An Improved User Authentication and Key Agreement Scheme Providing User Anonymity

Ya-Fen Chang and Pei-Yu Chang

Abstract—When accessing remote services over public networks, a user authentication mechanism is required because these activities are executed in an insecure communication environment. Recently, Wang et al. proposed an authentication and key agreement scheme preserving the privacy of secret keys and providing user anonymity. Later, Chang et al. indicated that their scheme suffers from two security flaws. First, it cannot resist DoS (denial-of-service) attack because the indicators for the next session are not consistent. Second, the user password may be modified by a malicious attacker because no authentication mechanism is applied before the user password is updated. To eliminate the security flaws and preserve the advantages of Wang et al.’s scheme, we propose an improvement in this paper.

Index Terms—Authentication, key agreement, smart card, user anonymity.

1. Introduction

As communication techniques have been developed rapidly in recent years, people can easily communicate with each other over distributed computer networks at any time any place. Because remote services are accessed over public but insecure distributed computer networks, user authentication plays an important role to prevent unauthorized users from accessing system resources[1]. Among different authentication mechanisms, password authentication is widely used in plenty of applications. However, most password authentication schemes need to maintain a verification table. This approach puts a heavy burden on the remote server because the verification table size is proportional to the number of clients. Moreover, password authentication schemes may suffer from password-related attacks such as the password guessing attack, and the server needs to protect the verification table from being invaded by an attacker[2],[3].

On the other hand, smart-card-based password authentication is another handy way to help a user to access the remote server. The card holder only needs to input an easy-to-remember password and takes the smart card at his fingertips. Therefore, many smart-card-based password authentication and key agreement schemes have been proposed. In 2000, Sun proposed a password authentication scheme using a smart card with light computation loads such that the server does not need to maintain a password table[4]. Chien et al. indicated that Sun’s scheme only achieves unilateral authentication, and they proposed an enhanced smart-card-based authentication scheme providing mutual authentication[5]. Ku et al. stated that Chien et al.’s scheme is vulnerable to reflection attack and insider attack and is insecure once a user’s permanent secret stored in the smart card is compromised, and they also proposed an enhanced scheme[6]. Later, Wang et al. proposed an improvement on Ku et al.’s scheme by preserving merits and adding some security properties[7].

Recently, Wang et al. found that Wang et al.’s improved scheme suffers from known-key attack and smart card loss problem[8]. Smart card loss problem means that an attacker could get the secret value stored in a lost smart card. Several researches have reported that secret values stored in smart card may be extracted by monitoring the power consumption and analyzing the leak information in the smart card[9]. Wang et al. proposed an authentication and key agreement scheme preserving the privacy of secret keys[8]. Furthermore, because user anonymity is an important issue in modern applications to protect users from being tracked, they also extended the first scheme to provide user anonymity.

The following requirements are essential to smart-card-based password authentication schemes preserving user anonymity[4]-[8].

1) The remote server does not need to maintain a password or verification table.

2) The scheme should be invulnerable to security problems such as smart card loss problem and privileged administrator attack.

3) The scheme can defend against famous attack such
as impersonate attack, replay attack, known-key attack, and password guessing attack.

4) The scheme should not be inclined to problems of clock synchronization and delay-time limitation.

5) The remote server and the client can establish a session key after mutual authentication to protect future communications.

6) The scheme should provide perfect forward secrecy even if one’s long-term secret is compromised by a malicious adversary.

7) The lost smart card can be revoked without changing the ordinary identity.

8) The remote server can detect an evicted user using overdue information.

9) The remote server should not know a user’s identity and password when the scheme is applied in a privacy concerned environment.

Later, Chang et al. indicated that Wang et al.’s scheme, providing user anonymity, suffers from two security flaws[10]. First, it cannot resist DoS (denial-of-service) attack because the indicators for the next session are not consistent. Second, the user password may be modified by a malicious attacker because no authentication mechanism is applied before the user password is updated.

In this paper, we propose an improved user authentication and key agreement scheme preserving user anonymity. The proposed scheme not only satisfies requirements mentioned above but also eliminates the security vulnerabilities of previous schemes. The remainder of this paper is organized as follows. Section 2 briefly reviews Wang et al.’s scheme and shows the corresponding security flaws. Our improvement scheme is presented in Section 3. In Section 4, security analyses of the improved scheme are made. Finally, some conclusions are drawn in Section 5.

