

Research Article

Security Enhanced Anonymous Multiserver Authenticated Key Agreement Scheme Using Smart Cards and Biometrics

**Younsung Choi,¹ Junghyun Nam,² Donghoon Lee,¹ Jiye Kim,¹
Jaewook Jung,¹ and Dongho Won¹**

¹ Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggi-do 440-746, Republic of Korea

² Department of Computer Engineering, Konkuk University, 268 Chungwondaero, Chungju, Chungcheongbuk-do 380-701, Republic of Korea

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 14 March 2014; Revised 28 July 2014; Accepted 29 July 2014; Published 8 September 2014

Academic Editor: Fei Yu

Copyright © 2014 Younsung Choi 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.

An anonymous user authentication scheme allows a user, who wants to access a remote application server, to achieve mutual authentication and session key establishment with the server in an anonymous manner. To enhance the security of such authentication schemes, recent researches combined user's biometrics with a password. However, these authentication schemes are designed for single server environment. So when a user wants to access different application servers, the user has to register many times. To solve this problem, Chuang and Chen proposed an anonymous multiserver authenticated key agreement scheme using smart cards together with passwords and biometrics. Chuang and Chen claimed that their scheme not only supports multiple servers but also achieves various security requirements. However, we show that this scheme is vulnerable to a masquerade attack, a smart card attack, a user impersonation attack, and a DoS attack and does not achieve perfect forward secrecy. We also propose a security enhanced anonymous multiserver authenticated key agreement scheme which addresses all the weaknesses identified in Chuang and Chen's scheme.

1. Introduction

With the rapid growth of internet technology, a system providing various services using the network often consists of many different servers around the world. The distribution of the remote system hardware allows its users to access resources efficiently and conveniently. In multiple server environments, an authentication mechanism is required to achieve a high level of security [1]. Lamport [2] first proposed a password authentication scheme for communication through an insecure channel. However, Lamport's scheme requires the server to manage a password table and is, thus, vulnerable to stolen-verifier attacks. To resist this attack, several researchers proposed improved password-based authentication schemes using smart cards. But, these schemes are still easily broken by simple dictionary attacks due to the low entropy of passwords and because the information stored in smart cards could be extracted by physically

monitoring power consumption [3, 4]. Therefore, many other researchers have combined users' biometrics and passwords to enhance the security of their user authentication schemes for multiserver environments; see, for example, references [5–7] for earlier work in this domain. Every human being has a different biometrics, and thus, it is difficult for the adversary to compute the biometric information [8, 9].

Relatively recently, D. Yang and B. Yang [10] and Yoon and Yoo [11] independently introduced a biometric-based multiserver authentication scheme. But, these schemes still do not consider user anonymity which has been identified as a major security property for privacy protection in many applications, including location-based services, anonymous web browsing, e-voting, and mobile roaming services. Moreover, D. Yang and B. Yang's scheme requires users to perform expensive exponentiation operations, while Yoon and Yoo's scheme, as demonstrated by He [12], is vulnerable to a privileged insider attack, a masquerade attack, and a stolen smart card attack.

Recently, Chuang and Chen [13] proposed an anonymous multiserver authenticated key agreement scheme to address the weaknesses in the D. Yang and B. Yang's scheme [10] and the Yoon-Yoo scheme [11]. This scheme is based on nonces and is very efficient in that it only requires users to perform hash function evaluations. Chuang and Chen claimed that their scheme satisfies all the desired security-related properties: anonymity, absence of verification tables, mutual authentication, resistance to forgery attack, resistance to modification attacks, resistance to replay attacks, fast error detection, resistance to off-line guessing attacks, resistance to insider attacks, simple and secure password choice and modification, biometric template protection, and session key agreement. However, we found that Chuang and Chen's scheme has various security problems. According to our analysis given in this paper, Chuang and Chen's scheme is vulnerable to a masquerade attack, a smart card attack, a user impersonation attack, and a denial-of-service (DoS) attack and does not achieve perfect forward secrecy. To solve these security problems with Chuang and Chen's scheme, we propose an improved anonymous multiserver authenticated key agreement scheme using a smart card together with biometrics and passwords.

The remainder of this paper is organized as follows. Section 2 describes security and efficiency requirements for anonymous user authentication schemes in multiserver environments. Section 3 briefly reviews Chuang and Chen's authentication scheme, while Section 4 provides a detailed security analysis on the scheme. Section 5 presents our security-enhanced authentication scheme and shows how the security weaknesses of Chuang and Chen's scheme are addressed in our scheme. Section 6 analyzes our scheme in terms of both security and efficiency. Section 7 concludes the paper.

2. Requirements for Multiserver Authentication Schemes

Most conventional password authentication methods, when they are deployed in a multiple server environment, require each network user not only to log into various remote servers repetitively but also to remember many sets of identities and passwords. Such inefficiency and complexity easily lead to the exposure of users' identities and passwords and necessarily make it difficult to manage the shared secret keys among the involved participants. Moreover, those conventional authentication methods usually do not provide user anonymity. In contrast, an anonymous multiserver authentication scheme is designed to allow users to be authenticated by multiple servers via only one registration with the registration center [1]. Figure 1 shows a framework of an anonymous user authentication system in a multiserver environment.

2.1. Security Properties. Various security requirements for a multiserver authentication scheme have been suggested in the previous studies [1, 7, 10, 13–24]. The most essential security properties include the following.

- (S1) *Anonymity*: anonymity is of increasing importance and is achieved when the user's identity is not disclosed to an unauthorized party.
- (S2) *Mutual authentication*: mutual authentication means that the two parties, user and server, authenticate each other. That is, both user and server are assured of each other's identity.
- (S3) *Session key agreement*: the user and server securely agree on a session key to be used for protecting their subsequent communications.
- (S4) *Perfect forward secrecy*: perfect forward secrecy means that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future.

2.2. Attack Resistance. To achieve these security properties, a multiserver authentication scheme has to resist various kinds of attacks. The most typical attacks include the following

- (A1) *Replay attack*: an adversary intercepts data transmissions for the purpose of making use of that data in some manner. Typically, this type of attack involves copying and possibly altering the data in various ways before releasing it for delivery to the intended recipient.
- (A2) *Modification attack*: an adversary intercepts the authentication message and attempts to modify it for illegal authentication.
- (A3) *Stolen-verifier attack*: an adversary steals the password-verifier from the server and directly uses it to masquerade as a legitimate user.
- (A4) *Off-line guessing attack*: an adversary guesses a password and verifies it in an off-line environment. The information stored in the smart card is often used in such an attack.
- (A5) *Forgery attack*: a malicious yet legitimate user attempts to forge an authentication message of another legitimate user.
- (A6) *Insider attack*: an insider attack literally means an attack mounted by a malicious insider. Malicious insiders have a distinct advantage over external adversaries because they have an authorized system access and also may be familiar with the network architecture and system policies/procedures. Typically, malicious insiders want to acquire users' private information such as their password and biometrics.
- (A7) *Masquerade attack*: an adversary is authenticated by the server using a fake user ID.
- (A8) *Smart card attack*: an adversary is authenticated by the server by using only the information obtained from a user's smart card but without the password or biometrics of the user.
- (A9) *User impersonation attack*: an adversary impersonates a legitimate user using only the user's smart card but without the password or biometric of the user.

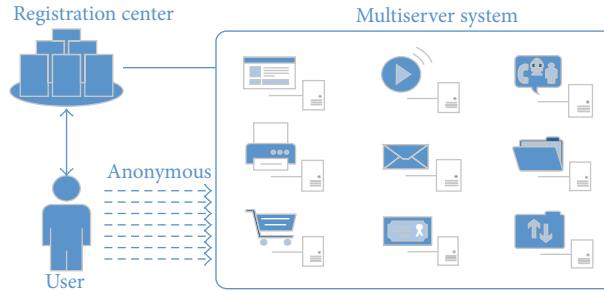


FIGURE 1: Framework of a multiserver authentication system.

