Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A lightweight authentication scheme with user untraceability*

Kuo-Hui YEH
(*Department of Information Management, National Dong Hwa University, Taiwan 974, Hualien*)
E-mail: khyeh@mail.ndhu.edu.tw

Received July 3, 2014; Revision accepted Nov. 13, 2014; Crosschecked Mar. 4, 2015

Abstract: With the rapid growth of electronic commerce and associated demands on variants of Internet based applications, application systems providing network resources and business services are in high demand around the world. To guarantee robust security and computational efficiency for service retrieval, a variety of authentication schemes have been proposed. However, most of these schemes have been found to be lacking when subject to a formal security analysis. Recently, Chang *et al.* (2014) introduced a formally provable secure authentication protocol with the property of user-untraceability. Unfortunately, based on our analysis, the proposed scheme fails to provide the property of user-untraceability as claimed, and is insecure against user impersonation attack, server counterfeit attack, and man-in-the-middle attack. In this paper, we demonstrate the details of these malicious attacks. A security enhanced authentication scheme is proposed to eliminate all identified weaknesses.

Key words: Authentication, Privacy, Security, Smart card, Untraceability
doi:10.1631/FITEE.1400232 **Document code:** A **CLC number:** TP309

1 Introduction

Following advances in network technologies and the widespread availability of remote system backup, many service applications have been developed to make legitimate user access network service more convenient and efficient. As a password-based authentication scheme provides an efficient and accurate way to identify valid remote users, and at the same time preserves secrecy of communication, many password-based authentication mechanisms have been investigated in recent years. However, due to the inherent trade-off between security robustness and computational complexity, designing an authentication scheme which simultaneously possesses system reliability and performance efficiency poses a difficult challenge. Since the first authentication protocol was proposed by Lamport (1981), the research community has focused considerable attention on this important research area. Liao and Wang (2009) developed a dynamic identity-based remote user authentication scheme. In their scheme, only lightweight cryptography modules, i.e., exclusive-or operation and hash function, are required to support mutual verification and session key agreement. In addition, the proposed scheme is based on two-factor security and a nonce-based mechanism. With an informal security analysis, the authors claimed that their scheme guaranteed computation efficiency and entity anonymity. In the same year, Hsiang and Shih (2009) demonstrated that Liao and Wang's scheme is insecure against insider attack, impersonation attack, and server spoofing attack, and cannot provide mutual authentication. Hsiang and Shih then introduced a remedy which is intended to repair the security vulnerabilities they discovered. They achieved the same level of computation efficiency by implementing a hash function and exclusive-or operation in the proposed scheme. Next, Sood *et al.* (2011) used a two-server paradigm design in which different levels of trust are assigned to the servers, and the user's

* Project supported by the Taiwan Information Security Center (TWISC) and the Ministry of Science and Technology, Taiwan (Nos. MOST 103-2221-E-259-016-MY2 and MOST 103-2221-E-011-090-MY2)
ORCID: Kuo-Hui YEH, http://orcid.org/0000-0003-0598-761X
© Zhejiang University and Springer-Verlag Berlin Heidelberg 2015

