PAPER Special Section on Trust, Security and Privacy in Computing and Communication Systems

A Privacy-Preserving Dynamic ID-Based Remote User Authentication Scheme with Access Control for Multi-Server Environment

Min-Hua SHAO[†a)], ***Member and*** **Ying-Chih CHIN**[†], ***Nonmember***

SUMMARY Since the number of server providing the facilities for users is usually more than one, remote user authentication schemes used for multi-server architectures, rather than single server circumstance, is considered. As far as security is concerned, privacy is the most important requirements, though some other properties are also desirable in practice. Recently, a number of dynamic ID-based user authentication schemes have been proposed. However, most of those schemes have more or less weaknesses and/or security flaws. In the worst case, user privacy cannot be achieved since malicious servers or users can mount some attacks, i.e., server spoofing attack and impersonation attack, to identify the unique identifier of users and masquerade of one entity as some other. In this paper, we analyze two latest research works and demonstrate that they cannot achieve true anonymity and have some other weaknesses. We further propose the improvements to avoid those security problems. Besides user privacy, the key features of our scheme are including no verification table, freely chosen password, mutual authentication, low computation and communication cost, single registration, session key agreement, and being secure against the related attacks.

key words: *anonymity, single registration, key agreement, smart card, security*

1. Introduction

Remote password authentication scheme is a method to authenticate remote users over an insecure channel and it has inspired a large variety of work [21]. A big step towards more efficient and secure solutions was that no password file or verification table is required to keep in a system for verifying the legitimacy of the login users [5], [19]. Sun further proposed a revised version based on one-way function to significantly reduce the communication and computation costs [13]. In order to protect from ID-theft, Das et al. proposed a dynamic ID-based remote user authentication scheme using smart cards [3]. The revised version of Das et al.'s scheme is provided by Chien and Chen to conquer the weakness of the protection of user's anonymity [2]. Besides the protection of user privacy against outside attacks, Kim et al.'s effort [8] is to provide traceable anonymity authentication. When perceiving a user doing a malicious act, the remote server will be able to trace the malicious user by receiving help from a trust agency. Since the number of server providing the facilities for users is usually more than one, remote user authentication schemes used for multi-server architectures, rather than single-server circumstance, is considered [20], [22]. These conventional single-server password authentication schemes cannot be directly applied to multi-server environment because each user needs to remember different sets of identities and passwords [16]–[18]. Li et al. [9] presented a remote password authentication scheme for multi-server environments in which the password authentication system is a pattern classification system based on an artificial neural network. An efficient solution with much less computational cost and key agreement is given in Juang's work [7]. Liao and Wang [11] indicated that a threat to user privacy is caused by the use of static ID for password authentication. Instead, a secure dynamic ID-based scheme to achieve user's anonymity is proposed in their work. Then, an efficient improvement over Liao–Wang's scheme is given by Hsiang and Shih in 2009. However, we demonstrate in this paper that the aforementioned two schemes suffer from a few severe security problems, and some of the security properties, especially, user's anonymity cannot be satisfied. For example, the computed data associated to the fixed secret identifier can be trace and identify the profile of the user until the user changes his/her password. A shared secret value among servers is not secure against server spoofing attack and impersonation attack [15]. In this paper, we give a thorough security analysis of the Liao-Wang's scheme and further propose the improvements to avoid these problems. The proposed scheme can meet the following set of security properties on a remote authentication scheme: (1) no verification table, (2) freely chosen password, (3) mutual authentication, (4) low computation and communication cost, (5) single registration, (6) session key agreement, (7) user anonymity, (8) access control, and (9) security.

The rest of this paper is structured as follows. In Sects. 2 and 3, we analyze Liao-Wang's and Hsiang-Shih's schemes, respectively. Section 4 presents the proposed scheme in detail. Section 5 gives the discussion of security, performance and functionality analysis on our scheme. Section 6 concludes the paper with a summary of our achievements.

Manuscript received March 8, 2011.
Manuscript revised June 29, 2011.

[†]The authors are with the Department of Management Information Systems, National Pingtung University of Science & Technology, Pingtung 912, Taiwan.
a) E-mail: mhshao@mail.npust.edu.tw

DOI: 10.1587/transinf.E95.D.161

2. Review of Liao-Wang's Scheme

2.1 The Liao-Wang's Scheme

For completeness and readability, the notations used in this paper are defined in Table 1. In 2007, Liao and Wang proposed a "secure dynamic ID based remote user authentication scheme for multi-server environment" that uses dynamic ID instead of static ID to achieve user's anonymity for verifying the legitimacy of a remote login user [11]. There are three different roles, including of the user, server and register center, and four phases, including of registration phase, login phase, mutual verification and session key agreement phase, and password change phase, involved in the scheme.

A. Registration phase
U_i chooses ID_i and PW_i and then submits them to RC. Upon receiving data, RC conducts the following steps:

Step 1: RC computes $T_i = h(ID_i||x)$, $V_i = T_i \oplus h(ID_i||PW_i)$, $B_i = h(PW_i) \oplus h(x)$ and $H_i = h(T_i)$.
Step 2: RC stores V_i, B_i, H_i, $h()$ and y into a smart card, and issues it to U_i via a secure channel.

B. Login phase
U_i presents smart card and keys his/her ID_i^*, PW_i^* and SID_j. Then, the smart card performs the following steps:

Step 1: Compute $T_i^* = V_i \oplus h(ID_i^*||PW_i^*)$, check whether H_i and $h(T_i^*)$ is met or not, and reject the case if the verification doesn't hold.
Step 2: Generate a nonce N_i and compute $CID_i = h(PW_i) \oplus h(T_i||y||N_i)$, $P_{ij} = T_i \oplus h(y||N_i||SID_j)$ and $Q_i = h(B_i||y||N_i)$.
Step 3: Send the login request message $<CID_i, P_{ij}, Q_i, N_i>$ to S_j.

Table 1 Notations.