(A10) *DoS Attack*. A DoS attack is any event that diminishes or eliminates a network's capability of performing its expected function. In other words, an adversary mounts a DoS attack to make the server unavailable.

2.3. *Efficiency Measures*. Efficiency is an important consideration in evaluating any schemes or protocols. The efficiency of a multiserver authentication scheme can be measured by the following metrics.

- (E1) *Single registration*: a single point of registration ought to allow users to gain access to all the servers in the system.
- (E2) *Simple and secure password modification*: the system should allow users to choose and change their passwords easily and securely. In other words, each user should be able to change their passwords without the help of any third trusted party once the authenticity of the user is verified by its smart card.
- (E3) *Fast error detection*: the smart card needs to check the user's incorrect password or any other discrepancy quickly.
- (E4) *Low computational cost*: the computational cost incurred by the scheme should be minimized for the participants.

3. A Review of Chuang and Chen's Scheme

This section describes Chuang and Chen's anonymous multiserver authenticated key agreement scheme which involves four phases: server registration, user registration, login and authentication, and password change. For convenience, the notations used throughout this paper are summarized in Notation Section.

3.1. *The Server Registration Phase*. The application server sends the RC a join message if it would like to become an authorized server. Then, the RC replies with the key (PSK) to the server through a secure channel. And then, the authorized server uses the PSK to check the user's authentication message. If the server needs to obtain the PSK from the RC to perform the authentication phase every session, authentication delay and the communication cost between the RC and the servers will increase substantially,

but this scheme and proposed scheme register only once so they are efficient.

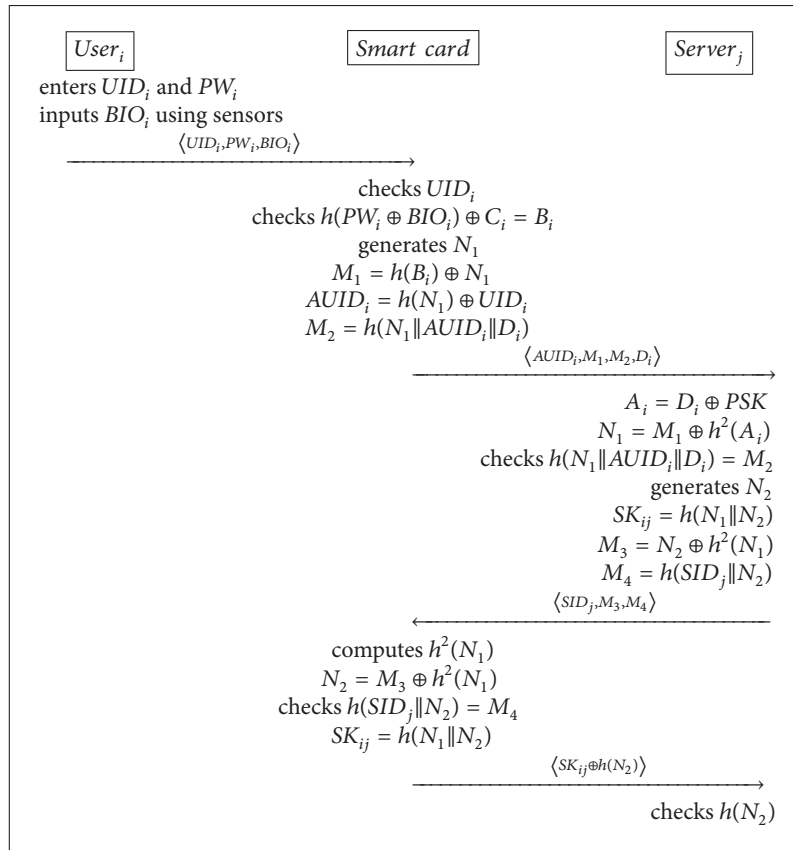
3.2. *The User Registration Phase*. For a user $user_i$, this phase is performed only once when $user_i$ registers itself with the registration center RC.

- (1) $user_i$ chooses his identity UID_i and password PW_i freely and inputs his biometrics BIO_i and sends the identity $user_i$ and $h(PW_i \oplus BIO_i)$ to RC via a secure channel.
- (2) RC computes $A_i = h(UID_i \| x)$ and $B_i = h^2(UID_i \| x) = h(A_i)$ and $C_i = h(PW_i \| BIO_i) \oplus B_i$ and $D_i = PSK \oplus A_i$ and issues $user_i$ a smart card loaded with $\langle UID_i, h(\cdot), B_i, C_i, D_i \rangle$.

3.3. *The Login and Authentication Phase*. In this phase, $user_i$ logs in to the smart card and is authenticated by server j . In login phase, is executed to check the user's legality. The smart card can detect an error event immediately using the user's identification, password, and biometrics information. And then, the smart card computes $\langle AUID_i, M_1, M_2, D_i \rangle$ for the authentication. In authentication phase, the smart card sends authentication messages to the server j after the user i finishes the login phase successfully. The smart card never send user's real identity to execute the authentication phase for providing the user's anonymity. During the phase, the session-key establishment is conducted between $user_i$ and server j . Algorithm 1 depicts how the login and authentication phase works.

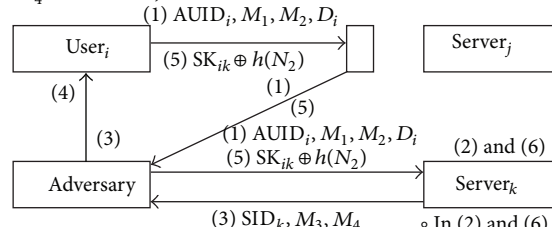
3.4. *The Password Change Phase*. One of the general guidelines to get better password security is to ensure that passwords are changed at regular intervals. Chuang and Chen's scheme allows legitimate users to freely change their passwords:

- (1) $user_i$ inserts his smart card into a card reader and enters both the current password PW_i and the new password PW_i^* .
- (2) The smart card checks UID_i and $h(PW_i \oplus BIO_i) \oplus C_i = B_i$.
- (3) The smart card computes $C_i^* = C_i \oplus h(PW_i \oplus BIO_i) \oplus h(PW_i^* \oplus BIO_i)$ and replaces C_i with C_i^* .



ALGORITHM 1: Login and authentication phase of Chuang and Chen's scheme.

- In (4), user_i does not check whether server_k wants to be authenticated with user_i or not.
- User_i only checks whether the SID in message (4) and the SID in M₄ are the same, or not.



- In (2) and (6), server_k does not check whether user_i wants to be authenticate with server_k, or not.
- Adversary can be authenticated with server_k.

FIGURE 2: Masquerade attack on Chuang and Chen's scheme.

4. Security Vulnerabilities in Chuang and Chen's Scheme

We analyze Chuang and Chen's scheme and figure out some security vulnerabilities. Their scheme is vulnerable to the masquerade attack, smart card attack, user impersonation attack, and DoS attack and does not achieve perfect forward secrecy.

4.1. A Masquerade Attack. Chuang and Chen's scheme is vulnerable to user masquerade attack. An adversary can be authenticated to another server_k using the messages that user_i sends to server_j for authentication. Figure 2 describes the masquerade attack on Chuang and Chen's scheme. When the user_i wants to be authenticate with server_j, the user_i logs on the smart card and then sends a message (1) to the server_j. After an adversary intercepts the message (1), the adversary

will send it to another server $server_k$. This is because that message (1) does not include about the server j as follows:

$$\begin{aligned} \text{Message (1)} &= \langle \text{AUID}_i, M_1, M_2, D_i \rangle, \\ \text{AUID}_i &= h(N_1) \oplus \text{UID}_i, \\ M_1 &= h(B_i) \oplus N_1, \\ M_2 &= h(N_1 \| \text{AUID}_i \| D_i), \\ D_i &= A_i \oplus \text{PSK}. \end{aligned} \quad (1)$$

So the server k executes operation (2) and sends the message (3) to the adversary without any suspicion of the attack. The adversary forwards the message (3) to the user i . The user i does not check the SID_j of the server j . It only checks the sameness with the SID of M_4 and the SID of the message (3) as follows:

$$\begin{aligned} \text{Message (3)} &= \langle \text{SID}_j, M_3, M_4 \rangle, \\ M_4 &= h(\text{SID}_j \| N_2). \end{aligned} \quad (2)$$

So the user i executes operation (4) and sends message (5) to server j without any suspicion of the attack. Then, an adversary intercepts the message (5) and sends it to another server k . Finally, the adversary can be authenticated with server k . Therefore, the adversary can masquerade as a legitimate user to server k . In this way, the scheme becomes vulnerable to the masquerade attack.