verifier information is distributed between a pair of servers, called the service provider server and the control server. As the control server contains all users' secret informaiton and is not directly accessible to the clients, it is less likely to be attacked. Nevertheless, the insecurity of the schemes proposed in Hsiang and Shih (2009) and Li *et al.* (2010) was proved by Yeh *et al.* (2011), He and Wu (2012), and Li *et al.* (2012), revealing that resistance to replay attack, impersonation attack, stolen smart card attack, and leak of verifier attack could not be provided. Chang and Lee (2012) presented a single-sign-on based authentication mechanism for distributed network environments. The concept of single sign-on can allow legal users to use a unitary token to access distributed service providers. A client-server architecture is assumed in the proposed scheme, and heavy exponential computation is adopted to deliver the strong security density of their protocol. Based on the proposed security arguments, their proposed mechanism seemed, prima facie, to be appropriately robust. However, Wang *et al.* (2013) pointed out that two types of attacks, i.e., user impersonation attack and credential recovering attack, can be invoked successfully against Chang and Lee's protocol. On the other hand, Juang *et al.* (2008) proposed a smart card based authenticated key agreement scheme. They provided a method to protect user identity during an authentication session. The security of Juang *et al.*'s mechanism is based on an elliptic curve cryptosystem and a symmetric cryptosystem. They claimed that the proposed scheme can achieve identity protection, session key agreement, resistance to insider attack, and low communication and computation cost via the elliptic curve cryptosystem. However, all these statements cannot be verified (Sun *et al.*, 2009; Li *et al.*, 2010). Later, Tsai *et al.* (2013) found that Li *et al.* (2010)'s scheme is vulnerable to de-synchronization attack. In addition, the secret update mechanism of Li *et al.* (2010)'s scheme is not well designed and the scalability of the registration table is thus not efficient. For these reasons, Tsai *et al.* (2013) demonstrated an anonymous authentication scheme. The distinguishing feature of Tsai *et al.*'s scheme is that the server does not need to maintain a registration table, which makes the scheme suitable for a large scale service level.

Wang (2012) conducted an interesting study to investigate the trust between smart cards and card readers; that is, the author wanted to examine the possibility of a user's information being compromised when an adversary possesses a stolen smart card with a compromised user password. Based on an adversary model consisting of three types of attackers, four important summary points were presented under the analyses of the robustness of four kinds of password based schemes against three attacker types: (1) a symmetric key based scheme is secure against the type I and II attackers, but not against a type III attacker; (2) a public key ID-based scheme (PSCAb) is secure against type I, II, and III attackers; (3) a public key HMQV-based scheme is secure against type I and II attackers, but not against the type III attacker; and (4) a public key based scheme with password validation data at server (PSCAV) is secure against type I, II, and III attackers. Later, Wang *et al.* (2012a) found that PSCAb has several practical pitfalls, and PSCAV is vulnerable in the type III security mode. In addition, they investigated numerous password-based authentication studies and presented 12 evaluation criteria for password based authentication schemes. Finally, a formally provable authentication scheme which satisfies the evaluation criteria was proposed. After that, Wang and Wang (2013) examined the security of two authentication schemes proposed by Hsieh and Leu (2012) and Wang (2012), and found that both schemes are vulnerable to offline dictionary attack under their assumption of the capabilities of the adversary model. In addition, they presented a comparative analysis of 'two-factor authentication schemes using smart cards' and 'common-memory-device-based two-factor schemes' under two self-defined adversary models. Huang *et al.* (2013) identified two specific security scenarios for smart card based password authentication in distributed systems, i.e., (1) adversaries with pre-computed data stored in the smart card, and (2) adversaries with different data (with respect to different time slots) stored in the smart card. Two attacks were shown to be practical via attack implementations on two authentication schemes, and corresponding countermeasures were proposed. Wang and Ma (2012) presented a five-phase authentication scheme including registration, login, verification, password change, and user

revoking phases. Unlike traditional password based authentication, a revoke phase was introduced to allow a user to revoke his/her stolen smart card. Then, Wang *et al.* (2014) investigated the possibility of designing an anonymous two-factor authentication scheme with the criteria from Madhusudhan Mittal's evaluation set. The authors found contradictions among the desired security properties. For example, the properties of 'local user password change' and 'resistance to smart card loss attack' are difficult to achieve simultaneously, while schemes without the local user password change property cannot provide the property of 'timely typo detection'. Later, Wang and Wang (2014) analyzed the trade-off between system efficiency and user anonymity, and presented an important finding: public-key techniques are intrinsically indispensable for a two-factor authentication scheme with user anonymity. Moreover, Wang *et al.* (2012b) demonstrated that a password-based user authentication scheme proposed by Li *et al.* (2011) cannot withstand offline password guessing attack and denial-of-service attack, and fails to provide user anonymity and forward secrecy. Wang *et al.* (2012b) further presented a robust scheme to overcome the identified drawbacks. Recently, Chang *et al.* (2014) proposed a smart card based authentication scheme to resist user traceability attack. The authors claimed that their scheme can withstand various attacks such as user impersonation attack, server counterfeit attack, replay attack, and password guessing attack. Unfortunately, we find that Chang *et al.* (2014)'s scheme is vulnerable to server counterfeit attack, user impersonation attack, and man-in-the-middle attack. In addition, this scheme cannot provide user-untraceability. All of these weaknesses will be presented in the following sections.

