,” *Computer Communications*, vol. 31, no. 10, pp. 2205–2209, 2008.

[13] C.-T. Li and M.-S. Hwang, “An efficient biometrics-based remote user authentication scheme using smart cards,” *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[14] R. Song, “Advanced smart card based password authentication protocol,” *Computer Standards and Interfaces*, vol. 32, no. 5-6, pp. 321–325, 2010.

[15] H.-C. Hsiang and W.-K. Shih, “Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment,” *Computer Standards and Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[16] S. K. Sood, A. K. Sarje, and K. Singh, “A secure dynamic identity based authentication protocol for multi-server architecture,” *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[17] X. Li, Y.-P. Xiong, J. Ma, and W.-D. Wang, “An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards,” *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

[18] K.-P. Xue, P.-L. Hong, and C.-S. Ma, “A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture,” *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[19] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Advances in Cryptology—CRYPTO’ 99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings*, vol. 1666 of Lecture Notes in Computer Science, pp. 388–397, Springer, Berlin, Germany, 1999.

[20] D.-J. He, M.-D. Ma, Y. Zhang, C. Chen, and J.-J. Bu, “A strong user authentication scheme with smart cards for wireless communications,” *Computer Communications*, vol. 34, no. 3, pp. 367–374, 2011.

[21] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[22] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[23] J.-L. Tsai, N.-W. Lo, and T.-C. Wu, “Novel anonymous authentication scheme using smart cards,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2004–2013, 2013.