Provably Secure and Efficient Three-Factor Authenticated Key Agreement Scheme with Untraceability

Ngoc-Tu Nguyen[1,2], Hai-Duong Le[2], and Chin-Chen Chang[2]
(Corresponding author: Chin-Chen Chang)

Faculty of Natural Science and Technology, Tay Nguyen University[1]
567 Le Duan Road, Buon Ma Thuot City, DakLak, Vietnam
Department of Information Engineering and Computer Science, Feng Chia University[2]
No. 100, Wenhwa Rd., Seatwen, Taichung, 40724, Taiwan, R.O.C.
(Email: alan3c@gmail.com)
(Received Apr. 28, 2015; revised and accepted June 10 & July 14, 2015)

Abstract

Authentication and key agreement protocol is indispensable for today network applications. Many two-factor authentication and key agreement protocols using smart card and password have been proposed over the last decade. However, many of these schemes are vulnerable to password guessing attack due to low-entropy passwords. In this paper, we show how to mount an offline password guessing attack against a two-factor authentication protocol. To counter against this type of attack we propose a new scheme which employs biometric information as the third authentication factor beside smart card and password. Biometric information has many positive characteristics that can fix the shortcoming of password. The proposed scheme also provides user untraceability, which is a desirable feature for ensuring users' privacy.

Keywords: Anonymity, authentication, biometric, biohashing, untraceability

1 Introduction

In the new era of Internet of Things (IoT), literally everything could be connected to networks, from a toaster to a coffee machine. In order to access the services provided over the Internet users need to authenticate with servers, and the communication channel between a user and a server must be secure by using encryption. For this purpose, in 1981, Lamport [14] introduced a remote authentication protocol which verifies users based on exchanging hashes of their passwords. In this scheme, the hashes of users' passwords are stored in a verification table instead of the plain passwords so that the secrecy of those passwords can be ensured. However, this protocol is susceptible to verification table modification and stolen verifier attacks. An adversary may replace the hash of a password with its own so it can masquerade as a legitimate user. In order to counteract these types of attacks, many studies [3, 4, 9, 12, 21, 22] discarded the verification table from their designs and employed smart card as the second authentication factor. Thus, users need to possess both password and smart card to authenticate with a server.

In 2008, Juang et al. [13] proposed a robust and efficient password-based authenticated key agreement scheme that could conceal users' identities from eavesdroppers. This type of initiator anonymity ensures users' privacy. In the authentication phase of this scheme, a ciphertext containing both user's identity and password's hash is sent to the server. This ciphertext is the same for all the login requests originated from one user; as the result, an adversary may recognize this value and trace it back to that user based on location and usage behaviors. Therefore, Li et al. [17] introduced an authentication protocol that features initiator untraceability which has a higher level of privacy than Juang et al.'s. All the parameters sent to the server are renewed after each successful login attempt. The login messages of a user in many sessions are indistinguishable from those of other users. However, Li et al.'s has two drawbacks that were pointed out by Chang et al. [5]. First, Li et al.'s employs a verification table which is susceptible to modification and stolen verifier attacks. Second, it is vulnerable to on-line password guessing attack.

Smart card and password provide a two-factor authentication, but one weakness of the password-based authentication scheme is that passwords have low entropy and are easy to break by dictionary attack. Moreover, if an adversary has successfully compromised a password and obtained its associated smart card's data, the authentica-

tion system would be completely defeated. Thus, adding biometric information of users to authentication schemes would improve the security significantly. Recent studies [7, 8, 11, 15, 16, 18, 20, 22] showed that three-factor authentication serves better for high secure environment. The biometric information could be obtained from fingerprints, iris scans, and voiceprints. These human characteristics are believed to provide a reliable authentication factor since they have high-entropy which is hard to guess or forge. Furthermore, it is difficult to duplicate or distribute biometric information; and most of all, they cannot be lost or forgotten easily.

Even using biometric information as the third authentication factor, some protocols are still prone to many attacks and flaws. For instance, Das [7] showed that Li-Hwang's three-factor authentication scheme [16] has flaws in authentication and password changing phases as well as in hashing biometric template with a common hash function. Das then proposed an improved scheme to sort out those flaws. However, Li et al. [11] pointed out that Das's scheme is vulnerable to denial-of-service, user impersonation and replay attacks. Das also repeated the Li-Hwang's flaw in hashing biometric template. Li et al.'s scheme tried to solve all those deficiencies, but it is found susceptible to server masquerading and stolen smart card attacks [6].