Notation	Definition
U_i	The i_{th} user
S_j	The j_{th} server
RC	The registration center
ID_i	The identification of U_i
SID_j	The identification of S_j
PW_i	The password of U_i
CID_i	The dynamic ID of U_i
$\oplus$	The exclusive-or operation
$\|\|$	The concatenation operation
$h()$	A one-way hash function
x	The master secret key of registration center
y, r	Two secret number of registration center
$\Rightarrow$	A secure channel
$\rightarrow$	A common channel

C. Mutual verification and session key agreement phase

Upon receiving the login request message, the service provider S_j authenticates the user U_i with the following steps:

Step 1: Compute $T_i = P_{ij} \oplus h(y||N_i||SID_j)$, $h(PW_i) = CID_i \oplus h(T_i||y||N_i)$ and $B_i = h(PW_i) \oplus h(x)$.
Step 2: Compare the computed value $h(B_i||N_i||y)$ with Q_i and abort the session if any component is invalid.
Step 3: Compute $M_{ij1} = h(B_i||N_i||y||SID_j)$ with nonce N_j, then send back the message $<M_{ij1}, N_j>$ to U_i.

When receiving the acknowledge message, the user U_i performs the following steps:

Step 4: Compare the computed value $h(B_i||N_i||y||SID_j)$ with M_{ij1}. If the verification holds, the legality of the S_j is authenticated; otherwise, the connection is terminated.
Step 5: Compute and send $M_{ij2} = h(B_i||N_j||y||SID_j)$ to S_j.

When receiving the message M_{ij2}, the service provider S_j responds to the following steps:

Step 6: Check the computed value $h(B_i||N_j||y||SID_j)$ with M_{ij2}. If it is hold, the identity of U_i can be assured.

After conducting mutual authentication, the session key $SK = h(B_i||N_i||N_j||y||SID_j)$ can be computed by S_j and U_i, respectively.

D. Password change phase
The user U_i presents the smart card with (ID_i^*, PW_i^*) and conducts the steps of login phase to verify the identity of the cardholder. After the authenticity of U_i is assured, the smart card allows the cardholder to resubmit a new password PW_i^{new}, computes $V_i^{new} = T_i \oplus h(ID_i||PW_i^{new})$ and $B_i^{new} = B_i \oplus h(PW_i) \oplus h(PW_i^{new})$, and updates the correspondents stored in the card.

2.2 Security Analysis of the Liao-Wang's Scheme

In Liao and Wang's effort, the use of dynamic-ID instead of static ID for password authentication is given due to user's anonymity. However, their scheme is not effective against server spoofing attack and impersonation attack, and failed to achieve the purpose of user anonymity. The key point to the failure is two secret values ($h(x)$, y) shared among service providers that observe the login request message $<CID_i, P_{ij}, Q_i, N_i>$ and perform the following operations to trace the identity of users.

$$T_i = P_{ij} \oplus h(y||N_i||SID_j) \tag{1}$$

$$h(PW_i) = CID_i \oplus h(T_i||y||N_i) \tag{2}$$

$$B_i = h(PW_i) \oplus h(x) \tag{3}$$

- Server spoofing attack: Once the user initiates the login phase, each of service providers can compute (T_i, $h(PW_i)$, B_i) and respond the computed value $M_{ij1} = h(B_i||N_i||y||SID_j)$ to the user. Figure 1 illustrates the details of the attacking steps.

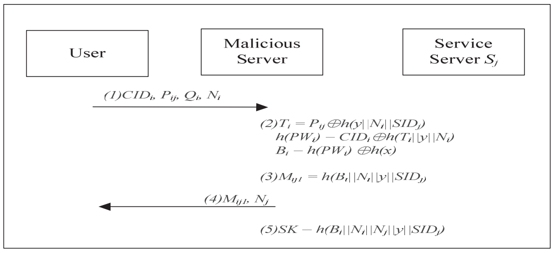
User
Malicious Server
Service Server S_j
(1)CID_i, P_{ij}, Q_i, N_i
(2)$T_i = P_{ij} \oplus h(y\|N_i\|SID_j)$
$h(PW_i) - CID_i \oplus h(T_i\|y\|N_i)$
$B_i - h(PW_i) \oplus h(x)$
(3)$M_{ij1} = h(B_i\|N_i\|y\|SID_j)$
(4)M_{ij1}, N_j
(5)$SK - h(B_i\|N_i\|N_j\|y\|SID_j)$

Fig. 1 Server spoofing attack on Liao-Wang's Scheme.

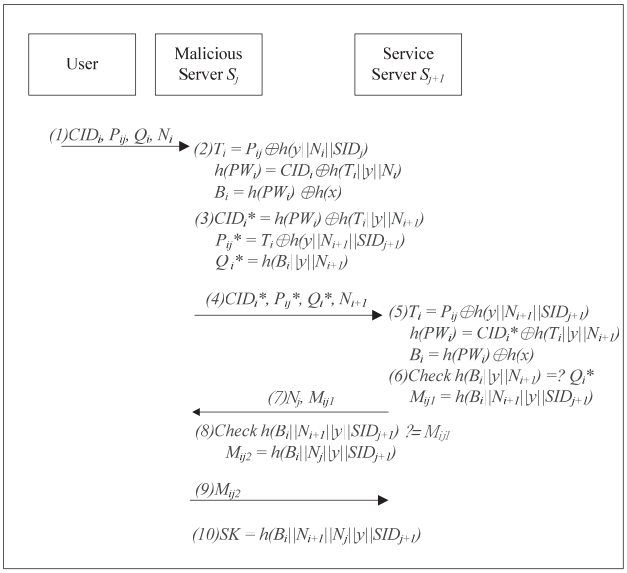
User
Malicious Server S_j
Service Server S_{j+1}
(1)CID_i, P_{ij}, Q_i, N_i
(2)$T_i = P_{ij} \oplus h(y\|N_i\|SID_j)$
$h(PW_i) = CID_i \oplus h(T_i\|y\|N_i)$
$B_i = h(PW_i) \oplus h(x)$
(3)$CID_i^* = h(PW_i) \oplus h(T_i\|y\|N_{i+1})$
$P_{ij}^* = T_i \oplus h(y\|N_{i+1}\|SID_{j+1})$
$Q_i^* = h(B_i\|y\|N_{i+1})$
(4)$CID_i^*, P_{ij}^*, Q_i^*, N_{i+1}$
(5)$T_i = P_{ij} \oplus h(y\|N_{i+1}\|SID_{j+1})$
$h(PW_i) = CID_i^* \oplus h(T_i\|y\|N_{i+1})$
$B_i = h(PW_i) \oplus h(x)$
(6)Check $h(B_i\|y\|N_{i+1}) =? Q_i^*$
$M_{ij1} = h(B_i\|N_{i+1}\|y\|SID_{j+1})$
(7)N_j, M_{ij1}
(8)Check $h(B_i\|N_{i+1}\|y\|SID_{j+1})$?= M_{ij1}
$M_{ij2} = h(B_i\|N_j\|y\|SID_{j+1})$
(9)M_{ij2}
(10)$SK = h(B_i\|N_{i+1}\|N_j\|y\|SID_{j+1})$

Fig. 2 Impersonation attack on Liao-Wang's Scheme.