2. Review and Security Weakness of Wang et al.’s Scheme

2.1 Notations

In this section, the used notations throughout this paper are listed as follows:

- $U_i$: a user;
- $S_j$: a remote server;
- $SC_i$: the smart card that $U_i$ holds;
- $cid$: the identity of $SC_i$;
- $pw_i$: the password chosen by $U_i$;
- $h()$: a one-way hash function;
- $E_k(M)$: a symmetric encryption algorithm using a key $K$ to encrypt the message $M$;
- $D_k(M)$: a symmetric decryption algorithm using a key $K$ to decrypt the message $M$;
- $\oplus$: an exclusive-OR (XOR) operation;
- $x$: the master key of $S_j$ which cannot be derived by the brute force attack.

2.2 Review of Wang et al.’s User Anonymity Scheme

In this section, we review Wang et al.’s scheme which preserves user anonymity[8]. Their scheme is composed of six phases: registration phase, precomputation phase, authentication and key agreement phase, password changing phase, revoking smart card phase, and user eviction phase. The details are as follows.

A. Registration Phase

To initialize the system, $S_j$ selects a large prime $p$ and two integers $a$ and $b$, where $p \geq 2^{160}$ and $4a^3 + 27b^2 \mod p \neq 0$. Then $S_j$ chooses an elliptic curve equation $E_p$ over finite field $p$: $y^2 = x^3 + ax + b \mod p$. $G$ is a base point of $E_p$ with a prime order $n$, and $nG = O$, where $n > 2^{160}$. When a user $U_i$ wants to access $S_j$, $U_i$ needs to register at $S_j$ as follows.

Step 1) $U_i$ sends a registration request to $S_j$.

Step 2) Upon receiving the registration request, $S_j$ issues an indicator IND, for $U_i$ and computes $B_i = h(x||IND||cid_i)G$, where $x$ is the master key of $S_j$.

Step 3) $S_j$ stores $(IND_i, B_i, G, E_p)$ into $SC_i$ and issues this smart card to $U_i$ via a secure channel. Meanwhile, $S_j$ maintains an ID table which includes $(IND_i, cid_i)$.

Step 4) After $U_i$ receives $SC_i$, $U_i$ activates $SC_i$ by inserting it into a card reader and inputting an easy-to-remember password $pw_i$. Then $SC_i$ computes $B_i' = B_i \oplus h(pw_i)$ and replaces $B_i$ with $B_i'$. Finally, $SC_i$ stores $(IND_i, B_i', G, E_p)$.

B. Precomputation Phase

In this phase, $SC_i$ can compute $T_j$, which will be used in authentication and key agreement phase. First, $SC_i$ chooses a random number $R$ in $Z_n$, and computes $T_j = RG$. Then, $SC_i$ stores $T_j$ into its memory. Finally, $SC_i$ contains $(IND_i, B_i', G, E_p, T_j)$.

C. Authentication and Key Agreement Phase

When $U_i$ wants to access $S_j$’s service, $U_i$ first inserts $SC_i$ into a card reader and inputs $pw_i$. Then $SC_i$ and $S_j$ will execute the authentication and key agreement procedure. Finally, $U_i$ and $S_j$ authenticate each other and share a common session key which can be used for secure communication. Moreover, the indicator will be renewed for the next session. The details are as follows:

Step 1: After $U_i$ inserts $SC_i$ into a card reader and inputs $pw_i$, $SC_i$ computes $B_i = B_i' \oplus h(pw_i) = h(x||IND||cid_i)G$ and $T_j = h(RB_i)$.

Step 2: $U_i$ sends $(IND_i, T_j, T_j)$ to $S_j$.

Step 3: After getting $(IND_i, T_j, T_j)$, $S_j$ checks the format
of IND, computes \( T'_2 = T_1 h(x||IND||cid) \), and checks if the digest value of \( T'_2 \) is equal to \( T_2 \).

Step 4: \( S_i \) selects a random number \( W \) in \( Z_4^* \) and a new indicator \( \text{IND}_{\text{new}} \). Then \( S_i \) computes \( K_i = h(WT_1) \), \( V_i = E_{K_i}(h(T'_1 + 1)||\text{IND}_{\text{new}}||B_{\text{new}}) \) and \( T_3 = WG \), where \( B_{\text{new}} = h(x||\text{IND}_{\text{new}}||cid)G \).