2 Review of Chang *et al.* (2014)'s scheme

In this section, we review the registration phase and the login and authentication phase of Chang *et al.* (2014)'s scheme. Note that, for clarity, the password change phase of Chang *et al.*'s authentication protocol is not mentioned here. In addition, we present the notations used throughout this paper in Table 1.

Registration phase:

Step 1: $U_i \rightarrow S$ (secure channel): ID_i, PW_i.

Table 1 Notations used in this paper

Parameter	Meaning
U_i	Legitimate user
S	Service provider
ID_i, PW_i	U_i's identity and password
CID_i	U_i's dynamic identity
T, T', T''	Timestamps
ΔT	Valid time interval
$h()$	A secure one-way hash function, such as SHA-2 (256 to 512 bits)
x	A secret key of S
y	A secret number of S
$\Vert$	Concatenate operation

Step 2: $S \rightarrow U_i$: a smart card containing parameters $\{N_i, y, h()\}$.

When a user U_i wants to access the service of service provider S, U_i chooses and sends his/her identity ID_i and password PW_i to S via a secure channel. After S receives the registration request, S computes $N_i=h(\mathrm{ID}_i \Vert x) \oplus h(\mathrm{PW}_i)$. Finally, S stores parameters $\{N_i, y, h()\}$ into U_i's smart card and issues this smart card to U_i securely.

Login and authentication phase (Fig. 1):

Step 1: U_i (with a smart card): compute $\mathrm{CID}_i=\mathrm{ID}_i \oplus h(N_i \Vert y \Vert T)$, $N_i'=N_i \oplus h(y \Vert T)$, $B=N_i \oplus h(\mathrm{PW}_i)=h(\mathrm{ID}_i \Vert x)$, and $C=h(N_i \Vert y \Vert B \Vert T)$.

Step 2: U_i (with a smart card)$\rightarrow S$: CID_i, N_i', C, T.

When U_i intends to access S, U_i inserts his/her smart card into a card reader, and inputs ID_i and PW_i. The smart card then computes $\mathrm{CID}_i=\mathrm{ID}_i \oplus h(N_i \Vert y \Vert T)$, $N_i'=N_i \oplus h(y \Vert T)$, $B=N_i \oplus h(\mathrm{PW}_i)=h(\mathrm{ID}_i \Vert x)$ and $C=h(N_i \Vert y \Vert B \Vert T)$. Next, the smart card sends $\{\mathrm{CID}_i, N_i', C, T\}$ to S through a common channel.

Step 3: S: check (1) $T'-T \le \Delta T$ and (2) if no login request with the same parameters $\{\mathrm{CID}_i, N_i', C, T\}$ is received at time from $T-\Delta T$ to $T+\Delta T$.

Step 4: S: compute $N_i^*=N_i' \oplus h(y \Vert T)$, $\mathrm{ID}_i^*=\mathrm{CID}_i \oplus h(N_i^* \Vert y \Vert T)$, $B^*=h(\mathrm{ID}_i^* \Vert x)$, $C^*=h(N_i^* \Vert y \Vert B^* \Vert T)$ and check the correctness of the received value C.

Step 5: $S \rightarrow U_i$: $a=h(B^* \Vert y \Vert T'')$, T''.

Step 6: U_i: compute $a^*=h(B \Vert y \Vert T'')$ and check the correctness of the received value a.