In this paper, we first illustrate an offline password guessing attack on Chang et al.'s scheme [5] to show the weakness of password in the two-factor authentication. Then, we propose a three-factor authentication and key agreement scheme, which provides initiator untraceability. We handle biometric information with biohashing technique [19], which verifies biohash code by calculating the Hamming distance between two biohash samples. In order to diminish the false detection, we employ the 100-bit biometric hash proposed by Jin et al. [10]. This type of biohashing technique guarantees both zero False Acceptance Rate (FAR) and False Rejection Rate (FRR). The paper is organized as follows. First, we review the Chang et al.'s scheme and its weakness in Section 2. The proposed scheme is described in detail in Section 3. Sections 4 and 5 analyze the proposed scheme's security and performance, respectively. We conclude the paper in Section 6. Table 1 shows the notations in use.

2 Chang et al.'s Scheme

In this section, we review Chang et al.'s authentication and key agreement scheme. This scheme consists of three phases: registration phase, login phase and password changing phase.

2.1 Registration Phase

The login phase is illustrated in Figure 1. In order to log in to the server, the user and smart card perform the following steps:

Table 1: Notations

U:	The user;
ID:	The user's identity;
PW:	The password of U;
S:	The server;
s,s_1,s_2:	The server's long-term secret keys;
BIO_{Re},BIO_t:	The biometric data of U in registration and authentication phases, respectively;
ε:	A predetermined biometric verification threshold;
$H(\cdot)$:	A biohashing function;
$E_x(\cdot)/D_x(\cdot)$:	A secure symmetric cipher with secret key x;
$h(\cdot)$:	A public one-way hash function.

Step 1. The user U chooses a password PW and random number r_0. Then it sends a registration request

$$m_{reg} = \{ID, h(PW) \oplus r_0\}$$

to the server.

Step 2. The server selects a random number r and computes $V = h(ID\|r)$, $IM = E_{s1}(ID\|r) \oplus s_2$, where s_1, s_2 are long-term secret keys of the server. It then computes $V_1 = V \oplus h(PW) \oplus r_0$ and issues the smart card

$$SC = \{V_1, IM\}$$

to the user U .

Step 3. Upon receiving the smart card, U computes $V_2 = V_1 \oplus r_0$ and replaces V_1 with V_2 in the smart card's memory.

2.2 Login Phase

When the user U logs into the server S, the smart card and the server carry out the following steps as depicted in Figure 2.

Step 1. The user U inserts the smart card SC into the card reader and inputs the password PW. The smart card chooses a random number r_1 and computes $V = V_2 \oplus h(PW)$, $T_1 = h(V \oplus r_1)$. It then sends the message

$$m_1 = \{r_1, T_1, IM\}$$

to the server.

Step 2. The server decrypts IM to get ID and r. It then computes $V' = h(ID\|r)$ and verifies if $T_1 = h(V' \oplus r_1)$. If T_1 is valid, the server continues the authentication process; otherwise, it terminates the session. The server chooses a random numbers r_2 and r_{new} and computes $V_{new} = h(ID \oplus$

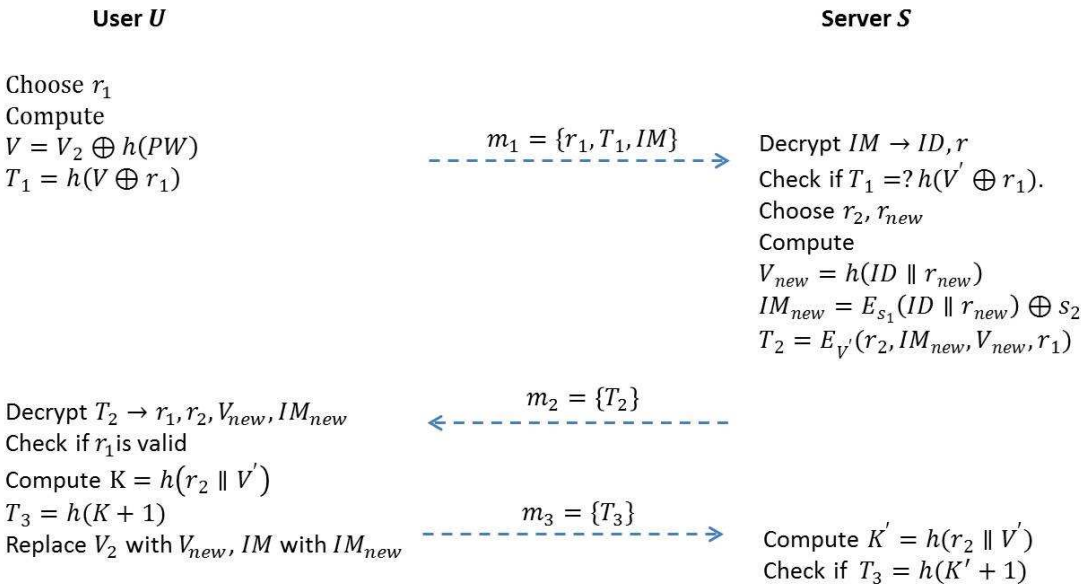

Figure 1: Chang et al.'s registration phase

r_{new}), $IM_{new} = E_{s_1}(ID \| r_{new}) \oplus s_2$, and $T_2 = E_{V'}(r_2, IM_{new}, V_{new}, r_1)$. It sends

$$m_2 = \{T_2\}$$

to SC.