- Impersonation attack: The malicious service provider can use $(T_i, h(PW_i), B_i)$ to login other servers on behalf of the legal of the user by computing the login request message. The detail treatment of impersonation attack is shown in Fig. 2.
- Invalid anonymity: The goal of user anonymity is to achieve by the mask CID_i of the real identity of users. However, malicious servers can get T_i by computing $P_{ij} \oplus h(N_i^*\|y\|SID_j*)$. $T_i = h(ID_i\|x)$ is a fixed value that aims to trace the behavior of the specific user.

3. Review of Hsiang-Shih's Scheme

3.1 The Hsiang-Shih's Scheme

Hsiang and Shih in 2009 [6] provided the improvements to Liao-Wang's work. In their work, there are three kinds of participants involved, that is, the user, the server and register center. It consists of four phases: registration phase, login phase, mutual verification and session key agreement phase and password change phase. The notations please refer to Table 1. In contrast to Liao-Wang's scheme, only *RC* knows the master secret key x and two secret numbers r and y.

A. Registration phase

- Server registration: A secret key $h(SID_j\|y)$ is computed by *RC* and kept at a service provider S_j.
- User registration: The user U_i submits ID_i and PW_i to *RC*, and *RC* performs the following steps:

Step R1: U_i freely selects a password PW_i, an arbitrary number b and computes $h(b \oplus PW_i)$.
Step R2: $U_i \Rightarrow RC$: ID_i and $h(b \oplus PW_i)$.
Step R3: *RC* computes $T_i = h(ID_i\|x)$, $V_i = T_i \oplus h(ID_i\|h(b \oplus PW_i))$, $A_i = h(h(b \oplus PW_i\|r) \oplus h(x \oplus r)$, $B_i = A_i \oplus h(b \oplus PW_i)$, $R_i = h(h(b \oplus PW_i)\|r)$ and $H_i = h(T_i)$.
Step R4: *RC* stores $<V_i, B_i, H_i, R_i, h()>$ into a smart card, and issues it to U_i via secure channel.
Step R5: U_i enters b into his/her smart card.

B. Login phase

In order to request services from S_j, U_i inserts the smart card into a card reader, and keys his/her ID_i, PW_i and SID_j. Then, the smart card performs the following steps:

Step L1: Compute $T_i = V_i \oplus h(ID_i\|h(b \oplus PW_i))$, $H_i^* = h(T_i)$, check whether H_i^* and H_i is equal or not, and abort the session if any component is invalid. Otherwise, U_i proceeds to the next step.
Step L2: Generate a nonce N_i and compute $A_i = B_i \oplus h(b \oplus PW_i)$, $CID_i = h(b \oplus PW_i) \oplus h(T_i\|A_i\|N_i)$, $P_{ij} = T_i \oplus h(A_i\|N_i\|SID_j)$, $Q_i = h(B_i\|A_i\|N_i)$, $D_i = R_i \oplus SID_j \oplus N_i$ and $C_0 = h(A_i\|N_i + 1\|SID_j)$.
Step L3: Send the login request message $<CID_i, P_{ij}, Q_i, D_i, C_0, N_i>$ to S_j.

C. Mutual verification and session key agreement phase

When receiving the login request message $<CID_i, P_{ij}, Q_i, D_i, C_0, N_i>$, S_j authenticates U_i with the following steps:

StepV1: Generate nonce N_{jr}, compute $M_{jr} = h(SID_j\|y) \oplus N_{jr}$, and forward $<M_{jr}, SID_j, D_i, C_0, N_i>$ to *RC*.

Upon receiving the message $<M_{jr}, SID_j, D_i, C_0, N_i>$, RC performs the following steps:

StepV2a: Compute $N_{jr}' = M_{jr} \oplus h(SID_j\|y)$, $R_i' = D_i \oplus SID_j \oplus N_i$ and $A_i' = R_i' \oplus h(x \oplus r)$.
StepV2b: Compare the computed value $C_0' = h(A_i'\|N_i + 1\|SID_j)$ with C_0 and terminate the session if the verification doesn't hold.
StepV2c: Generate nonce N_{rj}, compute $C_1 = h(N_{jr}'\|h(SID_j\|y)\|N_{rj})$, $C_2 = A_i \oplus h(h(SID_j\|y) \oplus N_{jr}')$, and send $<C_1, C_2, N_{rj}>$ back to S_j.

Upon receiving the response $<C_1, C_2, N_{rj}>$, S_j performs the following steps:

StepV3: Compare $C_1' = h(N_{jr}\|h(SID_j\|y)\|N_{rj})$ with C_1. If they are not equal, S_j reports a RC authentication error message and terminates this session.
StepV4: Compute $A_i = C_2 \oplus h(h(SID_j\|y) \oplus N_{rj})$, $T_i = P_{ij} \oplus h(A_i\|N_i\|SID_j)$, $h(b \oplus PW_i = CID_i \oplus h(T_i\|A_i\|N_i)$ and $B_i = A_i \oplus h(b \oplus PW_i)$.

StepV5: Compute $h(B_i||A_i||N_i)$ and compare it with Q_i. If they are not equal, S_j rejects the login request and terminates this session.

StepV6: Generate nonce N_j, compute $M_{ij}' = h(B_i||N_i||A_i||SID_j)$, and send the message $<M_{ij}', N_j>$ back to U_i.

Upon receiving the acknowledgement message $<M_{ij}', N_j>$, U_i performs the following steps:

StepV7: Compute $h(B_i||N_i||A_i||SID_j)$ and compare it with M_{ij}'. If they are equal, it indicates that the legitimacy of S_j is authenticated; otherwise, the connection is interrupted.

StepV8: Compute $M_{ij}'' = h(B_i||N_j||A_i||SID_j)$, and send back M_{ij}'' to S_j.