Once S gets $\{\mathrm{CID}_i, N_i', C, T\}$ at time T', S acts as follows:

1. S checks whether (1) $T'-T \le \Delta T$ and (2) if no login request with the same parameters $\{\mathrm{CID}_i, N_i', C, T\}$ is received at time from $T-\Delta T$ to $T+\Delta T$.

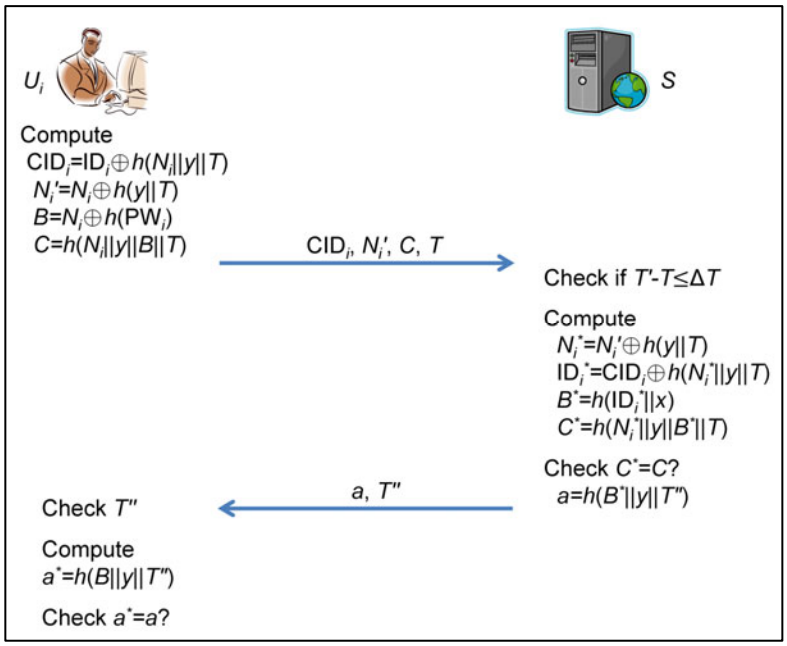

Fig. 1 Login and authentication phase of Chang *et al.* (2014)'s authentication scheme

If both conditions hold, this phase is passed. Otherwise, S aborts all login requests and immediately terminates this phase.

2. S computes $N_i^*=N_i'\oplus h(y\|T)$, $ID_i^*=CID_i\oplus h(N_i^*\|y\|T)$, $B^*=h(ID_i^*\|x)$, and $C^*=h(N_i^*\|y\|B^*\|T)$. After that, S checks if C^* equals C. If equal, U_i is successfully authenticated. Then, S computes $a=h(B^*\|y\|T'')$, where T'' is the current timestamp. Otherwise, S rejects U_i's login request and records ID_i^* and the number of cumulative failed requests for the resistance to replay attack. If three requests related to ID_i^* fail in a pre-defined interval, S will ignore U_i's following request within a guard interval.

3. S sends $\{a, T''\}$ to the smart card via a common channel.

Upon receiving $\{a, T''\}$ from S, the smart card checks the freshness of T''. If T'' is fresh in an expected time interval, the smart card computes $a^*=h(B\|y\|T'')$ and compares a^* with a. If values a^* and a are the same, U_i authenticates S.

3 Vulnerabilities of Chang *et al.* (2014)'s scheme

In this section, we demonstrate that Chang *et al.*'s scheme is insecure against user-traceability attack, user impersonation attack, server counterfeit attack, and man-in-the-middle attack.