Step 3. After decrypting T_2, SC obtains r_1, r_2, V_{new}, IM_{new}. If the received r_1 is valid, the smart card replaces V_2 and IM with V_{new} and IM_{new}, respectively. After that, it computes the session key $K = h(r_2 \oplus V)$ and $T_3 = h(K + 1)$, and sends

$$m_3 = T_3$$

to S.

Step 4. S computes the session key $K' = h(r_2 \oplus V')$ and verifies T_3. If $T_3 = h(K' + 1)$, the login phase has completed successfully; otherwise, it terminates the session.

2.3 Password Changing Phase

When changing the password, U inputs the new password PW_{new} and the old password PW at the terminal. SC computes $V_{2new} = V_2 \oplus h(PW) \oplus h(PW_{new})$ and replaces V_2 by V_{2new}.

2.4 Offline Password Guessing Attack against Chang et al.'s Scheme

In this attack, the adversary $\mathcal{A}$ first monitors the last login session of the user U to obtain the message $m_1 = \{r_1, T_1, IM\}$ sending from the user to the server S. It then steals U's smart card. From the smart card, the adversary obtains $V_2 = h(ID \oplus r) \oplus h(PW)$. It then performs password guessing attack. For each guessed password PW_g, A computes $V_g = V_2 \oplus h(PW_g)$. It checks if $h(V_g \oplus r_1) = T_1$. When there is a hit, the adversary has successfully guessed the user U's password. It can use this password and the smart card to access the server S.

3 The Proposed Scheme

The proposed scheme is based on biometric information and symmetric cryptosystem. It has four phases: registration phase, login and authentication phase, password changing phase and biohashing update phase.

3.1 Registration Phase

In this phase, the communication between user and server is a secure channel. This phase is depicted in Figure 3 and has the following steps.

Step 1. The user U chooses an identity ID, password pw and two random numbers b and r_0. After imprinting his/her biometric information at the sensor, U computes $PW = h(pw \oplus b)$ and $H(BIO_{Re})$. U then sends the message

$$m_{reg} = \{ID, PW \oplus H(BIO_{Re}), PW \oplus r_0\}$$

to the server for registration via secure channel.

Step 2. After verifying the identity of the user U, the server selects a random number r and computes $V_0 = h(ID \oplus r) \oplus PW \oplus r_0$. Then it computes the ciphertext $IM = E_s(ID \oplus r \oplus PW \oplus H(BIO_{Re}))$ using the

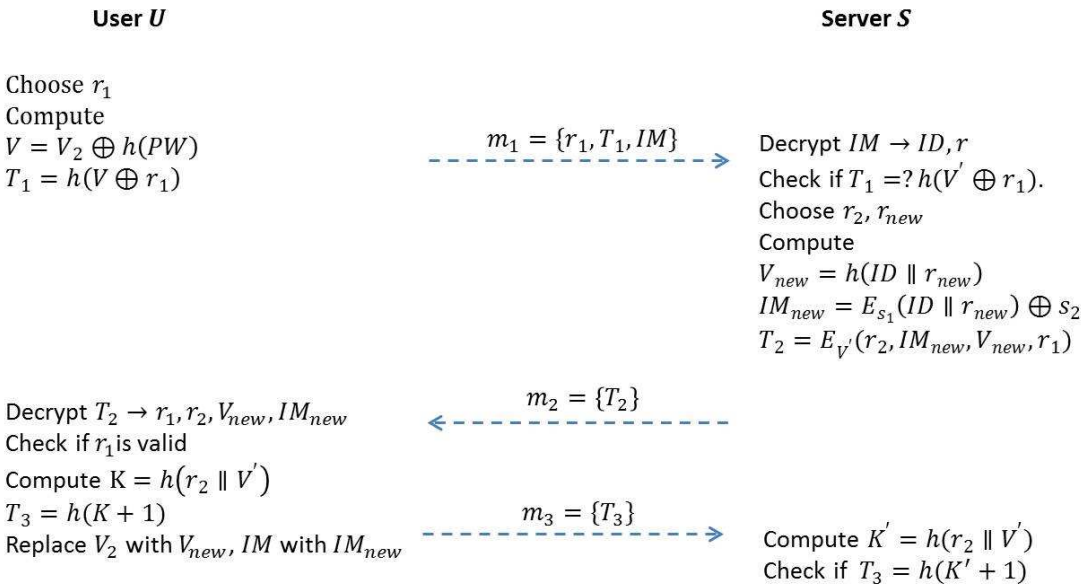

Figure 2: Chang et al.'s login phase

long-term secret key s. Finally, the smart card SC containing

$$\{V_0, IM\}$$

is issued to the user U.