The server k cannot check whether user i wants to be authenticated by server k or not. Thus server k authenticates all legitimate messages though these message are not sent to server k . And user i does not check whether server j wants to be authenticated with user i . Thus user i authenticates all legitimate messages though these message are sent by server k . The user i only checks whether SID in message (3) and SID in M_4 are the same or not. To solve this problem, the destination of message is added to authentication messages. So the information about SID of server j has to be added to the message (1), and this means that user i want to be authenticated with server j , not server k . And the information about AUID of user i has to be added to message (3); it means that the server j wants to be authenticated with anonymous user i .

4.2. A Smart Card Attack. When an adversary gets or steals the user's smart card, the adversary can compute the session key between the user i and server j without the user's password or biometric information. So the adversary can decrypt the all encrypted communications between the user i and server j because the adversary can compute all previous session keys. Algorithm 2 describes the smart card attack on Chuang and Chen's scheme.

When the adversary obtains the user's smart card, the adversary can extract information about the smart card using a side-channel attack such as SPA (simple power analysis) or DPA (differential power analysis). The adversary can obtain B_i in the user's smart card and M_1, M_3 in the public


communication channel. Then, the adversary can compute N_1 using M_1 and $h(B_i)$ and N_2 using M_3 and $h^2(N_1)$. Finally, the adversary can determine the session key user and server using N_1 and N_2 . This scheme uses the combination values with a password and biometrics, so the adversary cannot compute the user's password. However, using the smart card attack, the adversary can compute the session key between the user i and the server j without the information about user's password or biometrics.

Kocher et al. and Messerges et al. pointed out that confidential information stored in all existent smart cards could be extracted by physically monitoring power consumption [3, 4]. If a user loses his smart card, all secrets in the smart card may be revealed to the adversary. Using this information, the adversary can determine the session key between the user i and server j . To solve this problem, it is necessary to add authentication value that adversary cannot reveal using the side-channel attack. In other words, it is necessary to add the value that only legitimate user and server can compute using the secret information, which the adversary cannot know or compute.

4.3. A User Impersonation Attack. In Chuang and Chen's scheme, an adversary can be authenticated with the server using user's smart card without user's password or biometrics, so the adversary can impersonate the legitimate user. It is critical problem that the adversary can be authenticated with the server using user's smart card only. Figure 3 describes the user impersonation attack on Chuang and Chen's scheme. As described above, the adversary can illegally extract the secret values including B_i from the user's smart card by some means. And he can intercept the message (1) = $\langle \text{AUID}_i, M_1, M_2, D_i \rangle$ and acquire the AUID_i, M_1 , and D_i .

Next procedure for user impersonation attack occurs in the following steps. The adversary computes the N_1 using M_1 and $h(B_i)$. And then, he can figure out the UID_i using AUID_i and $h(N_1)$. Next, the adversary generates another random nonce N_{A1} and computes M_{A1}, AUID_{Ai} , and M_{A2} . Next, the adversary sends $\text{AUID}_{Ai}, M_{A1}, M_{A2}$, and D_i to server j . The adversary can be authenticate to server j because he knows B_i, N_{A1} , and UID_i and the server j cannot figure out the difference between the adversary and legitimate user. The user's password and biometric information are not used in authentication phase, so server j authenticates the adversary without doubt. server j does not store user's password or biometric information because Chuang and Chen's scheme is designed for anonymous user. Therefore, server cannot check the password or biometric information for authentication. To solve this problem, it is necessary to add the shared value between the user and servers. The share value can be computed by only the legitimate user using user's password and biometrics in login and authentication phase, and never be stored in the smart card.

4.4. A DoS Attack. The DoS attack is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out motives for and targets of the DoS attack may vary, it generally consists of

(i) *Adversary* gets (steals) user's smart card.
 \Rightarrow Extracting the information of smart card.
 (Using SPA and DPA...etc) \Rightarrow Obtains B_i

(ii) *Adversary* gets M_1 and M_3 in public channel.
 $\Rightarrow N_1 = M_1 \oplus h(B_i)$
 $\Rightarrow N_2 = M_3 \oplus h^2(N_1)$
 $\Rightarrow SK_{ij} = h^2(N_1 \| N_2)$

(iii) *Adversary* can compute the session key SK_{ij} between $User_i$ and $Server_j$.

ALGORITHM 2: Smart card attack on Chuang and Chen's scheme.

(i) *Adversary* got M_{p1} and M_{p3} in previous public channel.
(ii) *Adversary* knew one of user's long-term secret: A_i
 \Rightarrow *Adversary* has A_i, M_{p1} and M_{p3}
 $\Rightarrow N_{p1} = M_{p1} \oplus h^2(A_i)$
 $\Rightarrow N_{p2} = M_{p3} \oplus h^2(N_{p1})$
 $\Rightarrow SK_{p1j} = h^2(N_{p1} \| N_{p2})$
(iii) *Adversary* can compute all of previous session key SK_{p1j} .

ALGORITHM 3: No perfect forward secrecy on Chuang and Chen's scheme.

efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the networks. In Chuang and Chen's scheme, an adversary can implement the DoS attack without difficulty. Figure 4 describes DoS attack on Chuang and Chen's scheme. The adversary gets the previous message (1) from a legitimate user and sends it to the server_j. Then, the server_j executes operation (2) and sends message (3) to the user_i. The processes of operation (2) include executing the hash function 7 times, calculating the exclusive-or operation 3 times, and generating a random nonce once. The adversary can attempt to make the server or network resource unavailable if he uses a lot of intercepted authentication messages.

In Chuang and Chen's scheme, server_j does not check the freshness of authentication message from user_i. Thus, when an adversary sends the intercepted authentication messages to server_j, the server_j cannot know whether the message is current or outdated. So, server_j executes a lot of operations. To resist the DoS attack, the server_j has to check the freshness of messages using the timestamp or other means.

4.5. No Perfect Forward Secrecy. Perfect forward secrecy means that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future. Chuang and Chen's scheme does not achieve perfect forward secrecy. So the adversary can compute the all session key between the user_i and server_j if the adversary knows the one of long-term keys A_i in future. Algorithm 3 describes why Chuang and Chen's scheme does not achieve perfect forward secrecy. First, the adversary got M_{p1} and M_{p3} in previous communication between user_i and server_j. Next, the adversary knows one of user's long-term secrets A_i . So the adversary can calculate N_{p1} from $N_{p1} =$

$M_{p1} \oplus h^2(A_i)$ and N_{p2} from $N_{p2} = M_{p3} \oplus h^2(N_{p1})$. Finally, the adversary can compute the previous session key SK_{p1j} using N_{p1} and N_{p2} . Therefore, this scheme does not achieve perfect forward secrecy.

In Chuang and Chen's scheme, A_i is a secure shared key among RC and authenticated user_i. The RC computes A_i using UID_i and secret value x . And then, The RC sends the $h(A_i)$ to user_i within user's smart card. The $h(A_i)$ is unchanged even if user_i changes his password. So A_i is one of the long-term keys. If an adversary got the M_{p1} and M_{p3} in previous public channel and knows A_i at present, the adversary can compute the previous session key between the user_i and server_j. To solve this problem, it is needed that the adversary cannot compute the N_1 and N_2 using only A_i . By adding another secret information, it is necessary that the adversary cannot compromise the session key between user_i and server_j.