3.1 User-traceability attack

Suppose there exists a legitimate but malicious user U_k with a smart card containing $\{N_k, y, h()\}$, where $N_k=h(ID_k\|x)\oplus h(PW_k)$. Once U_k intends to launch a user-traceability attack for a specific user U_i, N_k performs the following steps:

1. Eavesdrop on all messages, i.e., $\{CID_i, N_i', C, T\}$ and $\{a, T''\}$, transmitted between U_i and S in any given session. Note that all the messages are involved with U_i's secret parameters: $CID_i=ID_i\oplus h(N_i\|y\|T)$, $N_i'=N_i\oplus h(y\|T)$, $B=N_i\oplus h(PW_i)=h(ID_i\|x)$, $C=h(N_i\|y\|B\|T)$, $a^*=h(B^*\|y\|T'')$.

2. Since the secret number y of S is maintained in U_k's smart card, U_k can easily retrieve ID_i from $N_i=N_i'\oplus h(y\|T)$ and $ID_i=CID_i\oplus h(N_i\|y\|T)$, where N_i', T, and CID_i are public, and y can be retrieved from U_k's smart card. The derived procedure can be presented as follows:

(1) Compute $h(y\|T)$ with public value T, and secret y maintained in U_k's smart card.

(2) With computed $h(y\|T)$, N_i can easily be derived via $N_i'\oplus h(y\|T)$.

(3) Compute $h(N_i\|y\|T)$ with T, y, and computed value N_i.

(4) With computed $h(N_i \| y \| T)$ and public value CID_i, ID_i can easily be derived via $CID_i \oplus h(N_i \| y \| T)$.

Now, all requests sent by U_i will be connected to the retrieved ID_i. Hence, the user-traceability property cannot be guaranteed in Chang *et al.*'s scheme.

3.2 User impersonation attack

Suppose there exists a legitimate but malicious user U_k with a smart card containing $\{N_k, y, h()\}$, where $N_k=h(ID_k \| x) \oplus h(PW_k)$. Once U_k intends to launch a user impersonation attack for a specific user U_i, N_k performs the following steps:

1. Eavesdrop on $\{CID_i, N_i', C, T\}$ transmitted between U_i and S in any given session, where $CID_i=ID_i \oplus h(N_i \| y \| T)$, $N_i'=N_i \oplus h(y \| T)$, $B=N_i \oplus h(PW_i)=h(ID_i \| x)$, and $C=h(N_i \| y \| B \| T)$.

2. U_k retrieves y from his/her own smart card, and derives ID_i from $N_i=N_i' \oplus h(y \| T)$ and $ID_i=CID_i \oplus h(N_i \| y \| T)$, where N_i', T, and CID_i are public. As the server does not maintain a table to record all the registered users (and identities), now U_k can utilize this identity ID_i and a new password PW_k' to register as a new and legal user at S side. That is, U_k can obtain a new set of parameters, i.e., $\{N_k'=h(ID_i \| x) \oplus h(PW_k'), y, h()\}$, corresponding with ID_i and PW_k' from the registration phase at S side. After that, it is obvious that $h(ID_i \| x)$ can be retrieved via $N_k' \oplus h(PW_k')$ by the malicious user U_k. Note that the adversary can alternatively exploit an off-line password guessing attack to correctly guess the password PW_i, and then derive the $h(ID_i \| x)$ via $N_i \oplus h(PW_i)=h(ID_i \| x)$. Such password guessing based attack procedures for deducing $h(ID_i \| x)$ are also workable:

(1) U_k derives N_i and ID_i from the attack procedure described in Section 3.1.

(2) U_k uses this ID_i and a new password PW_k' to register as a new and legal user at S side, and gets back a set of parameters $\{N_k'=h(ID_i \| x) \oplus h(PW_k'), y, h()\}$.

(3) Derive $h(ID_i \| x)$ via $N_k' \oplus h(PW_k')$.

3. With the derived value $h(ID_i \| x)$, U_k can totally impersonate the user U_i with a counterfeit but legitimate request $\{CID_k, N_k', C_k, T_k\}$, where $CID_k=ID_i \oplus h(N_i \| y \| T_k)$, $N_i'=N_i \oplus h(y \| T_k)$, $B_k=h(ID_i \| x)$, and $C=h(N_i \| y \| B_k \| T_k)$. Note that ID_i and N_i are derived by U_k at the beginning of step 2, $B_k=h(ID_i \| x)$ is derived at the end of step 2, and T_k can be correctly derived with a series of eavesdropped timestamps.