Upon receiving the message M_{ij}'', S_j responds to the following step:

StepV9: Compare $h(B_i||N_j||A_i||SID_j)$ with M_{ij}'' and the identity of U_i can be assured, if it is on hold.

After finishing mutual authentication phase, U_i and S_j compute $h(B_i||A_i||N_i||N_j||SID_j)$ as the session key.

D. Password change phase

When U_i wants to update his password without the help of *RC*, the steps are as follows:

Step C1: U_i inserts the smart card into reader device and inputs ID_i and PW_i.

Step C2: U_i's smart card computes $T_i = V_i \oplus h(ID_i||h(b \oplus PW_i))$ and $H_i* = h(T_i)$, checks whether H_i and H_i^* is equal or not, and rejects the case if the verification doesn't hold. Otherwise, U_i chooses a new password PW^{new}.

Step C3: U_i's smart card computes $V_i^{new} = T_i \oplus h(IDi||h(b \oplus PW^{new}))$ and $B_i^{new} = B_i \oplus h(b \oplus PW_i) \oplus h(b \oplus PW_i^{new})$. The parameter V_i^{new} and B_i^{new} are stored in the smart card to replace V_i and B_i, respectively.

3.2 Security Analysis of the Hsiang-Shih's Scheme

Hardware implementations of cryptosystems may leak sensitive information about the secret key [12]. The secret key of an executing cryptographic algorithm could be extracted by monitoring the power consumption of a smart card [14]. Unfortunately, Hsiang and Shih's work is failed to protect the secret values $(V_i, B_i, H_i, R_i, h(), b)$ stored in the smart card with some ways. The attacker can easily impersonate legitimate remote servers or users without knowing any password.

- Server spoofing attack: In Hsiang and Shih's scheme, *RC* only uses $h(x \oplus r)$ to verify the legitimacy of a user. From insider attacks, a legal user or remote server can impersonate as any other remote server because the legal user is possession of $h(x \oplus r)$ and the session key SK during login phase.

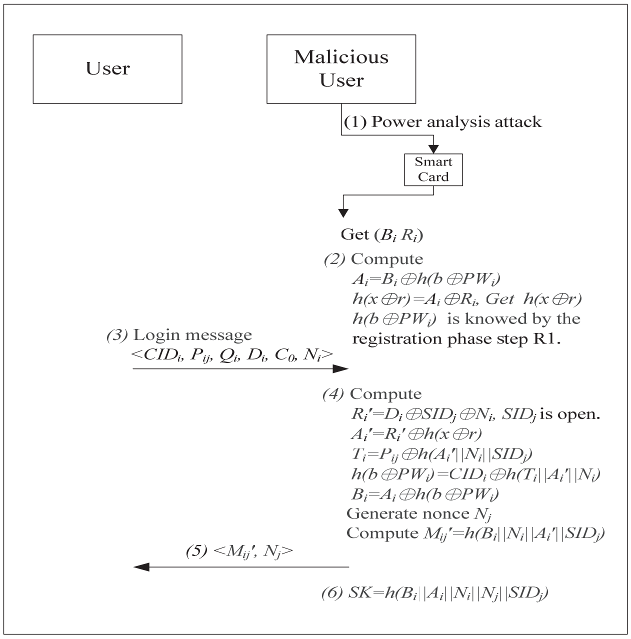
User
Malicious User
(1) Power analysis attack
Smart Card
Get (B_i R_i)
(2) Compute
$A_i = B_i \oplus h(b \oplus PW_i)$
$h(x \oplus r) = A_i \oplus R_i$, Get $h(x \oplus r)$
$h(b \oplus PW_i)$ is knowed by the registration phase step R1.
(3) Login message
$<CID_i, P_{ij}, Q_i, D_i, C_0, N_i>$
(4) Compute
$R_i' = D_i \oplus SID_j \oplus N_i$, SID_j is open.
$A_i' = R_i' \oplus h(x \oplus r)$
$T_i = P_{ij} \oplus h(A_i'||N_i||SID_j)$
$h(b \oplus PW_i) = CID_i \oplus h(T_i||A_i'||N_i)$
$B_i = A_i \oplus h(b \oplus PW_i)$
Generate nonce N_j
Compute $M_{ij}' = h(B_i||N_i||A_i'||SID_j)$
(5) $<M_{ij}', N_j>$
(6) $SK = h(B_i||A_i||N_i||N_j||SID_j)$

Fig. 3 Server spoofing attack by malicious users on Hsiang-Shih's scheme.