Step 3. When the user U activates the smart card SC, he/she inserts the smart card to a smart card reader and inputs b, r_0. The smart card then computes $V = V_0 \oplus r_0$, replaces V_0 with V and writes b into its memory. In the end, the smart card contains

$$SC = \{b, V, IM\}.$$

3.2 Login and Authentication Phase

Figure 4 summarizes the login and authentication phase. The details of this phase are described as follows. To log into the server S, the user U first inserts the smart card into a card reader, imprints his/her biometric template BIO_t at the sensor, and inputs the password pw. The smart card interacts with the server in order to authenticate the user as follows:

Step 1. The smart card computes $PW' = h(pw \oplus b)$ and $V' = V \oplus PW = h(ID \oplus r)$. Then it chooses a number r_1 at random and computes $T_1 = h(V' \oplus r_1) \oplus PW' \oplus H(BIO_t)$. SC sends the login request

$$m_1 = \{r_1, T_1, IM\}$$

to S.

Step 2. Upon receipt of the login request m_1, S decrypts IM to obtain ID, r, and $PW \oplus H(BIO_{Re})$. Using ID and r, the server computes $PW' \oplus H(BIO_t) = T_1 \oplus h((ID \oplus r) \oplus r_1)$. It then checks whether the Hamming distance $(PW \oplus H(BIO_t), PW' \oplus H(BIO_Re)) < \varepsilon$, where ε is a predefined threshold for verifying biometric hashing. If it holds, the user is authentic; otherwise, the server terminates the session.

Step 3. After authenticating the user, the server computes the parameters for the user to use in the next session. The server first chooses a random number r_{new} and computes $V_{new} = h(ID \oplus r_{new})$, $IM_{new} = E_s(ID \oplus r_{new} \oplus PW \oplus H(BIO_{Re}))$. It then selects r_2 at random and computes the ciphertext $T_2 = E_{V'}(r_1 \oplus r_2 \oplus V_{new} \oplus IM_{new})$ using the key $V' = h(ID \oplus r)$. In the end, the server S replies to U with the message

$$m_2 = \{T_2\}.$$

Step 4. Once receiving m_2, the smart card decrypts T_2 and obtains r_1, r_2, V_{new}, IM_{new} using the key $V' = h(ID \oplus r)$. If the value r_1 in T_2 is not valid, the session is terminated; otherwise, the smart card believes that T_2 is computed by the server. It then computes $K = h(r_2 \oplus V')$ and sends confirmation message

$$m_3 = \{T_3 = h(r_2 + 1)\}$$

to server.

Step 5. The server verifies T_3. If it is not valid, S terminates the session; otherwise, it computes the session key $K = h(r_2 \oplus V')$.

Step 6. After successfully communicating with the server using the session key K, the smart card updates $V = V_{new} \oplus PW$ and $IM = IM_{new}$ in its memory.