Based on the foregoing, we can conclude that the resistance to user impersonation attack cannot be guaranteed in Chang *et al.*'s scheme.

3.3 Server counterfeit attack

Suppose there exists a legitimate but malicious user U_k with a smart card containing $\{N_k, y, h()\}$, where $N_k=h(ID_k \| x) \oplus h(PW_k)$. Once U_k intends to launch a server counterfeit attack for a specific user U_i, N_k performs the following steps:

1. Eavesdrop on $\{CID_i, N_i', C, T\}$ transmitted between U_i and S in any given session, where $CID_i=ID_i \oplus h(N_i \| y \| T)$, $N_i'=N_i \oplus h(y \| T)$, $B=N_i \oplus h(PW_i)=h(ID_i \| x)$, and $C=h(N_i \| y \| B \| T)$.

2. Similar to step 2 of the user impersonation attack described in Section 3.2, U_k retrieves y from his/her own smart card, and derives N_i and ID_i. Then, U_k can utilize a new registration at S side or password guessing based attack procedures to obtain $h(ID_i \| x)$.

3. With the derived value $h(ID_i \| x)$, U_k can totally cheat the user U_i into confounding U_k with a legal server with a valid response message $\{a_k, T_k''\}$, where $a_k=h(h(ID_i \| x) \| y \| T_k'')$. Note that T_k'' can be correctly derived after observing a series of transmitted timestamps.

In brief, the resistance to server counterfeit attack cannot be guaranteed in Chang *et al.*'s scheme.

3.4 Man-in-the-middle attack

Suppose there exists a legitimate but malicious user U_k with a smart card containing $\{N_k, y, h()\}$, where $N_k=h(ID_k \| x) \oplus h(PW_k)$. Once U_k intends to launch a man-in-the-middle attack for a specific user U_i, N_k performs the following steps:

1. Similar to the above attacks (e.g., user impersonation attack and server counterfeit attack), N_k first eavesdrops on $\{CID_i, N_i', C, T\}$ transmitted between U_i and S in a previous session, where $CID_i=ID_i \oplus h(N_i \| y \| T)$, $N_i'=N_i \oplus h(y \| T)$, $B=N_i \oplus h(PW_i)=h(ID_i \| x)$, and $C=h(N_i \| y \| B \| T)$.

2. U_k retrieves y from his/her own smart card, and derives N_i and ID_i. Then, U_k can utilize a new registration at S side or password guessing based attack procedures to obtain $h(ID_i \| x)$.

3. Once a new session is held between U_i and S, N_k acts as follows:

(1) Once U_i intends to send $\{CID_i, N_i', C, T\}$ to

S, U_k interrupts $\{\mathrm{CID}_i, N_i', C, T\}$ and impersonates U_i to issue a counterfeit but legitimate request $\{\mathrm{CID}_k, N_k', C_k, T_k\}$, where $\mathrm{CID}_k=\mathrm{ID}_i\oplus h(N_i\|y\|T_k)$, $N_k'=N_i\oplus h(y\|T_k)$, $B_k=h(\mathrm{ID}_i\|x)$, and $C_k=h(N_i\|y\|B_k\|T_k)$. Note that T_k is a valid timestamp chosen by U_k, and ID_i, N_i, and $h(\mathrm{ID}_i\|x)$ are derived.

(2) Once S gets $\{\mathrm{CID}_k, N_k', C_k, T_k\}$ at time T', S checks whether (1) $T'-T_k\le\Delta T$ and (2) if no login request with the same parameters $\{\mathrm{CID}_k, N_k', C_k, T_k\}$ is received at time from $T-\Delta T$ to $T+\Delta T$. Obviously, these two conditions hold.