5. Our Proposed Scheme

Our proposed scheme improves Chuang and Chen's scheme in various aspects: (1) it checks the destination of messages and so it prevents the masquerade attack, (2) it withstands the smart card attack and the user impersonation attack even when the information in the smart card is disclosed, (3) it resists DoS attacks by checking the freshness of messages, and (4) it protects the security of previously-established session keys even when the adversary knows the long-term key A_i , thereby achieving perfect forward secrecy.

5.1. Countermeasures. The vulnerability of Chuang and Chen's scheme to the masquerade attack is due to the fact that

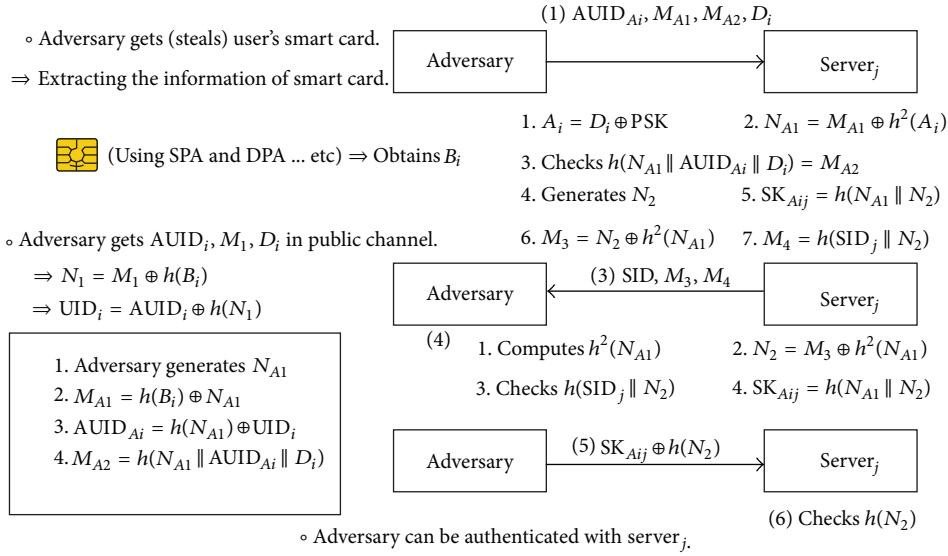


FIGURE 3: User impersonation attack on Chuang and Chen's scheme.

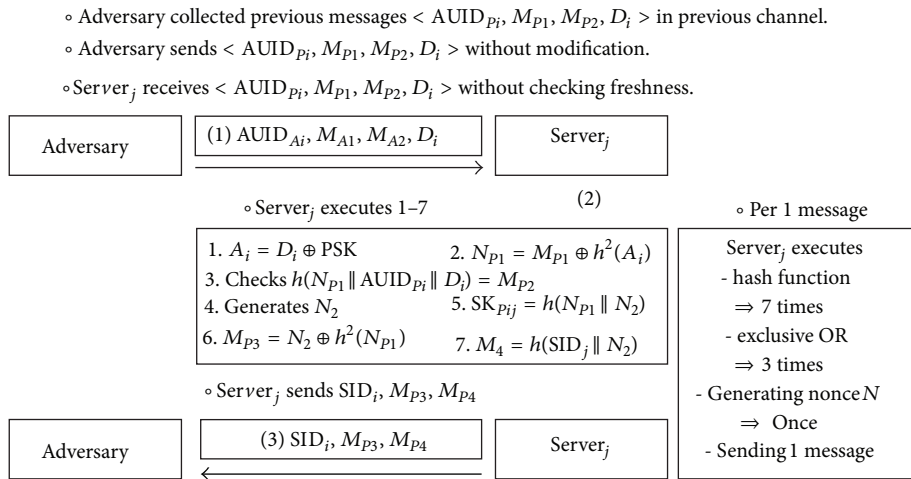


FIGURE 4: DoS attack on Chuang and Chen's scheme.

- (i) there is no way for server_j to check whether the user wants to be authenticated with it or with another server, server_k;
- (ii) user_i cannot check whether the server wants to be authenticated with him or with another user, user_j.

This design flaw allows the adversary to be authenticated with server_k using user_i's message directed to server_j. Therefore, to prevent the masquerade attack, we suggest to modify the computations of M_2 and M_4 from $M_2 = h(N_1 || AUID_i || D_i)$ and $M_4 = h(SID_j || N_2)$ to

$$M_2 = h(N_1 || AUID_i || D_i || SID_j), \quad (3)$$

$$M_4 = h(SID_j || N_2 || AUID_i).$$

The server ID, SID_j , and the anonymous user ID, $AUID_i$, are now included as part of the inputs of the hash function. The

inclusion of SID_j and $AUID_i$ allows server_j and user_i to confirm the destination of the messages M_2 and M_4 , respectively, and therefore effectively prevents the masquerade attack.

The Dos attack is possible because server_j performs all its operations without checking the freshness of incoming messages, and thus it can be prevented by modifying the computation of M_2 to

$$M_2 = h(N_1 || AUID_i || D_i || SID_j || T_i), \quad (4)$$

where T_i is the timestamp retrieved by user_i and sent to server_j. The inclusion of the timestamp T_i to the computation of M_2 enables server_j to check and confirm the freshness of the user's authentication message and prevents the DoS attack. Due to this modification, the authentication message of user_i should be also modified as follows:

$$\langle AUID_i, M_1, M_2, D_i \rangle \longrightarrow \langle AUID_i, M_1, M_2, D_i, T_i \rangle. \quad (5)$$

We next present a possible way of eliminating the vulnerability of Chuang and Chen's scheme to the smart card attack. Recall that this vulnerability is due to that the value B_i stored in the smart card together with M_1 and M_3 exchanged between user_{*i*} and server_{*j*} enables the adversary to compute N_1 and N_2 and thereby to derive the session key $SK_{ij} = h^2(N_1 \| N_2)$. Therefore, to prevent the smart card attack, we suggest to modify the computations of M_1 and M_3 from $M_1 = h(B_i) \oplus N_1$ and $M_3 = N_2 \oplus h^2(N_1)$ to

$$\begin{aligned} M_1 &= h(B_i) \oplus N_1 \oplus h(\text{PSK}), \\ M_3 &= N_2 \oplus h^2(N_1) \oplus h(\text{PSK}). \end{aligned} \quad (6)$$

With this modification, the adversary now cannot compute N_1 and N_2 without the hash value $h(\text{PSK})$. To make this countermeasure work, we add a new value $E_i = h(\text{PSK}) \oplus h(\text{PW}_i \oplus \text{BIO}_i)$ to user_{*i*}'s smart card so that only user_{*i*} can extract $h(\text{PSK})$ from its password and biometrics.

However, with the modifications described above, Chuang and Chen's scheme is still vulnerable to the user impersonation attack as the adversary can obtain $h(\text{PW}_i \oplus \text{BIO}_i)$ from B_i and $C_i = h(\text{PW}_i \oplus \text{BIO}_i) \oplus B_i$ which are stored in the smart card. To prevent the user impersonation attack, we modify the computation of C_i to

$$C_i = h(\text{PW}_i \oplus \text{BIO}_i) \oplus B_i \oplus h(\text{PSK}). \quad (7)$$

The adversary now cannot calculate $h(\text{PW}_i \oplus \text{BIO}_i)$ as it does not know $h(\text{PSK})$.