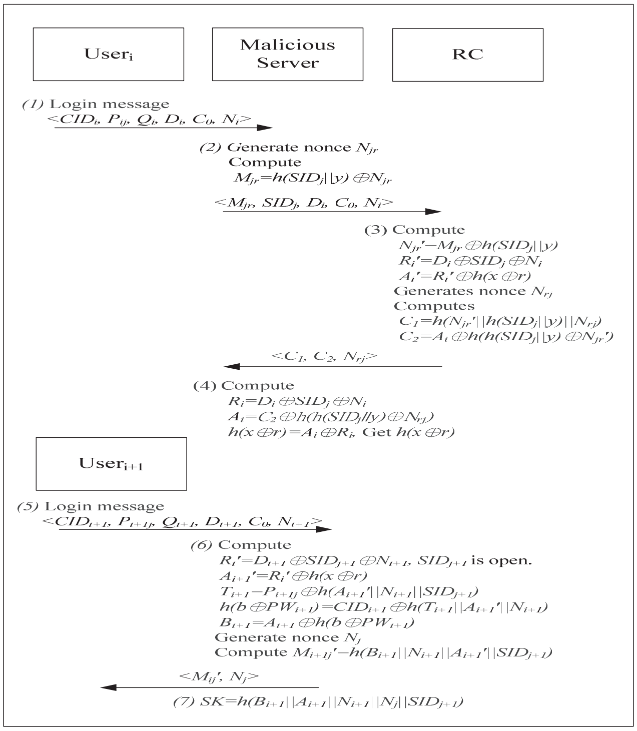
User$_i$
Malicious Server
RC
(1) Login message
$<CID_i, P_{ij}, Q_i, D_i, C_0, N_i>$
(2) Generate nonce N_{jr}
Compute
$M_{jr} = h(SID_j||y) \oplus N_{jr}$
$<M_{jr}, SID_j, D_i, C_0, N_i>$
(3) Compute
$N_{jr}' - M_{jr} \oplus h(SID_j||y)$
$R_i' = D_i \oplus SID_j \oplus N_i$
$A_i' = R_i' \oplus h(x \oplus r)$
Generates nonce N_{rj}
Computes
$C_1 = h(N_{jr}'||h(SID_j||y)||N_{rj})$
$C_2 = A_i \oplus h(h(SID_j||y) \oplus N_{jr}')$
$<C_1, C_2, N_{rj}>$
(4) Compute
$R_i = D_i \oplus SID_j \oplus N_i$
$A_i = C_2 \oplus h(h(SID_j//y) \oplus N_{rj})$
$h(x \oplus r) = A_i \oplus R_i$, Get $h(x \oplus r)$
User$_{i+1}$
(5) Login message
$<CID_{i+1}, P_{i+1j}, Q_{i+1}, D_{i+1}, C_0, N_{i+1}>$
(6) Compute
$R_i' = D_{i+1} \oplus SID_{j+1} \oplus N_{i+1}$, SID_{j+1} is open.
$A_{i+1}' = R_i' \oplus h(x \oplus r)$
$T_{i+1} - P_{i+1j} \oplus h(A_{i+1}'||N_{i+1}||SID_{j+1})$
$h(b \oplus PW_{i+1}) = CID_{i+1} \oplus h(T_{i+1}||A_{i+1}'||N_{i+1})$
$B_{i+1} = A_{i+1} \oplus h(b \oplus PW_{i+1})$
Generate nonce N_j
Compute $M_{i+1j}' - h(B_{i+1}||N_{i+1}||A_{i+1}'||SID_{j+1})$
$<M_{ij}', N_j>$
(7) $SK = h(B_{i+1}||A_{i+1}||N_{i+1}||N_j||SID_{j+1})$

Fig. 4 Server spoofing attack by malicious server on Hsiang-Shih's scheme.

The malicious user conducts server spoofing attack with the following steps that are illustrated in Fig. 3:

Step 1: The malicious user performs power analysis attack [12] to get B_i and R_i.

Step 2: The $h(b \oplus PW_i)$ is known at the step R1 of registration phase. $A_i = B_i \oplus h(b \oplus PW_i)$ and $h(x \oplus r) = A_i \oplus R_i$ are computed to obtain $h(x \oplus r)$.

The malicious server conducts server spoofing attack with the following steps that are illustrated in Fig. 4:

Step 1: The remote server receives message $<CID_i, P_{ij}, Q_i, D_i, C_0, N_i>$.
Step 2: S_j generates nonce N_{jr}, computes $M_{jr} = h(SID_j\|y) \oplus N_{jr}$, and sends the message $<M_{jr}, SID_j, D_i, C_0, N_i>$ to RC.
Step 3: RC computes $N_{jr}' = M_{jr} \oplus h(SID_j\|y)$, $R_i' = D_i \oplus SID_j \oplus N_i, A_i' = R_i' \oplus h(x \oplus r)$, $C_1 = h(N_{jr}'\|h(SID_j\|y)\|N_{rj})$ and $C_2 = A_i \oplus h(h(SID_j\|y) \oplus N_{jr}')$, and then sends the message $<C_1, C_2, N_{rj}>$ back to S_j.
Step 4: The malicious server computes $R_i = D_i \oplus SID_j \oplus N_i A_i = C_2 \oplus h(h(SID_j\|y) \oplus N_{rj})$ and $h(x \oplus r) = A_i \oplus R_i$ to get $h(x \oplus r)$.

- Invalid anonymity: The computation of CID_i, P_{ij}, Q_i and nonce is used to achieve the goal of anonymity. However, the malicious server can acquire T_i by computing $T_i = P_{ij} \oplus h(A_i\|N_i\|SID_j))$ to trace the behavior of a user because $T_i = h(ID_i\|x)$ and $A_i = h(h(b \oplus PW_i\|r) \oplus h(x \oplus r)$ are two fixed values that may leak user identity information.

4. The Proposed Scheme

A novel privacy-preserving dynamic ID-based remote user authentication scheme with access control for multi-server environment is proposed to solve those security problems. In which, Lagrange's interpolating polynomial to compute a curve polynomial $F(L)$ [9] is used for the purpose of access control. The assumption of server codes and role values for access control is (1, 3) (2, 11) (3, 17) to compute Lagrange's curve polynomial $F(L) = -x^2 + 11x - 7$. The proposed scheme consists of five phases: the registration phase, login phase, authentication phase and password change phase, and track phase.

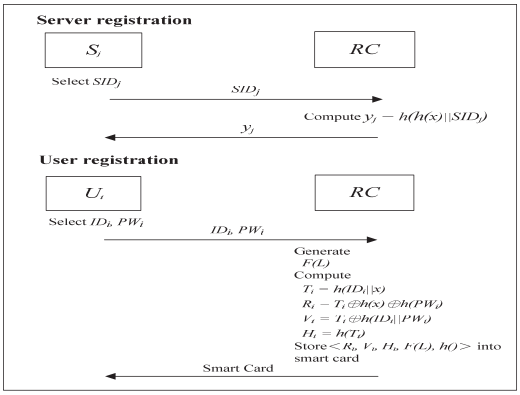

Fig. 5 Registration phase.

A. Registration phase

The registration phase illustrated in Fig. 5 deals with server registration and user registration.

- Server registration: Before service operation, the service provider S_j submits his/her SID_j to the trusted RC via secure channel. RC computes a secret number $y_j = h(h(x)\|SID_j)$ and sends it to S_j via secure channel.
- User registration: Before a user U_i can access services, U_i has to submit ID_i and PW_i to RC via secure channel. RC generates the curve polynomial $F(L)$ as access rights of U_i and computes $T_i = h(ID_i\|x)$, $R_i = T_i \oplus h(x) \oplus h(PW_i)$, $V_i = T_i \oplus h(ID_i\|PW_i)$ and $H_i = h(T_i)$. Then, RC stores $<R_i, V_i, H_i, F(L), h()>$ into a smart card and issues it to U_i via secure channel.