(3) S computes $N_i^*=N_i'\oplus h(y\|T_k)$, $\mathrm{ID}_i^*=\mathrm{CID}_i\oplus h(N_i^*\|y\|T_k)$, $B^*=h(\mathrm{ID}_i^*\|x)$, and $C^*=h(N_i^*\|y\|B^*\|T_k)$, and U_k will be successfully authenticated as $C^*=C$. Then, S computes $a=h(B^*\|y\|T'')$, where T'' is the current timestamp. S sends $\{a, T''\}$ to the smart card via a common channel.

(4) After U_k receives $\{a, T''\}$, U_k pretends that he/she is the server S, and sends a valid response message $\{a_k, T_k''\}$ to U_i, where $a_k=h(h(\mathrm{ID}_i\|x)\|y\|T_k'')$.

(5) Upon receiving $\{a_k, T_k''\}$ from S (actually U_k), the smart card at U_i side checks the freshness of T'', and computes $a^*=h(B\|y\|T_k'')$ and compares a^* with a. Since all the values are valid, U_i authenticates S (actually U_k).

With the above attack procedures, we can conclude that a man-in-the-middle attack cannot be resisted in Chang *et al.*'s scheme.

4 The proposed scheme

In this section, we introduce a novel authentication scheme consisting of a registration phase, a log-in and authentication phase, and a password change phase. The newly proposed scheme guarantees security robustness without the weaknesses identified in the previous section. The assumptions of our scheme are: (1) We adopt a limited number of cumulative failed requests as a management policy for resisting the offline password guessing attack; (2) Once the user's smart card is lost (or stolen), the user will report the loss, and suspend the lost smart card online to avoid malicious manipulations of it; (3) Our scheme does not consider side-channel attacks.

Registration phase: When a user U_i wants to access the service of S, U_i chooses and sends his/her identity ID_i, password PW_i, and two chosen random numbers r_1 and r_2 to S via a secure channel. After S receives the registration request, S computes $M_i=h(y\|r_2)\oplus h(\mathrm{PW}_i\|r_1)$ and $N_i=h(\mathrm{ID}_i\|x)\oplus h(\mathrm{PW}_i\|r_1)$. Finally, S stores parameters $\{N_i, r_1, M_i, h()\}$ into U_i's smart card and issues this smart card to U_i securely. At the same time, S stores $h(h(y\|r_2))$ without any information connected to U_i. That is, S first deletes the registration information related to U_i, and then maintains the secret value $h(h(y\|r_2))$ as a random number in a pre-defined table T; hence, S cannot recognize U_i via $h(h(y\|r_2))$ or any other information from this table.

Step 1: $U_i\rightarrow S$ (secure channel): ID_i, PW_i, r_1, r_2.

Step 2: $S\rightarrow U_i$: a smart card containing parameters $\{N_i, r_1, M_i, h()\}$.

Login and authentication phase (Fig. 2):

Step 1: U_i (with a smart card): compute $A=M_i\oplus h(\mathrm{PW}_i\|r_1)=h(y\|r_2)$, $D=r_3\oplus h(A)$, $B=N_i\oplus h(\mathrm{PW}_i\|r_1)=h(\mathrm{ID}_i\|x)$, $N_i'=N_i\oplus h(h(A\|r_3))$, $\mathrm{CID}_i=\mathrm{ID}_i\oplus h(N_i\|h(A)\|r_3)$, $C=h(N_i\|h(A)\|B\|r_3)$.

Step 2: U_i (with a smart card)$\rightarrow S$: D, CID_i, N_i', C.

When U_i intends to access S, U_i inserts his/her smart card into a card reader, and inputs ID_i and PW_i. The smart card then generates a robust strongly-random number r_3, and computes $A=M_i\oplus h(\mathrm{PW}_i\|r_1)=h(y\|r_2)$, $D=r_3\oplus h(A)$, $B=N_i\oplus h(\mathrm{PW}_i\|r_1)=h(\mathrm{ID}_i\|x)$, $N_i'=N_i\oplus h(h(A)\|r_3)$, $\mathrm{CID}_i=\mathrm{ID}_i\oplus h(N_i\|h(A)\|r_3)$, and $C=h(N_i\|h(A)\|B\|r_3)$. Next, the smart card sends $\{D, \mathrm{CID}_i, N_i', C\}$ to S through a public channel.