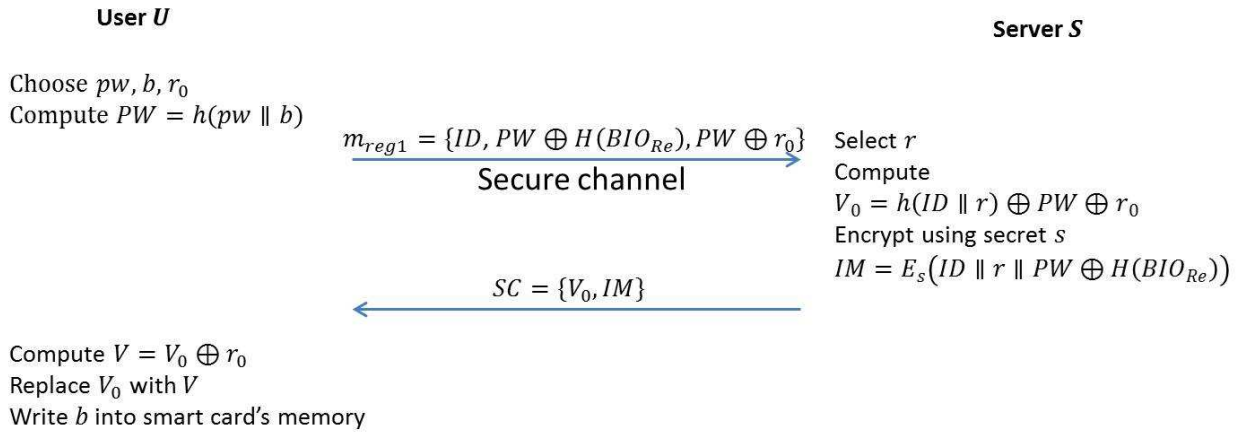

Figure 3: Registration phase

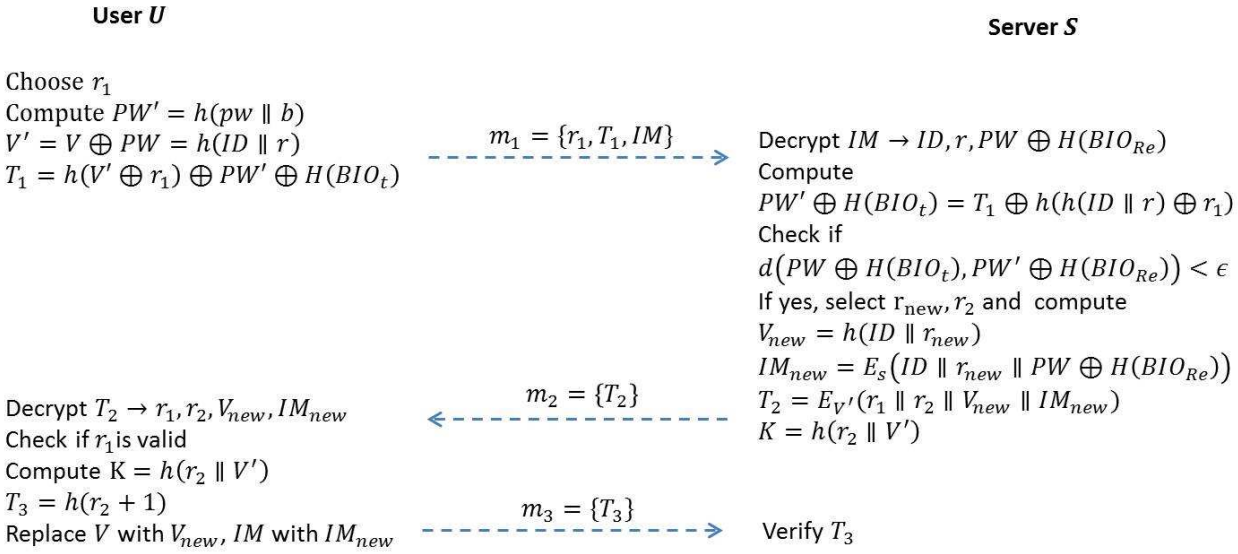

Figure 4: Login and authentication phase

3.3 Password Changing Phase

In the proposed scheme, the user can change his/her password offline. When U needs to change the password, it submits both the old and new passwords, pw and pw_{new}. The new value V is computed, $V_{new} = V \oplus h(pw \oplus b) \oplus h(pw_{new} \oplus b)$, by the smart card. Then it replaces V with V_{new} in its memory.

3.4 Biohashing Update Phase

The biohashing algorithm is based on a random vector which is generated from a hash key stored in the smart card. When user wants to update the biohash code $H(BIO_{Re})$, he/she first logs in to the server. The smart card changes the hash key value and computes a new biohash code $H(BIO_{Re,new})$. It then sends $h(pw \oplus b) \oplus H(BIO_{Re})$ to S over the established secure channel.

Upon receipt of updating biohashing request, S computes $V_{0,new} = h(ID \oplus r_{new})$, $IM_{new} = E_s(ID \oplus r_{new} \oplus h(pw \oplus b) \oplus H(BIO_{Re,new}))$, where r_{new} is chosen randomly. The server sends $V_{0,new}$ and IM_{new} to the smart card. The smart card then updates the values V_0 and IM in its memory with the received values from server.

4 Security Analysis

In this paper, we prove that our scheme is semantically secure in the real-or-random model (ROR) [1].

4.1 Security Model

Here we define the concept of security for authenticated key exchange scheme.

Participants. Let Π_S^k and $\Pi_{U_i}^j$ be the k^{th} and j^{th} instances of the server S and the user U_i, respectively.

Partnering. If Π_S^k and $\Pi_{U_i}^j$ share the same session key in the same session, the instance Π_S^k is the partner of the instance $\Pi_{U_i}^j$, and vice versa. The instance Π_S^k is the partner ID ($pid_{U_i}^j$) of the instance $\Pi_{U_i}^j$. The session ID is the transcript of all the messages exchanged between the user U_i and the server S; and it is unique. We define partnering by the session ID and the partner ID of a user U_i or the server S.

Freshness. The instance Π_S^k or $\Pi_{U_i}^j$ is *fresh* if their session key for the current session has not compromised by the adversary $\mathcal{A}$.