B. Login phase

When U_i wants to access S_j, U_i inserts smart card into the card reader and inputs ID_i^* and PW_i^* for verifying the legality of a cardholder. Then, the smart card performs the following steps that are shown in Fig. 6:

Step 1: Compute $T_i^* = V_i \oplus h(ID_i^*\|PW_i^*)$ and check whether H_i and $h(T_i^*)$ is equal or not. If the verification holds, the legitimacy of U_i can be assured and proceeds to the next step; otherwise, reject the login request.
Step 2: Generate the nonce N_i and compute secret number $y_j^* = h(R_i \oplus T_i^* \oplus h(PW_i^*)\|SID_j^*)$, $CID_i = ID_i^* \oplus h(R_i \oplus T_i^* \oplus h(PW_i*)\|N_i)$, and $Q = h(T_i^*\|N_i)$.
Step 3: Use the identity SID_j of S_j into polynomial $F(L)$ to get the role value P and compute $P_L = P \oplus h(y_j^*\|N_i)$.
Step 4: Compute $G_i = CID \oplus h(y_j^*\|N_i)$ and $C = h(CID_i\|Q\|P\|N_i)$.
Step 5: Send the login request message $<C, G_i, Q, P_L, N_i>$ to the service provider S_j.

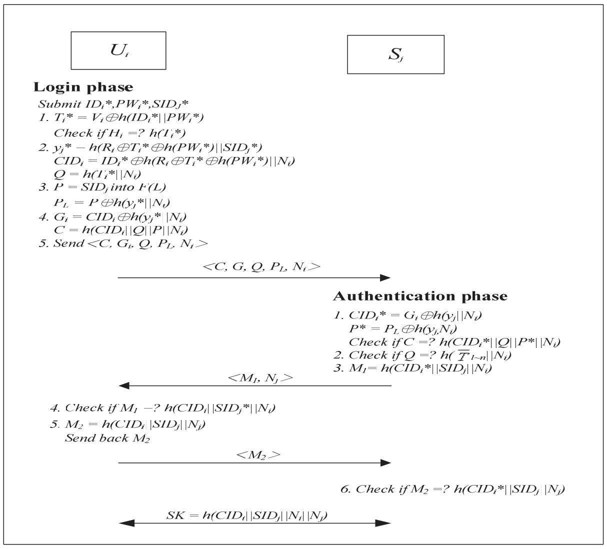
U_i
S_j
Login phase
Submit ID_i*,PW_i*,SID_j*
1. $T_i^* = V_i \oplus h(ID_i^*\|PW_i^*)$
Check if $H_i =?\ h(T_i^*)$
2. $y_j^* - h(R_i \oplus T_i^* \oplus h(PW_i^*)\|SID_j^*)$
$CID_i = ID_i^* \oplus h(R_i \oplus T_i^* \oplus h(PW_i^*)\|N_i)$
$Q = h(T_i^*\|N_i)$
3. $P = SID_j$ into $F(L)$
$P_L = P \oplus h(y_j^*\|N_i)$
4. $G_i = CID_i \oplus h(y_j^*\|N_i)$
$C = h(CID_i\|Q\|P\|N_i)$
5. Send $<C, G_i, Q, P_L, N_i>$
$<C, G, Q, P_L, N_i>$
Authentication phase
1. $CID_i^* = G_i \oplus h(y_j\|N_i)$
$P^* = P_L \oplus h(y_j,N_i)$
Check if $C =?\ h(CID_i^*\|Q\|P^*\|N_i)$
2. Check if $Q =?\ h(\overline{T}_{l\sim n}\|N_i)$
3. $M_1 = h(CID_i^*\|SID_j\|N_i)$
$<M_1, N_j>$
4. Check if $M_1 -?\ h(CID_i\|SID_j^*\|N_i)$
5. $M_2 = h(CID_i\|SID_j\|N_j)$
Send back M_2
$<M_2>$
6. Check if $M_2 =?\ h(CID_i^*\|SID_j\|N_j)$
$SK = h(CID_i\|SID_j\|N_i\|N_j)$

Fig. 6 Login and authentication phase.

C. Authentication phase

When receiving the login request message $<C, G_i, Q, P_L, N_i>$, S_j authenticates the user U_i with the following steps that are shown in Fig. 6:

Step 1: Compute $CID_i^* = G_i \oplus h(y_j||N_i)$ and $P^* = P_L \oplus h(y_j, N_i)$ and check whether C and $h(CID_i^*||Q||P^*||N_i)$ is equal or not. If the verification doesn't hold, S_j rejects the login request and terminates the session.

Step 2: Verify Q with every data $h(\overline{T_i}||N_i)$ stored in the blacklist of malicious users given by RC at Track phase. When a match is found, S_j rejects the login request.

Step 3: Generate the nonce N_j, compute $M_1 = h(CID^*||SID_j||N_i)$, and then send back the message $<M_1, N_j>$ to U_i.

Upon receiving the acknowledgement message, U_i performs the following steps:

Step 4: Compute $h(CID_i||SID_j^*||N_i)$ and compare it with M_1. If they are equivalent, it indicates that the legality of S_j. Otherwise, this session is interrupted.

Step 5: Compute $M_2 = h(CID_i||SID_j||N_j)$ and send back M_2 to S_j.

After receiving the response M_2 from U_i, S_j conducts the last step at authentication phase.

Step 6: Check whether M_2 and $h(CID_i^*||SID_j||N_j)$ is equal or not and abort the session if the verification doesn't hold.

After finishing authentication phase, the user U_i and the service provider S_j compute $h(CID_i||SID_j||N_i||N_j)$ as the session key SK.

D. Password Change Phase

The user U_i can update his/her password without the help of RC. U_i inserts the smart card to card reader and input (ID_i^*, PW_i^*) corresponding to the smart card. Then, the smart card performs the following steps:

Step 1: Compute $T_i^* = V_i \oplus h(ID_i^*||PW_i^*)$ and check whether H_i and $h(T_i^*)$ is equal or not. If not, reject the password change request; otherwise, choose a new password PW^{new}.

Step 2: Compute $V_i^{new} = T_i^* \oplus h(ID_i * ||PW^{new})$ and $R_i^{new} = R_i \oplus h(PW_i^*) \oplus h(PW^{new})$. The parameter V_i^{new} and R_i^{new} is stored in the smart card to replace V_i and R_i, respectively.

E. Track phase

On discovering a malicious user, the service provider S_j collects the relevant data regarding the user U_i and obtains U_i's real identity with the cooperation of RC.