Finally, to provide the perfect forward secrecy in our proposed scheme, we modify the computation of D_i from $D_i = \text{PSK} \oplus A_i$ to

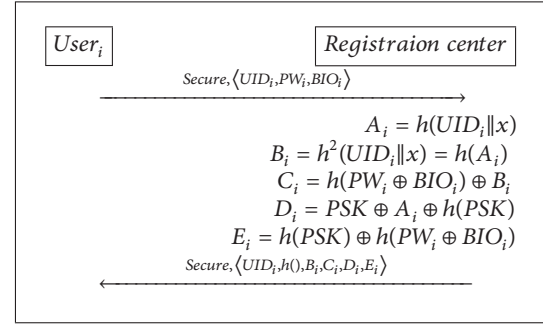
$$D_i = \text{PSK} \oplus A_i \oplus h(\text{PSK}). \quad (8)$$

With this modification, the adversary cannot derive PSK from the long-term key A_i and, thus, cannot compute N_1, N_2 , and the previous session key $SK_{ij} = h(N_1 \| N_2)$.

The password update phase should be also modified for consistency purpose (see Section 5.5 for details). Combining all the modifications above together yields an improved authentication scheme described in the following subsections.

5.2. The Server Registration Phase. The application server sends a message for join to the RC when they want to become an authorized server. Then, the RC sends the key(PSK) to the server using secure communication. And then, the server is ready to compute $h(\text{PSK})$ for user authentication. Next, the authorized server uses the shared information like PSK and $h(\text{PSK})$ to check the user's legitimacy in authentication phase.

5.3. The User Registration Phase. The registration phase of proposed scheme is described in Algorithm 4. user_{*i*} needs to perform the user registration phase with the registration center using a secure channel. In this phase, RC sends to user_{*i*} the information about PSK and $h(\text{PSK})$. PSK is included in $D_i = \text{PSK} \oplus A_i \oplus h(\text{PSK})$. user_{*i*} can be authenticated



ALGORITHM 4: Our registration phase.

with server_{*j*} using D_i but cannot compute the PSK and A_i even if he knows the D_i and $h(\text{PSK})$. And user_{*i*} can calculate the $h(\text{PSK})$ using user's password and biometrics from $E_i = h(\text{PSK}) \oplus h(\text{PW}_i \oplus \text{BIO}_i)$. In other words, the user_{*i*} receives the hidden PSK and $h(\text{PSK})$ in D_i and E_i , respectively, included in smart card for user's login and authentication. Detailed steps are explained as follows.

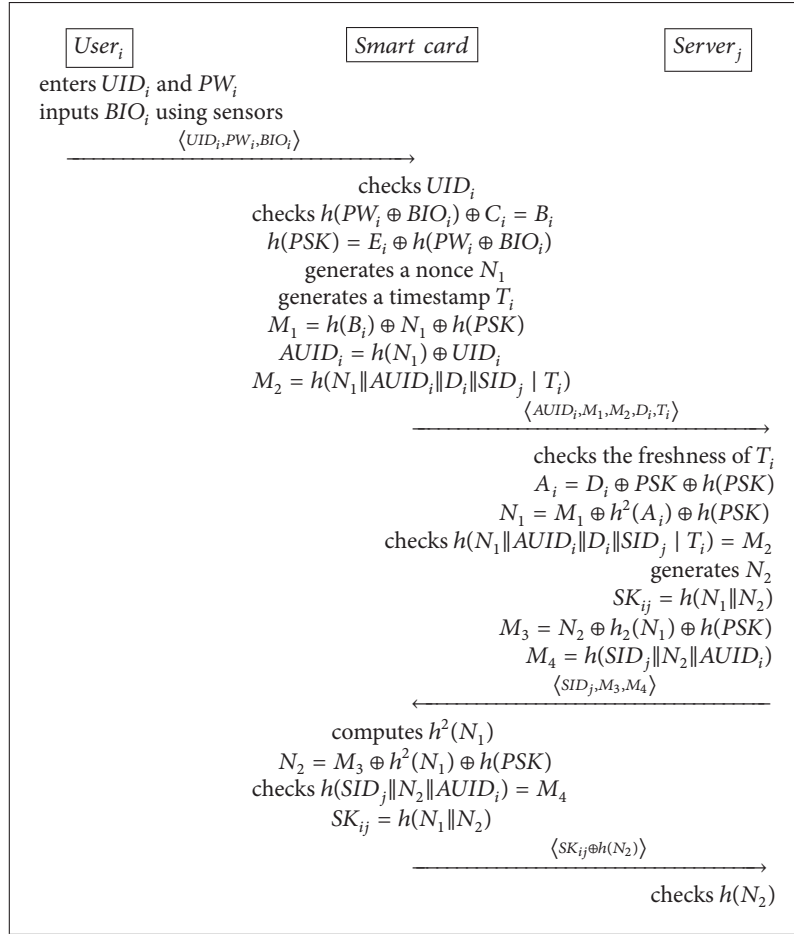
- (1) The user_{*i*} sends UID_i and $h(\text{PW}_i \oplus \text{BIO}_i)$ to the RC through a secure channel.
- (2) After receiving the user_{*i*}'s information, the RC computes the authentication parameters for the user_{*i*} as follows:

$$\begin{aligned} A_i &= h(\text{UID}_i \| x), \\ B_i &= h^2(\text{UID}_i \| x) = h(A_i), \\ C_i &= h(\text{PW}_i \oplus \text{BIO}_i) \oplus B_i, \\ D_i &= \text{PSK} \oplus A_i \oplus h(\text{PSK}), \\ E_i &= h(\text{PSK}) \oplus h(\text{PW}_i \oplus \text{BIO}_i). \end{aligned} \quad (9)$$

- (3) The RC stores these authentication parameters $\langle \text{UID}_i, h(), B_i, C_i, D_i, E_i \rangle$ in a smart card and sends the smart card to user_{*i*} via a secure channel.

The RC does not store the user's password or biometrics information. Therefore, our proposed scheme is secure against a stolen-verifier attack. The registered user cannot fake another legitimate user successfully though the user obtains these parameters $\langle \text{UID}_i, h(), B_i, C_i, D_i, E_i \rangle$. This is because that the user does not know the secret value x and PSK. The authenticated user can only compute $h(\text{PSK})$ using his password and biometrics.

5.4. The Login and Authentication Phases. The login and authentication phases for the proposed scheme are described in Algorithm 5. In the login phase, the smart card checks the legitimacy of the user. The smart card checks an error event immediately using identification, password, and biometric information. Detailed steps of the login phase are explained as follows.



ALGORITHM 5: Our login and authentication phase.

- (1) The user_i inserts his smart card into a card reader and enters his UID_i and PW_i. Then, the user_i inputs his biometric information BIO_i using the sensor.
- (2) The smart card checks the UID_i and confirms that B_i in smart card is same to h(PW_i ⊕ BIO_i) ⊕ C_i. If all information is accurate, then the smart card generates a random nonce N₁ and a timestamp T_i and computes the h(PSK) using E_i and h(PW_i ⊕ BIO_i). Next the smart card computes the following:

$$\begin{aligned}
 M_1 &= h(B_i) \oplus N_1 \oplus h(\text{PSK}), \\
 \text{AUID}_i &= h(N_1) \oplus \text{UID}_i,
 \end{aligned} \tag{10}$$

$$M_2 = h(N_1 \parallel \text{AUID}_i \parallel D_i \parallel \text{SID}_j \parallel T_i).$$

In the authentication phase, the smart card sends an authentication message to the server after the user_i finishes the login phase successfully. The proposed scheme only uses the anonymous identity AUID_i to perform the authentication phase. The detailed steps of the authentication phase are explained as follows.

- (3) The smart card sends the message ⟨AUID_i, M₁, M₂, D_i, T_i⟩ to the server_j for the user_i's authentication.