Adversary. In this model, the adversary $\mathcal{A}$ has total control over the data transmission between the user and the server. The adversary has the abilities to intercept, read, modify and inject messages. These capabilities are simulated using the following oracles:

- $Execute(\Pi_S^k, \Pi_{U_i}^j)$: this oracle simulates a passive eavesdropping attack. The messages exchanged between the server instance and the user instance are collected and returned to the adversary $\mathcal{A}$.
- $Send(\Pi, m)$: this models an active attack in which the adversary sends a message m to the instance Π. The oracle returns the reply message from that instance.
- $CorruptSC(\Pi_{U_i}^j)$: this models a smart card lost attack. The outputs are the information stored on the smart card.
- $CorruptPW(\Pi_{U_i}^j)$: this models the scenario where the user's password is compromised.
- $CorruptBIO(\Pi_{U_i}^j)$: this simulates the scenario where the user's biometric information is compromised.
- $Test(\Pi)$: in ROR, the scheme is secure if the advantage of the adversary in distinguishing between a random number and a real session key is negligible. When the adversary is ready, it queries $Test(\Pi)$ on an instance Π. If the random bit b, whose value was set at the start of the experiment, equals to 1 and the instance Π is fresh, the output is the real session key; otherwise, $Test(\Pi)$ outputs a random number. $Test(\Pi)$ outputs the same value, depending on b, no matter how many times A queries. If the session key was not yet established, the output is *null*.

At the end of the experiment, $\mathcal{A}$ has to output a bit b'. Let us denote $Succ$ the event where b' equals b. If the probability of $Succ$ is $Pr[Succ] \leq 1/2+\epsilon$, where ϵ is negligible, we say that the protocol is semantically secure. We define the advantage of the adversary in breaching the authenticated key agreement protocol $\mathcal{P}$ by $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = 2Pr[Succ]-1$. Thus, if $Adv_{\mathcal{P}}^{ake}(\mathcal{A})$ is negligible, the protocol $\mathcal{P}$ is semantically secure in ROR.

Ideal Cipher [2]. The cipher used in this paper is treated as ideal cipher which is a random one-to-one function for a specific key. The output of the encryption is indistinguishable from a random number.

Random Oracle. In random oracle model, the hash function is treated as a random function $h : \{0,1\}^* \rightarrow \{0,1\}^l$. The output of the hash function is indistinguishable from a random number.

4.2 Security Proof

In this section, we first assert the advantage of the adversary in Theorem 1 below. Then we prove it to show that our scheme is secure in the ROR model.

Theorem 1. *Suppose there is a polynomial time adversary $\mathcal{A}$ who wants to break the semantic security of the authenticated key agreement protocol $\mathcal{P}$ in ideal cipher and*

random oracle model, and D be a uniformly distributed password dictionary. Then

$$Adv_{\mathcal{P}}^{ake}(\mathcal{A}) \leq q_{send}/|D|,$$

where q_{send} and $|D|$ denote the total number of Send queries and the size of D, respectively.

Proof. Let G_0 refer to the game defined in real-or-random model above. Before the game starts, we choose a bit b at random. In the end, the adversary $\mathcal{A}$ ouputs a guess b' and $Succ_0$ is the event that $b' = b$. By definition, the advantage of the adversary is $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = 2Pr[Succ_0] - 1$. The game G_0 is the simulation of passive attack in which the adversary queries $Execute(\Pi_S^k, \Pi_{U_i}^j)$ and obtains the transcripts of the communications between two instances Π_S^k and $\Pi_{U_i}^j$.

Game G_1 is the same as G_0 except that we simulate the hash functions by a random oracle and the cipher by ideal cipher. Since the outputs of the random oracle and the ideal cipher are indistinguishable from random numbers, the messages m_1, m_2, and m_3 all content parameters that are indistinguishable from random numbers. Thus, $Pr[Succ_1] = Pr[Succ_0]$.

Game G_2 is the same as G_1 except that the adversary queries either $CorruptPW(\Pi_{U_i}^j)$ or both. In this game, the adversary does not have the user's smart card. Without the smart card, the adversary cannot compute neither the message $m_1 = \{r_1, T_1, IM\}$ nor the value $V' = V \oplus PW$, where V and IM are stored on the smart card. Without V', $\mathcal{A}$ cannot decrypt T_2 to obtain r_2, V_{new} and IM_{new}. Thus, the session key $K = h(r_2 \oplus V')$ is indistinguishable from a random number. Therefore, $Pr[Succ_2] = Pr[Succ_1]$.