Step 5: \( S_i \) sends \( (T_3, V_i) \) to \( U_i \).

Step 6: Upon receiving the message \( (T_3, V_i) \), SCi computes \( K'_i = h(RT_1) \), \( D_{K_i}(V_i) \), and \( V_2 = h(RB_i + 2) \).

Step 7: \( U_i \) checks if \( h(RB_i + 1) \) is included in the decryption result of \( V_i \). If it holds, \( S_i \) is authenticated by \( U_i \). Then SCi replaces \( (\text{IND}, B_i) \) with \( (\text{IND}_{\text{new}}, B_{\text{new}}) \) and sends \( V_2 \) to \( S_i \).

Step 8: After getting \( V_2 \), \( S_i \) checks if \( V_2 = h(T'_2 + 2) \). If it holds, \( U_i \) is authenticated by \( S_i \), and \( S_j \) updates the ID table with \( (\text{IND}_{\text{new}}, \text{cid}) \).

After above steps, \( U_i \) and \( S_j \) share a common session key \( K = K'_i \) for secure communication.

D. Password Changing Phase

This phase is invoked whenever \( U_i \) wants to change his/her password \( pw_i \). \( U_i \) first inserts SCi into a card reader and inputs the original password \( pw_i \) and the new password \( \text{new}_{pw_i} \). Then SCi computes \( B'_i = B'_i \oplus h(pw_i) = h(x||\text{IND}||\text{cid})G \) and \( B''_i = B''_i \oplus h(\text{new}_{pw_i}) \). Finally, SCi replaces \( B'_i \) with \( B''_i \) and stores \( (\text{IND}, B''_i, G, E_p) \).

E. Revoking Smart Card Phase

This phase is invoked whenever \( U_i \) wants to revoke a lost smart card, \( U_i \) still can use the previous password and indicator to register again. The details are as follows.

Step 1: \( S_i \) computes \( B'' = h(x||\text{IND}||\text{cid}_{\text{new}})G \), then writes \( (\text{IND}, B''_i, G, E_p) \) into the new smart card SCi and issues it to \( U_i \), where \( \text{cid}_{\text{new}} \) is the indicator of the new smart card SCi.

Step 2: \( S_i \) replaces \( (\text{IND}, \text{cid}) \) with \( (\text{IND}, \text{cid}_{\text{new}}) \).

Step 3: Upon receiving the smart card, \( U_i \) activates SCi by inserting it into a card reader and inputting \( pw_i \). Then SCi computes \( B'_i = B'_i \oplus h(pw_i) \) and replaces \( B'_i \) with \( B''_i \).

Finally, SCi stores \( (\text{IND}, B''_i, G, E_p) \).

F. User Eviction Phase

This phase is invoked when a client is evicted by the server. The server will delete the client’s indicator and the record in the ID table. When an evicted user wants to login to the server by using the overdue information in the smart card, the server can detect him/her by checking the record in the ID table.

2.3 Weakness of Wang et al.’s Scheme

Although Wang et al. claimed that their scheme providing user anonymity was secure to resist well-known attack, Chang et al. found that their scheme suffers from some security flaws[10]. First, it cannot resist DoS attack because the indicators for the next session are not consistent. Second, the user password may be modified by a malicious attacker because no authentication mechanism is applied before the user password is updated. In the following, the details are given.

A. Dos Attack

Because data is transmitted over public but insecure channels, a malicious user may intercept and modify the transmitted messages. In authentication and key agreement phase, SCi replaces \( (\text{IND}, B_i) \) with \( (\text{IND}_{\text{new}}, B_{\text{new}}) \) and sends \( V_2 \) to \( S_j \). After \( S_j \) is authenticated by \( S_j \), \( S_j \) replaces (IND, cid) with (IND, cid). From now on, indicators kept by \( S_j \) and \( S_j \) are \( \text{IND}_{\text{new}} \) and \( \text{IND}_{\text{new}} \), respectively. Later, if \( U_i \) sends \( (\text{IND}_{\text{new}}, T_1, T_2) \) to \( S_j \) as a request, this request will be rejected by \( S_j \) because no entry stored in the ID table is matched.

B. Password Changing without Verification

In password changing phase of Wang et al.’s scheme, a user inserts his/her personal smart card into a card reader and input the original and new passwords to update his/her password. Unfortunately, no verification is involved such that a malicious user can get a legal user’s smart card and modify the legal user’s password such that the innocent user cannot login to the system.