Step 1: The server S_j sends CID_i and N_i to RC.

Step 2: When receiving the track request message (CID_i, N_i), RC computes $ID_i = CID_i \oplus h(h(x)||N_i)$ and $\overline{T_i} = h(ID_i||x)$.

Step 3: RC updates the blacklist of malicious users with $\overline{T_i}$ and sends the latest version of the blacklist to all of servers in order to trace and reject a malicious user at once.

5. Discussions

5.1 Security Analysis

A security analysis on the proposed scheme is given in accordance with the list of requirements discussed in academic literature.

- Replay attack: Replay attack is one of active attacks that an adversary records a communication session and replays the entire session, or a portion thereof, at some later point in time. The use of nonce N_i and N_j by demonstrating knowledge of a secret known to be associated with the involved parties S_j and U_i in login and authentication phases to compute the related data, including of C, Q, G_i, A_L, M_1 and M_2. Obviously, our scheme is effective against such an attack.
- Parallel session attack: A parallel session attack occurs when two or more protocol runs are executed concurrently and messages from one run (the reference session) are used to form spoofed messages in another run (the attack session). In which, it can also successfully impersonate a legal user, however, without having to hazard the possibility of exposing the attacker's identity. In view of this, the secret number $y_j^* = h(h(x)||SID_j^*)$ in our scheme is computed by the smart card and only the designate server can verify the legitimacy of the user.
- Stolen verified attack: Our scheme does not maintain any verification table and thus can achieve stolen-verified attack resistance.
- Guess password attack: In our scheme, the password is encapsulated in the message $<C, G_i, Q, P_L, N_i>$ by the use of the hash operation and the exclusive-or operation during login and authentication phase. Hence, it is effective against more common methods of password cracking.
- Server spoofing attack: The unique secret key y_j is the key to obtain the required data CID_i from G_i and compute the authentication message $M_1 = h(CID_i||SID_j||N_j)$. The inability of malicious users or servers to masquerade as a valid service provider is assured because of a lack of y_j only known by the authorized S_j.
- Impersonation attack: The adversary cannot masquerade as a legal user to login the remote server S_j even though he may get R_i, V_i, H_i, $F(L)$ and $h(\cdot)$ from the user's smart card. This is because the secret key x, y_j and U_i's password PW_i cannot be derived from these data and thus a valid login message cannot be generated. Besides, a legal user U_i cannot masquerade as the other legal user U_j to login the remote server S_j without the password PW_j.

5.2 Performance and Functionality Analysis

We then summarize some features related to practicability and effectiveness in this section.

- Freely chosen password: In the beginning of the password change phase, the validity of the cardholder will be confirmed. The user U_i inserts his smart card into a card reader and enters his identity ID_i^* and password PW_i^* corresponding to his smart card. Then, the legal cardholder can freely choose and change password at will without the help of RC.
- Session key agreement: In consideration of the security of session key generation, three security criteria are crucial for session key agreement [1], [4], [7].

  (1) Known key security: Only knowing a compromised session key cannot determine the other used session keys. In our scheme, the session key SK_i associated to CID_i and T_i is dynamic at every session to conduct mutual authentication with the user and the server. Even though a session key SK_i is disclosed, it is infeasible to compute SK_j due to insufficient data CID_i and T_i based on the security of one-way hash function and exclusive-or operation.

  (2) Session key security: At the end of the key agreement, the session key is known to nobody but the user and the server. In our scheme, the session key $SK = h(CID_i \| SID_j \| N_i \| N_j)$ is known to nobody but S_j and U_i since the secrecy of CID_i is concealed in the parameter G_i.

  (3) Forward secrecy: A compromised secret key cannot derive the session keys used before. In our scheme, a compromised long-lived secret key x cannot be used to derive the session key $SK = h(CID_i \| SID_j \| N_i \| N_j)$ used before without knowing the used nonce values N_i and N_j.

- Single registration: A new user just needs to register at RC once that is issued a smart card with the necessary secret information. Then, the user can login all the legal servers of the remote system by the use of the smart card and his/her password.
- User anonymity: The user U_i will send the login request (C, G_i, Q, P_L, N_i) to the server S_j in each login. Due to nonce N_i, the login message associated to the secret ID_i, however, is dynamic for running sessions to fulfill mutual authentication. The proposed scheme is accomplishment of anonymity goals.
- Traceability of malicious users: To make privacy protection more secure, the protection of user anonymity from remote servers is achieved in our scheme. In which, we deal with the balance of anonymity versus accountability known as accountable anonymity. That is, the server has the ability to identify malicious users from the blacklist $\overline{T}_i = h(ID_i \oplus x)$ issued by RC in the track phase.
- Comparison of functionality: Table 2 summarizes the functionality discussion given above and gives the comparison results among our scheme and the related schemes. Unfortunately, most recent studies only provide the countermeasure against outside attackers and thus they cannot provide user privacy as claimed. The proposed scheme provides the protection of user anonymity from inside attackers and further, against conspiracy attack. No one else but the trusted RC knows the real identity of a user. On discovering malicious behaviors of a user, all of servers have the blacklist of malicious users given by RC and reject the service request when a mach is found.

Table 2 Comparison of functionality.

	CP	MA	SKA	SR	SUK	UAO	UAS	TR
Juang [7]	Yes	Yes	Yes	Yes	Yes	No	No	No
Hu et al. [4]	Yes	No*	Yes	Yes	No	No	No	No
Liao&Wang [11]	Yes	No*	Yes	Yes	No	Yes	No	No
Hsiang&Shih [6]	Yes	No*	Yes	Yes	Yes	Yes	No	No
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

CP: securely and freely to change password; MA: mutual authentication; SKA: session key agreement; SUK: unique secret number among servers; SR: single registration; UAO: user anonymity against outside attackers; UAS: user anonymity against remote servers; TR: traceability of malicious user; *: fails to protect the mutual authentication.

Table 3 Comparison of performance.