- (4) The server_j confirms the legality of the user_i and the freshness of authentication message. First, the server_j checks the freshness of T_i. If T_i is not fresh, the server_j rejects the user_i's request. The server_j uses PSK and h(PSK) to obtain A_i from the D_i. The server_j computes the value of N₁ (N₁ = M₁ ⊕ h²(A_i) ⊕ h(PSK)) and then confirms whether h(N₁ || AUID_i || D_i || SID_j || T_i) is same to M₂. If the result of M₂ is not same, the server_j terminates this session. Then, the server_j computes UID_i using h(N₁) and checks the legitimacy of UID_i. Next, the server_j generates a random nonce N₂ and computes the following:

$$\text{SK}_{ij} = h(N_1 \parallel N_2),$$

$$M_3 = N_2 \oplus h^2(N_1) \oplus h(\text{PSK}), \tag{11}$$

$$M_4 = h(\text{SID}_j \parallel N_2 \parallel \text{AUID}_i).$$

- (5) The server_j sends back the authentication message ⟨SID_j, M₃, M₄⟩ to the smart card.
- (6) The smart card confirms the legality of the server_j. It computes h²(N₁) and then calculates N₂ using M₃,

$h^2(N_1)$, and $h(\text{PSK})$. Next, the smart card checks whether

$$h(\text{SID}_j \| N_2 \| \text{AUID}_i) = M_4. \quad (12)$$

Next, the smart card computes the session key SK_{ij} as $h(N_1 \| N_2)$. Finally, the smart card computes $\text{SK}_{ij} \oplus h(N_2)$.

- (7) The smart card sends the message $\langle \text{SK}_{ij} \oplus h(N_2) \rangle$ to the server j .
- (8) The server j uses the session key SK_{ij} for checking $\text{SK}_{ij} \oplus h(N_2)$, and if $h(N_2)$ is correct, the server j authenticates the user i . From now on, the server j can communicate securely with user i using the SK_{ij} .

5.5. The Password Change Phase. The password change phase for the proposed scheme is described in Algorithm 6. The proposed password change phase is executed when the user i wants to update his password. In this phase, the user i can easily change his password without any assistance from the registration center. Detailed processes are as follows.

- (1) The user i inserts his smart card into a card reader and enters both the current password PW_i and the new password PW_i^* with UID_i and BIO_i .
- (2) The smart card checks UID_i and computes $h(\text{PSK}) = E_i \oplus h(\text{PW}_i \oplus \text{BIO}_i)$ and then checks whether

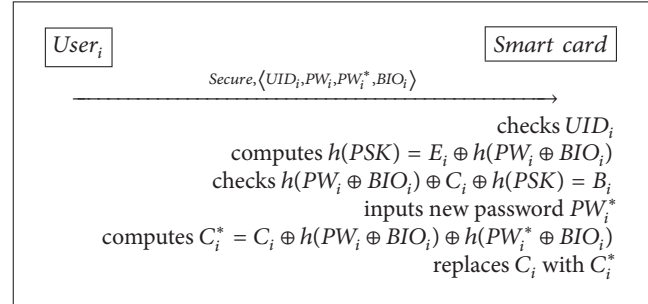
$$h(\text{PW}_i \oplus \text{BIO}_i) \oplus C_i \oplus h(\text{PSK}) = B_i. \quad (13)$$
- (3) The smart card computes $C_i^* = C_i \oplus h(\text{PW}_i \oplus \text{BIO}_i) \oplus h(\text{PW}_i^* \oplus \text{BIO}_i)$ and then replaces C_i with C_i^* .

6. Analysis of Our Scheme

An anonymous multiserver authenticated key agreement scheme has three important requirements: the security properties, the attack resistance, and the efficiency, so it needs to analyze the proposed scheme using them. In this section, we explain how the proposed scheme is satisfied with the requirements and compare the proposed scheme with other authentication schemes.

6.1. Security Properties

- (S1) *Anonymity*: in the proposed scheme, an adversary cannot compute the user's real identity UID_i without $h(N_1)$ because the real identity of user i is always converted using $\text{AUID}_i = h(N_1) \oplus \text{UID}_i$. Only legitimate server can compute and check the user's real identity, because the server has the PSK and can compute the N_1 from $N_1 = M_1 \oplus h^2(A_i) \oplus h(\text{PSK})$ using the PSK, M_1 , and A_i . Thus, only authorized server confirms the UID of user. As a result, the adversary cannot obtain the user's real identity, but legitimate user i can anonymously be authenticated with server j .



ALGORITHM 6: Our password change phase.

- (S2) *Mutual authentication*: the mutual authentication means that two parties authenticate each other. In proposed scheme, the user and server authenticated each other using N_1 , N_2 , $h(\text{PSK})$, and D_i . In the authentication phase, the server authenticates the user if the M_2 is correct as follows:

$$M_2 = h(N_1 \| \text{AUID}_i \| D_i \| \text{SID}_j \| T_i). \quad (14)$$

And the user authenticates the server using M_4 and N_2 ; it checks whether the M_4 is correct as follows:

$$M_4 = h(\text{SID}_j \| N_2 \| \text{AUID}_i). \quad (15)$$

Though an adversary intercepts the messages and wants to fake a legitimate user/server, the adversary cannot compute the accurate values, so it cannot send valid reply message to the user/server. This is because that the adversary does not know the secret key PSK, $h(\text{PSK})$ and random nonce N_1 and N_2 .

- (S3) *Session key agreement*: in the proposed scheme, the user and server can share the session key after the authentication phase. Then, they can communicate securely using the shared session key, which encrypts the communication packets. The session key is generated using $h(N_1 \| N_2)$. N_1 and N_2 change in every session, so session key is different in each session. Therefore, it is difficult for the adversary to compute the session key from the intercepted messages.
- (S4) *Perfect forward secrecy*: the proposed scheme computes the session key between the user i and server j as follows:

$$A_i = D_i \oplus \text{PSK} \oplus h(\text{PSK}),$$

$$N_1 = M_1 \oplus h^2(A_i) \oplus h(\text{PSK}),$$

$$N_2 = M_3 \oplus h^2(N_1) \oplus h(\text{PSK}),$$

$$\text{SK}_{ij} = h(N_1 \| N_2).$$

Though the user's long-term key A_i is compromised, the adversary cannot compute N_1 or N_2 because the adversary cannot calculate the $h(\text{PSK})$ and PSK,

so it cannot generate session key between user_i and server_j. Therefore, the proposed scheme achieves perfect forward secrecy. Table 1 shows the analysis on the security properties of various multisever authenticated key agreement schemes.

6.2. Attack Resistance

(A1) *Replay attack resistance*: the proposed scheme is secure against replay attack by adding the random nonce N_1 and the timestamp T_i into the message. Though an adversary intercepts the previous authentication message $\langle \text{AUID}_i, M_1, M_2, D_i, T_i \rangle$ and sends it to the server, the server can check the illegality of the request using checking N_1 and T_i as follows:

$$\text{checks } M_2 = h(N_1 \parallel \text{AUID}_i \parallel D_i \parallel \text{SID}_j \parallel T_i). \quad (17)$$

So the proposed scheme can prevent the replay attack using N_1 and T_i because the adversary cannot compute another M_2 in T_i

(A2) *Modification attack resistance*: the adversary can intercept the authentication message and attempt to modify it for illegal authentication. Using a one-way hash function, the proposed scheme checks whether authentication information is modified or not. The adversary cannot obtain the random nonce N_i or $h(\text{PSK})$, so the adversary cannot compute a legitimate authentication message. Therefore, the server and user can check whether the authentication message is modified by the adversary or not. Therefore, the proposed scheme is secure against modification attack.

(A3) *Stolen-verifier attack resistance*: the registration center and application servers do not have the user's ID/password table or the biometrics. The application server server_j authenticates the legitimate user using $h(\text{PSK})$ and D_i . Therefore, the adversary cannot obtain the authentication information about legitimate users even if the adversary gets the authority to access the database of the RC or application servers. Thus, proposed scheme is secure against stolen-verifier attack.