Game G_3 is the same as G_1 except that the adversary queries $CorruptSC(\Pi_{U_i}^j)$. This game simulates the stolen smart card attack. $\mathcal{A}$ obtains $V = h(ID \oplus r) \oplus h(pw \oplus b)$ and $IM = E_s(ID \oplus r \oplus h(pw \oplus b) \oplus H(BIO_{Re}))$. Because the value r is random and *fresh* for each session, the values V and IM are indistinguishable from random values. The adversary might try to construct $T_1 = h(V' \oplus r_1) \oplus PW' \oplus H(BIO_t)$ which depends on the biometric of the user. Since BIO_t has very high entropy, $h(pw \oplus b) \oplus H(BIO_t)$ is treated as a random number in random oracle model. Therefore, T_1 is indistinguishable from a random number, and we have $Pr[Succ_3] = Pr[Succ_1]$.

Game G_4 is the same as G_3 except that the adversary also queries $CorruptPW(\Pi_{U_i}^j)$. Similar to G_3, the value T_1 is still indistinguishable from a random number since $H(BIO_t)$ is unknown. Thus, $Pr[Succ_4] = Pr[Succ_3]$.

Game G_5 is the same as G_3 except that the adversary queries $CorruptBIO(\Pi_{U_i}^j)$ in addition to $CorruptSC(\Pi_{U_i}^j)$. The adversary $\mathcal{A}$ might try to guess the password using dictionary attack. For each guessed password pw_g, $\mathcal{A}$ computes $PW_g = h(pw \oplus b)$, $V' = V \oplus PW_g$ and $T_1 = h(V' \oplus r_1) \oplus PW_g \oplus H(BIO_t)$, where r_1 is chosen randomly and BIO_t is the output of $CorruptBIO(\Pi_{U_i}^j)$. Then, the adversary queries $Send(\Pi_S^k, m_1)$. If the output of the Send query is not *null* and meaningful, pw_g is the correct password. Then, we have $|Pr[Succ_5] - Pr[Succ_3]| \leq q_{send}/|D|$.

In the last game, when all the attacks are unsuccessful, the adversary has to purely guess the value of b. Thus, $Pr[Succ_5] = 1/2$. From all the games, we have $|Pr[Succ_0] - 1/2| \leq q_{send}/|D|$.

Since $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = 2Pr[Succ_0] - 1$, we have $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) \leq q_{send}/|D|$. □

5 Comparisons of Performance

In this section, we show the computation cost of our scheme in comparison with the cost in other schemes. Then we compare our scheme security features against others. Table 2 showed the computation costs of related schemes and ours. The three biometric-based authentication schemes (Das [7], Li et al. [11], Li-Hwang [16]) utilize only hash function and XOR operations; therefore, at the first glance, they have better performance in term computation cost. Our scheme and Chang et al.'s [5], besides using hash function and XOR operations, also employ the symmetric cryptographic system; thus, they have higher computation cost. Moreover, the latter two schemes require more computation because they provide key agreement and user untraceability as shown in Table 3. Between our scheme and Chang et al.'s scheme, ours has higher computation cost since we feature three-factor authentication and key agreement; implementation of biometric template as an authentication factor results more workload. This is the trade-off between performance and security.

Table 2 shows that there is no computation cost at the user side in [7, 11, 16] because, in those scheme, they let the users to submit their identities, passwords, and biometric templates directly to a registration center. However, this leaves these schemes open to insider attack.

The computation cost is distributed quite balance between user and server in [7, 11, 16]. In our scheme, the workload at the user side is reduced significantly compared to others; thus, our scheme is more suitable for mobile systems since the mobile devices have low computation power, the workload should be put at the server side.

Table 3 shows that our scheme has more security features then others. It provides both authentication and key agreement; and it ensures user privacy while other schemes fail to protect user and server against few attacks. Most notable is that the other schemes are insecure when smart cards are stolen.

Table 2: Comparison of computation cost

	Chang et al. [5]	Li-Hwang [16]	Das [7]	Li et al. [11]	Ours
Registration					
User	$1t_h + 1t_X$	0	0	0	$2t_h + 2t_X$
Server	$1t_s + 1t_h + 2t_X$	$3t_h + 1t_X$	$3t_h + 2t_X$	$4t_h + 2t_X$	$1t_s + 1t_h + 1t_X$
Login and authentication					
User	$1t_s + 3t_h + 3t_X$	$4t_h + 3t_X$	$5t_h + 4t_X$	$2t_h + 5t_X$	$1t_s + 5t_h + 4t_X$
Server	$3t_s + 3t_h + 3t_X$	$3t_h + 2t_X$	$5t_h + 2t_X$	$3t_h + 4t_X$	$3t_s + 5t_h + 1t_X$

Table 3: Comparison of security features

	Chang et al. [5]	Li-Hwang [16]	Das [7]	Li et al. [11]	Ours
Mutual authentication	Yes	No	No	Yes	Yes
Key Agreement	Yes	No	No	No	Yes
User untraceabilitty	Yes	No	No	No	Yes
Dictionary attack	Yes	No	Yes	No	No
Replay attack	No	Yes	Yes	No	No
Impersonation attack	No	Yes	Yes	No	No
Server masquerading attack	No	Yes	Yes	Yes	No
Stolen smart card attack	Yes	Yes	Yes	Yes	No
Man-in-the-middle attack	No	No	Yes	No	No
Insider attack	No	No	Yes	No	No
Denial-of-service	No	Yes	Yes	No	No
Zero FAR and FRR errors	N/A	No	No	No	Yes