	ERU	**ERS**	**EL**	**EA**
Juang [7]	$(2t+1)*T_S$, $T_\oplus$	$t*T_h$	$2T_h$, T_S, $T_\oplus$	$3T_h$, $6T_S$
Hu et al. [4]	$2T_h$	*none*	$2T_h$, T_S	$10T_h$, T_S
Liao&Wang [11]	$5T_h$, $2T_\oplus$	*none*	$6T_h$, $3T_\oplus$	$9T_h$, $3T_\oplus$
Hsiang&Shih [6]	$7T_h$, $5T_\oplus$	T_h	$7T_h$, $7T_\oplus$	$15T_h$, $12T_\oplus$
Our scheme	$3T_h$	$5T_h$, $3T_\oplus$	$7T_h$, $7T_\oplus$	$7T_h$, $T_\oplus$

ERU: the needed computation cost for user registration; ERS: the needed computation cost for server registration; EL: the needed computation cost of the login phase; EA: the needed computation cost of the authentication and key agreement phase; T_h: the computation operation of the hash function; T_S: the symmetric encryption/decryption operation; $T_\oplus$: the exclusive-or operation; t: the number of remote servers.

- Comparison of performance: Table 3 shows the comparison results in the computation cost. The computation cost of the XOR operation is smaller than the hash function; and further, the computation cost of the hash function is smaller than the symmetric encryption/decryption. It is obvious that the proposed scheme is more efficient than the others.

6. Conclusion

Within the rapid growth of those e-commerce applications a remote user authentication with anonymity toward security and privacy concerns is very promising. Users will find it unacceptable that their daily online activities can be freely recorded, linked and traced back to their identities unconditionally. In this paper, we briefly reviewed Liao-Wang's and Hsiang-Shih's scheme that is effective against user's anonymity in most cases. However, we found that it cannot achieve true anonymity, i.e., in case of insider attacks. We further proposed the improvements to avoid such an attack. The key features of our scheme are including of no verification table, freely chosen password, mutual authentication, low computation and communication cost, single registration, and secure against the related attacks. Moreover, an integrated management environment for authentication and key agreement is required for practical applications and introduced in the proposed scheme.

Acknowledgments

The author would like to thank the anonymous reviewers for their valuable comments and suggestions. This paper was supported in part by the National Science Council, Taiwan, under contract NSC 100-2410-H-020-004.

References

[1] C.C. Chang and J.Y. Kuo, "An efficient multi-server password authenticated key agreement scheme using smart cards with access control," International Conference on Advanced Information Networking and Applications, vol.2, pp.257–260, 2005.

[2] H.Y. Chien and C.H. Chen, "A remote authentication scheme preserving user anonymity," International Conference on Advanced Information Networking and Applications, vol.2, pp.245–248, 2005.

[3] M.L. Das, A. Saxena, and V.P. Gulati, "A dynamic ID-based remote user authentication scheme," IEEE Trans. Consum. Electron., vol.50, no.2, pp.629–631, 2004.

[4] L. Hu, X. Niu, and Y. Yang, "An efficient multi-server password authenticated key agreement scheme using smart cards," International Conference on Multimedia and Ubiquitous Engineering, pp.903–907, 2007.

[5] M.S. Hwang and I.H. Li, "A new remote user authentication scheme using sinart cards," IEEE Trans. Consum. Electron., vol.46, no.I, pp.28–30, 2000.

[6] H.C. Hsiang and W.K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," Computer Standards & Interfaces, vol.31, pp.1118–1123, 2009.

[7] W.S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," IEEE Trans. Consum. Electron., vol.50, no.1, pp.251–255, 2004.

[8] S. Kim, H.S. Rhee, J.Y. Chun, and D.H. Lee, "Anonymous and traceable authentication scheme using smart cards," International Conference on Information Security and Assurance, pp.162–165, 2008.

[9] L.H. Li, L.C. Lin, and M.S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," IEEE Trans. Neural Netw., vol.12, no.6, pp.1498–1504, 2001.

[10] M. Lin, "A study of web based single sign-on with RBAC authorization mechanism," shih Hsin University, 2005.

[11] Y.P. Liao and S.S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," Computer Standards & Interfaces, pp.24–29, 2009.

[12] T.S. Messergers, E.A. Dabbish, and R.H. Sloan, "Examining smart card security under the threat of power analysis attacks," IEEE Trans. Comput., vol.51, no.5, pp.541–552, 2002.

[13] H.M. Sun, "An efficient remote use authentication scheme using smart cards," IEEE Trans. Consum. Electron., vol.46, no.4, pp.958–961, 2000.

[14] P. Kocher, J. Jaffe, and B. Jun, "Introduction to Differential Power Analysis and Related Attacks," 1998. Available at http://www.cryptography.com/dpa/technical

[15] R.C. Wang, W.S. Juang, and C.L. Lei, "User authentication scheme with privacy-preservation for multi-server environment," IEEE Commun. Lett., vol.13, pp.157–159, 2009.

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

[17] W.S. Juang, H.C. Tseng, and Y.Y. Shue, "An efficient and privacy protection multi-server authentication scheme for low-cost RFID tags," International Computer Symposium, pp.279–283, 2010.

[18] Q. Xie and D. Chen, "Hash function and smart card based multi-server authentication protocol," WASE International Conference on Information Engineering (ICIE), pp.17–19, 2010.

[19] T. Liu and H. Zhu, "An ID-based multi-server authentication with key agreement scheme without verification table on elliptic curve cryptosystem," International Conference on Computational Aspects of Social Networks (CASoN), pp.61–64, 2010.

[20] E.J. Yoon and K.-Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," J. Supercomputing, Nov. 2010.

[21] M.H. Shao and Y.C. Chin, "A novel dynamic ID-based remote user authentication and access control scheme for multi-server environment," The Third International Symposium on Trust, Security and Privacy for Emerging Applications (TSP2010), pp.1102–1107, Bradford, UK, 2010.

[22] D. Yang and B. Yang, "A provable security biometric password multi-server authentication scheme with smart card," The Second International Symposium on Data, Privacy and E-Commerce (ISDPE), pp.33–38, 2010.

Min-Hua Shao is an Associate Professor with the Department of Management Information Systems at National Pingtung University of Science & Technology, Taiwan. She received the Ph.D. degree in Information Management from National Chiao Tung University, Taiwan, in 2005. Her research interests include information security, cryptographic protocol, wireless mobile communication security, and trust and privacy issues over electronic and digital commerce.

Ying-Chih Chin received the M.S. degree in Department of Management Information Systems from National Pingtung University of Science and Technology, Pingtung, Taiwan, in 2010. His research interests include information security management system.