3. Proposed Improvement

Wang et al.’s scheme possesses superior properties although it suffers from security flaws mentioned above. To eliminate the security flaws and preserve the advantages of Wang et al.’s scheme, we propose an improvement. The proposed scheme is also composed of six phases: registration phase, precomputation phase, user authentication and key agreement phase, password changing phase, revoking smart card phase, and user eviction phase. Because precomputation phase and user eviction phase are identical to those of Wang et al.’s scheme, only registration phase, user authentication and key agreement phase, password changing phase, and
revoking smart card phase are shown. The details are as follows.

3.1 Registration Phase

The steps in registration phase of the proposed improvement are almost the same as those of Wang et al.’s scheme except the followings.

In Step 3, $S_j$ stores $(IND_{old}, IND, B_{old}, B, G, E_p)$ into the smart card, issues this smart card to $U_i$ via a secure channel, and saves the entry $(IND, cid)$ in the ID table, where $IND_{old}$=IND, and $B_{i}=B_{old}=h(x)||IND||cid)$. $G=h(x)||IND_{old}||cid)$. Though $IND_{old}$=IND, and $B_{i}=B_{old}$ in registration phase, $SC_i$ has to store them to protect the proposed improvement from DoS attack that Chang et al. found. Finally, $SC_i$ contains $(IND_{old}, IND, B_{old}', B', G, E_p)$, where $B_i'=B_i\oplus h(pw_i)$ and $B_{old}'=B_{old}\oplus h(pw_i)$.

3.2 User Authentication and Key Agreement Phase

When $U_i$ wants to access $S_j$’s service, this phase will be invoked. There are two cases in this phase: 1) $U_i$ is authenticated by sending the login request $(IND, T_1, T_2)$ and 2) $U_i$ is authenticated by sending the login request $(IND_{old}, T_1, T_2)$. User authentication and key agreement phase is depicted in Fig. 1. The details are as follows.

A. Case 1

Step 1: $U_i$ inputs $pw_i$, and $SC_i$ computes $B_{i}=B_i'\oplus h(pw_i)$ and $T_2=h(RB)$. $U_i$ sends $(IND, T_1, T_2)$ to $S_j$.

Step 2: $S_j$ checks whether IND is in the ID table. If it does not hold, $S_j$ aborts this request and proceeds to Case 2 to start over the session. If it holds, $S_j$ computes $T_2'=T_2\oplus h(x)||IND||cid)$ and checks if the digest value of $T_2'$ is equal to $T_2$. If it holds, $S_j$ selects a random number $W$ in $Z_n^*$, computes a symmetric session key $K_{i'=h(WT)}$, $V_i=E_{K_i}(h(T_2+1)||IND_{new}||B_{new})$ and an authentication message $T_1=WG$, and sends $(T_1, V)$ to $U_i$, where $IND_{new}$ is issued by $S_j$ and only the legal user can retrieve it.

Step 3: After receiving $(T_1, V_i)$, $SC_i$ computes the session key $K_{i'}=h(RT_i)$, uses $K_{i'}$ to decrypt $V_i$, and checks if $h(RB_{i'}+1)$ is included in the decrypted result. If it holds, $U_i$ is convinced that $S_j$ is a legal server. Then $SC_i$ computes $V_i'=h(RB_{i'}+2)$, sends $V_2$ to $S_j$, and replaces $(IND_{old}, IND, B_{i}'_{old}, B_{i}'')$ with $(IND, IND_{new}, B_i', B_{new}'')$, where $B_{new}'=B_{new}\oplus h(pw_i)$.

Step 4: Upon receiving $V_2$, $SC_i$ computes $V_2'=h(T_2'+2)$ and checks if $V_2'=V_2'$. If it holds, $S_j$ is convinced that $U_i$ is an authorized user. After above steps are finished, $U_i$ and $S_j$ establish a session key $K_i=K_{i'}$ and they can employ this session key to provide the confidentiality of subsequent communications. Finally, $S_j$ renews the entry in the maintained ID table with $(IND_{new}, cid)$.

B. Case 2

This case is invoked when $U_i$’s login request $(IND, T_1, T_2)$ fails and the session is restarted by sending $(IND_{old}, T_1, T_2)$ to $S_j$. Processes in Case 2 are almost identical to those in Case 1. Only differences are shown as follows.