It is important to point out that our scheme adapts Jin el al.'s biohashing technique [19] which can ensure zero False Acceptance Rate (FAR) and zero False Rejection Rate (FRR). Other schemes, except Chang et al.'s, cannot guarantee the same level of accuracy in verifying biohash codes as in our scheme. They compared two biohash codes directly; thus, the False Rejection Rate would be extremely high since there are no two identical biohash codes sampled from the same entity. Therefore, our scheme is more practical compared to others.

6 Conclusions

In this paper, we first review Chang et al.'s password-based authentication and key agreement scheme with smart card. We show that the scheme is vulnerable to offline password guessing attack when user's smart card is stolen. In order to improve the scheme and protect users from this type of attack, we propose a three-factor authentication and agreement scheme that features biometric template as the third authentication factor. The proposed scheme provides a highly desirable feature, user untraceability, which protects user's privacy. This feature and good performance (due to the use of symmetric cryptography) make the scheme suitable for mobile applications. Moreover, our scheme provide practical implementation of biohashing to ensure zero False Acceptance Rate and zero False Rejection Rate in verifying biohash codes. And lastly, our scheme is proved formally to be secure in random oracle and real-or-random models; the proof would provide practitioners more confident in the scheme.

References

[1] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Public Key Cryptography (PKC'05)*, pp. 65–84, Springer, 2005.

[2] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology (Eurocrypt'00)*, pp. 139–155, Springer, 2000.

[3] C. C. Chang and S. J. Hwang, "Using smart cards to authenticate remote passwords," *Computers & Mathematics with Applications*, vol. 26, no. 7, pp. 19–27, 1993.

[4] C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," *IEE Proceedings of Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.

[5] C. C. Chang, H. D. Le, and C. H. Chang, "Novel untraceable authenticated key agreement protocol suitable for mobile communication," *Wireless Personal Communications*, vol. 71, no. 1, pp. 425–437, 2013.

[6] Y. Choi, D. Lee, J. Kim, J. Jung, and D. Won, "Cryptanalysis of improved biometric-based user authentication scheme for c/s system," *International Journal of Information and Education Technology*, vol. 5, no. 7, pp. 538, 2015.

[7] A. K Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *IET Information Security*, vol. 5, no. 3, pp. 145–151, 2011.

[8] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

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

[10] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.

[11] Li Jiping, D. Yaoming, X. Zenggang, and L. Shouyin, "An improved biometric-based user authentication scheme for c/s system," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[12] W. S. Juang, S. T. Chen, and H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2551–2556, 2008.

[13] W. S. Juang and W. K. Nien, "Efficient password authenticated key agreement using bilinear pairings," *Mathematical and Computer Modelling*, vol. 47, no. 11, pp. 1238–1245, 2008.

[14] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[15] C. C. Lee and C. W. Hsu, "A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps," *Nonlinear Dynamics*, vol. 71, no. 1-2, pp. 201–211, 2013.

[16] C. Ta 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.

[17] 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.

[18] X. Li, J. Niu, Z. Wang, and C. Chen, "Applying biometrics to design three-factor remote user authentication scheme with key agreement," *Security and Communication Networks*, vol. 7, no. 10, pp. 1488–1497, 2014.

[19] A. Lumini and L. Nanni, "An improved biohashing for human authentication," *Pattern Recognition*, vol. 40, no. 3, pp. 1057–1065, 2007.

[20] Y. Tang, "A user authentication protocol based on multiple factors," *Journal of Networks*, vol. 9, no. 10, pp. 2796–2804, 2014.

[21] J. Ho Yang and C. C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computers & Security*, vol. 28, no. 3, pp. 138–143, 2009.

[22] 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.

Ngoc-Tu Nguyen received his Bachelor of Mathematics degree in 2000 and Master of Mathematical Analysis degree in 2002 at Vinh University, Vietnam. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from Feng Chia University, Taichung, Taiwan. His current research interests include applied mathematics, information security, computer cryptography, and mobile communications.

Hai-Duong Le received his B.E. degree in 2004 at University of Tasmania, Australia, and his M.I.T degree in 2006 at James Cook University, Australia. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from Feng Chia University, Taichung, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R.O.C., AceR Dragon Award of the Ten

Most Outstanding Talents, Outstanding Scholar Award of the R.O.C., Outstanding Engineering Professor Award of the R.O.C., Distinguished Research Awards of National Science Council of the R.O.C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.