(A4) *Off-line guessing attack resistance*: an adversary can extract the information stored in smart card using a side-channel attack such as SPA or DPA. So the adversary can know $\text{UID}_i, B_i, C_i, D_i,$ and E_i , but he cannot figure out a user's password because $h(\text{PSK})$, $\text{PSK}, \text{BIO}_i,$ and x are unknown to the adversary. In proposed scheme, the user's password is always used with the biometrics of the user; $h(\text{PW}_i \oplus \text{BIO}_i)$, which are protected by the one-way hash function. Therefore, the adversary cannot calculate the user's password because biometric information has high entropy. Moreover, the adversary cannot figure out the biometrics because it is impossible for any two people to have the same biometrics template. Therefore, the proposed scheme is secure on off-line guessing attack.

(A5) *Forgery attack resistance*: a legitimate user cannot attempt to forge another legitimate user. The legitimate user_i can know his parameters $\langle \text{UID}_i, B_i, C_i, D_i, E_i, \text{PW}_i$ and $\text{BIO}_i \rangle$. However the user_i cannot calculate another user's real identity because another user's anonymous identity AUID_i changes in every session and is protected using a random nonce; $\text{AUID}_i = h(N_1) \oplus \text{UID}_i$. Therefore, the proposed scheme is secure against the forgery attack.

(A6) *Insider attack resistance*: in the proposed scheme, the user_i never send plain PW_i and BIO_i to the registration center RC. The user_i sends only $h(\text{PW}_i \oplus \text{BIO}_i)$, so the RC cannot obtain the user's password or biometrics. And the RC cannot compute the PW_i using $h(\text{PW}_i \oplus \text{BIO}_i)$ because the biometric information has high entropy. Moreover, $h(\text{PW}_i \oplus \text{BIO}_i)$ is sent through a secure channel and needs not store in the database of RC. So, it is difficult for even insider adversary to figure out user's PW_i and BIO_i . Therefore, the proposed scheme is secure against the insider attack.

(A7) *Masquerade attack resistance*: the masquerade attack means that an adversary is authenticated with the legitimate server using a fake or real authentication information such as the authentication messages. In Chuang and Chen's scheme, the adversary uses the authentication message between user_i and server_j to gain unauthorized access of server_k. This problem occurred because user_i and server_j cannot check the destination of authentication message. To solve this problem, the proposed scheme uses AUID_i and SID_j including M_2 as follows:

$$M_2 = h(N_1 \parallel \text{AUID}_i \parallel D_i \parallel \text{SID}_j \parallel T_i). \quad (18)$$

AUID_i includes UID_i . So the server_j can check whether user_i wants to be authenticated with server_j or not. And also M_4 include AUID_i and SID_j as follows:

$$M_4 = h(\text{SID}_j \parallel N_2 \parallel \text{AUID}_i). \quad (19)$$

So the user_i can check whether server_j wants to be authenticated with user_i or not. The adversary cannot compute M_2 and M_4 because the adversary cannot compute N_1 and N_2 . Therefore the proposed scheme is resistant to the masquerade attack.

(A8) *Smart card attack resistance*: In the proposed scheme, the smart card stores various information such as $\langle \text{UID}_i, B_i, C_i, D_i, E_i, h(\cdot) \rangle$. An adversary can obtain all information stored in user's smart card using SPA or DPA. But the adversary cannot compute the session key between user_i and server_j using M_1 and M_3

TABLE I: Comparison of security properties.

Security properties	D. Yang and B. Yang scheme [10]	Yoon and Yoo scheme [11]	Chuang and Chen scheme [13]	Our scheme
(S1) Anonymity	×	×	○	○
(S2) Mutual authentication	○	○	○	○
(S3) Session key agreement	○	○	○	○
(S4) Perfect forward secrecy	○	○	○	○

because the adversary cannot compute $h(\text{PSK})$ using obtained information as follows:

$$\begin{aligned}
 N_1 &= M_1 \oplus h(B_i) \oplus h(\text{PSK}), \\
 N_2 &= M_3 \oplus h(N_1) \oplus h(\text{PSK}), \\
 \text{SK}_{ij} &= h(N_1 \| N_2).
 \end{aligned} \tag{20}$$

Though the adversary obtains B_i and M_1 , the adversary cannot compute N_1 because of the ignorance about $h(\text{PSK})$. Thus the adversary cannot compute N_2 and SK_{ij} . Therefore the proposed scheme is secure against smart card attack.

- (A9) *User impersonation attack resistance*: in Chuang and Chen's scheme, an adversary can impersonate the legitimate user using only user's smart card because the adversary can be authenticated to the server_j using user's smart card without user's password or biometrics. However, the proposed scheme uses $h(\text{PSK})$ for protecting D_i , N_1 , N_2 , M_1 , and M_3 . For example, even though the adversary knows M_1 and B_i in $M_1 = N_1 \oplus h(B_i) \oplus h(\text{PSK})$, the adversary cannot compute N_1 without $h(\text{PSK})$, so he cannot generate the SK_{ij} . The adversary cannot know $h(\text{PSK})$ without user's password or biometric. So the adversary cannot impersonate a legal user. Therefore the proposed scheme is secure against the user impersonation attack.
- (A10) *DoS attack resistance*: the proposed scheme checks the freshness of message using timestamp, so it is useless that an adversary sends the previous message to the server. Moreover, the proposed scheme uses $M_2 = h(N_1 \| \text{AUID}_i \| D_i \| \text{SID}_j \| T_i)$ that includes timestamp T_i . The server can check the freshness and legality of M_2 because M_2 and the timestamp do not match even though the adversary sends the previous M_2 with the current timestamp. Therefore the proposed scheme is more secure against the DoS attack than Chuang and Chen's scheme.

The proposed scheme is more secure than Chuang and Chen's scheme against the masquerade attack, smart card attack, user impersonation attack, and DoS attack, and also it achieves perfect forward secrecy. Moreover, the proposed scheme is also satisfactory with regard to the anonymity, mutual authentication, session key agreement, replay attack resistance, modification attack resistance, stolen-verifier attack resistance, off-line guessing attack resistance, forgery attack resistance, and insider attack resistance.

Table 2 shows the analysis on attack resistance of various multisever authenticated key agreement schemes.

6.3. *Efficiency*. The efficiency measures include single registration, simple and secure password modification, fast error detection, and low computational cost. In performance, the proposed scheme has similar computational with Chuang and Chen's scheme. Chuang and Chen's scheme has slightly lower computational cost than the proposed scheme, but it is vulnerable to various attacks. The proposed scheme has a little higher computational cost, but it is more secure than Chuang and Chen's scheme. In other words, the proposed scheme solves security problems using similar computational cost as compared with Chuang and Chen's scheme.

- (E1) *Single registration*: in the proposed scheme, a user can be authenticated with various servers. However, the user does not need to register with every servers. To use the server's services, the user registers only one time with the registration center. The proposed scheme provides single registration so the user can anonymously use multiserver system using one registration.
- (E2) *Simple and secure password modification*: in the proposed scheme, the user can change the user's password conveniently so that it is easy for the user to change the password anytime. And, the password change phase does not need any communication with the RC. Moreover, an adversary cannot change the password even though the adversary can obtain the smart card and the user's password. This is because that the smart card can check the incorrect biometric information using PW_i , BIO_i , C_i , and B_i . The smart card verifies whether $h(\text{PW}_i \oplus \text{BIO}_i) \oplus C_i$ is the same to B_i as follows:

$$\text{checks } B_i = h(\text{PW}_i \oplus \text{BIO}_i) \oplus C_i. \tag{21}$$

- (E3) *Fast error detection*: during the login and password change phases, the smart card detects the error or mistake immediately when the adversary inputs the wrong identification, password, and biometrics information. The smart card can check the error or mistake without the RC's assistance. Therefore the proposed scheme provides fast error detection.