In Step 1, $SC_i$ computes $B_{i}=B_{old}'\oplus h(pw_i)$ and $T_2=h(RB)$ and sends $(IND_{old}, T_1, T_2)$ to $S_j$. After $S_j$ receives $(IND_{old}, T_1, T_2)$, $S_j$ checks if the received IND_{old} is in the ID table. If it holds, $S_j$ computes $T_2'=T_2\oplus h(x)||IND_{old}||cid)$ and checks if the digest value of $T_2'$ is equal to $T_2$. In Step 3, $SC_i$ replaces $(IND_{old}, IND, B_i', B_{old}'', B_{i}'')$ with $(IND_{old}, IND_{new}, B_{old}'', B_{new}'')$, where $B_{new}'=B_{new}\oplus h(pw_i)$. In Step 4, $S_j$ renews the entry in maintained ID table with $(IND_{new}, cid)$.

In the proposed improvement, an attacker cannot mount DoS attack which Wang et al.’s scheme suffers from. If $S_j$ does not renew the ID table with $(IND_{new}, cid)$ when an attacker modifies the transmitted message, the legal user $U_i$ still can be authenticated by sending $(IND_{old}, T_1, T_2)$ to $S_j$ in Case 2. 

3.3 Password Changing Phase

This phase is invoked when a user wants to change his/her password. The user first inputs original and new passwords, $pw_i$ and $new_{pw_i}$. Before changing the user’s password, user authentication and key agreement phase needs to be performed. If the user is authenticated successfully by the server, the original password stored in the smart card will be updated with the new one.

There are two cases in this phase: 1) $U_i$ is authenticated by sending the login request $(IND, T_1, T_2)$ in the user authentication and key agreement phase and 2) $U_i$ is authenticated by sending the login request $(IND_{old}, T_1, T_2)$ in the user authentication and key agreement phase.

A. Case 1

$U_i$ is authenticated by sending $(IND, T_1, T_2)$ in the user authentication and key agreement phase. $SC_i$ replaces $(B_i', B_{new})$ with $(B_i'\oplus h(new_{pw_i})$, $B_{new}''\oplus h(new_{pw_i})$).

B. Case 2

$U_i$ is authenticated by sending the login request $(IND_{old}, T_1, T_2)$ in the user authentication and key agreement phase. $SC_i$ replaces $(B_{old}', B_{new})$ with $(B_{old}'\oplus h(new_{pw_i})$, $B_{new}''\oplus h(new_{pw_i})$).
3.4 Revoking Smart Card Phase

This phase is almost the same as that of Wang et al.'s scheme except the followings.

Step 1: \( S_i \) computes \( B_i = h(x) | IND_i || cid_{new} || G \) and \( B_{old} = h(x) | IND_{old} || cid_{new} \), stores \( IND_{old}, IND_i, B_{old}, B_i, G, E_p \) into the new smart card \( SC_i \) and, issues it to \( U_i \), where \( cid_{new} \) is the identity of \( SC_i \).

Step 2: \( S_i \) replaces \( \text{IND}_i, \text{cid}_i \) with \( \text{IND}_{old}, \text{cid}_{new} \).

Step 3: Upon receiving the smart card, \( U_i \) activates \( SC_i \) by inserting it into a card reader and inputting \( pw_c \). Then \( SC_i \) computes \( B_i' = B_i \oplus h(pw_c) \) and \( B_{old}' = B_{old} \oplus h(pw_c) \) and replaces \( B_i \) with \( B_i' \). Finally, \( SC_i \) stores \( IND_{old}, IND_{old}, B_{old}', B_i', G, E_p \).

4. Security Analyses

In this section, the security analysis of the proposed scheme are given to show that it achieves aforementioned security requirements.

4.1 No Verification Table

In the proposed scheme, no verification table or password table is maintained by the server. The remote server only has to record \( \text{IND}_i, \text{cid}_i \) and utilizes them to authenticate users.

4.2 Resistance of Smart Card Loss Problem

The smart card stores \( IND_{old}, IND_i, B_{old}', B_i', G, E_p \) in it. Assume that an attacker gets someone's smart card and extracts data stored in the smart card. It is hard for an attacker to retrieve \( B_i \) without knowing the client's password \( pw_c \). If the attacker performs online password guessing attack in the user authentication and key agreement phase to obtain \( pw_c \), this attack will be detected by the remote server.