In Table 3, we use the following notations: “.”: that there is no computational cost in that phase, n : the number of users, m : the number of application servers, C_h : executing time of one-way hash function, C_F : executing time of the

TABLE 2: Comparison of attack resistance.

Attack resistance	D. Yang and B. Yang scheme [10]	Yoon and Yoo scheme [11]	Chuang and Chen scheme [13]	Our scheme
(A1) Replay attack	○	×	○	○
(A2) Modification attack	○	○	○	○
(A3) Stolen-verifier attack	○	○	○	○
(A4) Off-line guessing attack	○	×	○	○
(A5) Forgery attack	○	×	○	○
(A6) Insider attack	×	×	○	○
(A7) Masquerade attack	×	×	×	○
(A8) Smart card attack	○	×	×	○
(A9) User impersonation attack	○	○	×	○
(A10) DoS attack	×	×	×	○

TABLE 3: Comparison of efficiency measures.

Efficiency measures	D. Yang and B. Yang scheme [10]	Yoon and Yoo scheme [11]	Chuang and Chen scheme [13]	Our scheme
(E1) Single registration	○	○	○	○
(E2) S/S PW modification	○	○	○	○
(E3) Fast error detection	○	○	○	○
(E4) Low computational cost				
Registration user	.	C_h	C_h	C_h
Registration server
Registration RC	$n(3C_h + C_{EXP} + C_F)$	$(n + m)C_h$	$n(2C_h)$	$n(2C_h) + C_h$
Login user	$4C_h + C_{EXP} + C_F$	$2C_h + C_{ECC}$	$4C_h$	$4C_h$
Login server
Authentication user	$C_h + C_{EXP}$	$3C_h + C_{ECC}$	$5C_h$	$5C_h$
Authentication server	$3C_h + 2C_{EXP}$	$5C_h + 2C_{ECC}$	$8C_h$	$9C_h$
Authentication RC	.	$7C_h$.	.
PW change user	$3C_h + C_F$	$2C_h$	$3C_h$	$3C_h$
PW change RC

fuzzy extractor, C_{ECC} : executing time of the elliptic curve encryption or decryption operation, and C_{EXP} : executing time of the exponential operation. C_{EXP} is higher than C_{ECC} . And C_{EXP} and C_{ECC} are considerably higher than C_h . Therefore, the comparison of computational cost on above-mentioned operations is as follows:

$$C_{EXP} > C_{ECC} > C_h. \tag{22}$$

And the hash function is generally executed quickly, so it is about 1000 times faster than asymmetric encryption. In D. Yang and B. Yang’s scheme, the exponential operation is executed. In Yoon and Yoo’s scheme, the elliptic curve encryption or decryption operation is executed. But in Chuang and Chen’s scheme and proposed scheme, they use only one-way hash function. Therefore, Chuang and Chen’s scheme and proposed scheme are faster than both D. Yang and B. Yang’s scheme and Yoon and Yoo’s scheme. And our proposed scheme adds only one C_h on RC’s operation in the registration phase and also adds only one C_h on server’s operation in authentication phase in comparison with Chuang and Chen’s scheme. C_h has a little computational cost. Therefore, our proposed scheme has similar computational cost as compared with Chuang and Chen’s scheme, but Chuang and Chen’s scheme has security vulnerabilities on

the masquerade attack, smart card attack, user impersonation attack, and DoS attack as well as no perfect forward secrecy. Our proposed scheme similarly maintains the computational performance and solves the security problems of Chuang and Chen’s scheme. Therefore, the proposed scheme is the security enhanced anonymous multiserver authenticated key agreement scheme using the smart card and biometrics.

7. Conclusion

Chuang and Chen proposed an anonymous multiserver authenticated key agreement scheme. This scheme is efficient in that it only requires users to perform hash function evaluations but has various security vulnerabilities. So, we show that this scheme is vulnerable to a masquerade attack, a smart card attack, a user impersonation attack, and a DoS attack and does not achieve perfect forward secrecy. To solve the security problems of Chuang and Chen’s scheme, we propose a security enhanced anonymous multiserver authenticated key agreement scheme using smart cards and biometrics. And also, we show how the security weaknesses of Chuang and Chen’s scheme are addressed in our scheme and lastly analyze our scheme in terms of both security and efficiency.

Notations

x :	A secret value of the registration center
RC:	The registration center
UID_i :	The identification of user $_i$
SID_j :	The identification of server $_j$
$AUID_i$:	The anonymous identification of user $_i$
PW_i :	The password of user $_i$
BIO_i :	The biometrics information of user $_i$
$h()$:	A secure one-way hash function
M_i :	i th authenticator exchanged between user $_i$ and server $_j$
N_i :	A random nonce
PSK:	A secure pre-shared key among RC and servers
\parallel :	A string concatenation operation
\oplus :	A string XOR operation
\leftrightarrow :	Communication through a public channel
\leftrightarrow Secure:	Communication through a secure channel.

Conflict of Interests

The authors do not have a direct financial relation with any institution or organization mentioned in the paper that might lead to a conflict of interests for any of them.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, and Future Planning (2014R1A1A2002775).

References

- [1] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards and Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.
- [2] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO '99*, Lecture Notes in Computer Science, pp. 388–397, Springer, Berlin, Germany, 1999.
- [4] T. S. Messerges, E. A. Dabbish, and R. . . Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.
- [5] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," *Computer Communications*, vol. 32, no. 4, pp. 611–618, 2009.
- [6] M. K. Khan and J. Zhang, "An efficient and practical fingerprint-based remote user authentication scheme with smart cards," in *Information Security Practice and Experience 2006*, pp. 260–268, Springer, Berlin, Germany, 2006.
- [7] W. C. Ku, S. T. Chang, and M. H. Chiang, "Further cryptanalysis of fingerprint-based remote user authentication scheme using smartcards," *Electronics Letters*, vol. 41, no. 5, pp. 240–241, 2005.
- [8] 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.
- [9] X. Li, J. W. Niu, J. Ma, W. D. Wang, and C. L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.
- [10] D. Yang and B. Yang, "A biometric password-based multi-server authentication scheme with smart card," in *Proceedings of the International Conference on Computer Design and Applications (ICCCA '10)*, vol. 5, pp. 554–559, Qinhuaungdao, China, June 2010.
- [11] E.-J. Yoon and K.-Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.
- [12] D. He, "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," *IACR Cryptology ePrint Archive*, vol. 365, 2011.
- [13] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.
- [14] W. J. Tsaur, "A flexible user authentication scheme for multi-server internet services," in *Networking—ICN 2001*, pp. 174–183, Springer, Berlin, Germany, 2001.
- [15] L. Li, I. Lin, and M. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.
- [16] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.
- [17] J. Kim, D. Lee, W. Jeon, Y. Lee, and D. Won, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, pp. 6443–6462, 2014.
- [18] J. Nam, J. Paik, and D. Won, "Security improvement on Wu and Zhu's protocol for password-authenticated group key exchange," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94-A, no. 2, pp. 865–868, 2011.
- [19] W. Tsaur, C. Wu, and W. Lee, "An enhanced user authentication scheme for multi-server internet services," *Applied Mathematics and Computation*, vol. 170, no. 1, pp. 258–266, 2005.
- [20] T. Wu and C. Hsu, "Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks," *Computers and Security*, vol. 23, no. 2, pp. 120–125, 2004.
- [21] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.
- [22] J. Nam, K. K. R. Choo, J. Kim, H. K. Kang, J. Paik, and D. Won, "Password-only authenticated three-party key exchange with provable security in the standard model," *The Scientific World Journal*, vol. 2014, Article ID 825072, 11 pages, 2014.

- [23] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, pp. 10081–10106, 2014.
- [24] W. Jeon, J. Kim, J. Nam, Y. Lee, and D. Won, "An enhanced secure authentication scheme with anonymity for wireless environments," *IEICE Transactions on Communications*, vol. 95, no. 7, pp. 2505–2508, 2012.