4.3 Resistance of Administrator Attack

The server only maintains \( IND_{old}, IND_i, \text{cid}_i \) for the user \( U_i \). No information related to the user's password \( pw_c \) can be obtained. Thus, administrator attack cannot be...
successfully mounted on the proposed scheme.

If the server gets an activated smart card and extracts data \((\text{IND}_{\text{old}}, \text{IND}_i, B_{i\text{old}}', B_i', G, E_p)\) stored in it, the server computes \(B_i = h(x||\text{IND}_i||\text{cid}_i)\) and checks if \(B_i = B_{i'} \oplus h(p'_w)\) by guessing \(p'_w = p_w\). If it does not hold, the server guesses \(U_i\)'s password and checks if \(B_i = B_{i'} \oplus h(p'_w)\) until it holds. When \(B_i = B_{i'} \oplus h(p'_w)\), \(p'_w\) does not absolutely equal \(p_w\). Instead, it denotes that the hash values of \(p'_w\) and \(p_w\) are equal. However, this assumption is unreasonable because efforts and benefits are not equivalent.

4.4 Resistance of Replay Attack and Clock Synchronization Problem

An attacker might eavesdrop while the server and the user start the session. The attacker intercepts the login request (IND, \(T_i, T_2\)) or \((\text{IND}_{\text{old}}, T_i, T_2)\) and forwards to the server. However, the server will detect replay attack because login requests in different sessions differ from each other. Even if \(V_2\) in this session is modified such that the server does not update the ID table with the new indicator, replay attack still cannot be mounted successfully because the server always chooses a new random number \(W\) such that only the legal user can compute the correct authentication parameter \(V_2\). In addition, there is no clock synchronization problem since the proposed scheme employs no timestamp to solve replay attack.

4.5 Resistance of Impersonation Attack

The proposed scheme can resist impersonation attack on both server side and client side.

Server side: It is hard for an attacker to compute \(T_3\) and \(V_1\) without knowing the master key \(x\) and the random nonce \(W\).

Client side: The attacker cannot compute the correct \(V_2\) without knowing the random nonce \(R\) and the secret value \(B_i\).

4.6 Resistance of Known-Key Attack and Perfect Forward Secrecy

Suppose that an attacker obtains a session key of one previous session. The attacker still cannot derive the latest session key because the session key is negotiated with the secret \(B_i\) and random numbers \(W\) and \(R\). If the long-term key \(B_i\) is retrieved by an attacker, he cannot obtain previous session keys because the session key is negotiated with random numbers \(W\) and \(R\). Therefore, our improved scheme provides perfect forward secrecy.

4.7 Mutual Authentication and User Anonymity

In the user authentication and key agreement phase, a remote server and user can authenticate each other such that no malicious user can impersonate any participant. On the other hand, the transmitted indicator will be updated in each session so no one can trace the user by eavesdropping. Thus, the proposed scheme provides mutual authentication and user anonymity.

5. Conclusions

In this paper, we review Wang et al.'s authentication and key agreement scheme preserving the privacy of the client and the security flaws which suffers from DoS attack. We propose an improvement to eliminate the security flaws and preserve the advantages of Wang et al.'s scheme. The proposed scheme achieves requirements essential to smart-card-based password authentication schemes preserving user anonymity, and the computation load is light because only simple operations are executed. Via the proposed scheme, a legal user can negotiate the shared session key with the server without leaking any secret and preserving user anonymity at the same time. These properties make the proposed scheme suit applications with computation efficiency and user anonymity taken into consideration.

References


Ya-Fen Chang received the B.S. degree in computer science and information engineering from National Chiao Tung University in 2000 and the Ph.D. degree in computer science and information engineering in 2005 from National Chung Cheng University. From August 2006 to March 2010, she worked as an assistant professor with the Department of Computer Science and Information Engineering, National Taichung Institute of Technology, where she has been an associate professor since April 2010. Her current research interests include electronic commerce, information security, cryptography, and mobile communications.

Pei-Yu Chang received the B.S. degree in information management from National Taichung Institute of Technology in 2010. He has been a graduate student with the Department of Computer Science and Information Engineering, National Taichung Institute of Technology since September 2010. His current research interests include electronic commerce, information security, cryptography, and computer networks.