Step 3: S: compute $r_3^*=D\oplus h(h(y\|r_2))^*$, $N_i^*=N_i'\oplus h(h(h(y\|r_2))^*\|r_3^*)$, $\mathrm{ID}_i^*=\mathrm{CID}_i\oplus h(N_i^*\|h(h(y\|r_2))^*\|r_3^*)$, $B^*=h(\mathrm{ID}_i^*\|x)$, $C^*=h(N_i^*\|h(h(y\|r_2))^*\|B^*\|r_3^*)$, and check the correctness of the received value C.

Step 4: S: examines (1) the freshness of r_3^*, and (2) if no login request with the same parameters $\{D, \mathrm{CID}_i, N_i', C\}$ is received.

Step 5: $S\rightarrow U_i$: $a=h(B^*\|h(h(y\|r_2))^*\|r_4)$, r_4.

Step 6: U_i: compute $a^*=h(B\|h(A)\|r_4)$ and check the correctness of the received value a.

Once S gets $\{D, \mathrm{CID}_i, N_i', C\}$, S iteratively retrieves each value $h(h(y\|r_2))^*$ from the maintained table T, and computes $r_3^*=D\oplus h(h(y\|r_2))^*$, $N_i^*=N_i'\oplus h(h(h(y\|r_2))^*\|r_3^*)$, $\mathrm{ID}_i^*=\mathrm{CID}_i\oplus h(N_i^*\|h(h(y\|r_2))^*\|r_3^*)$, $B^*=h(\mathrm{ID}_i^*\|x)$, and $C^*=h(N_i^*\|h(h(y\|r_2))^*\|B^*\|r_3^*)$. After that, S checks if C^* equals C. If equal, U_i is successfully authenticated. Next, S examines (1) the freshness of r_3^*, and (2) if no login request with

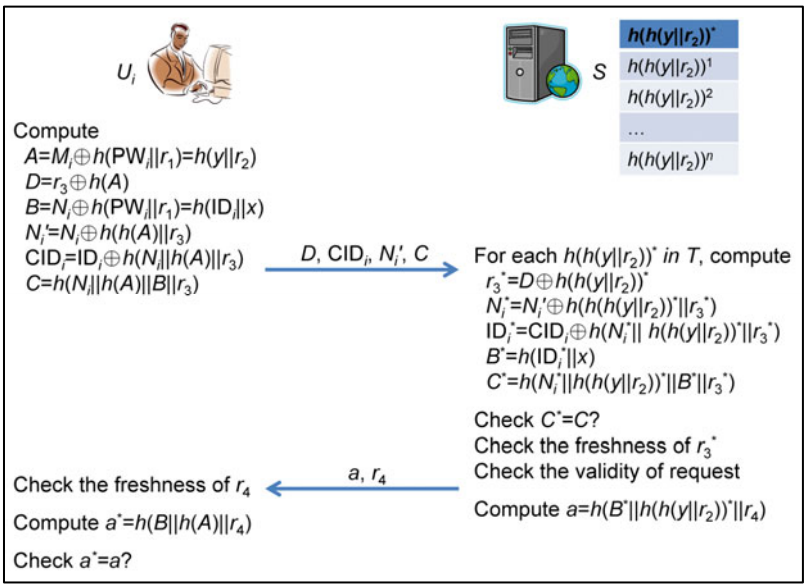

Fig. 2 Login and authentication phase of our proposed authentication scheme

the same parameters $\{D, CID_i, N_i', C\}$ is received within a predefined time interval. Once both conditions hold, this phase is passed. Otherwise, S immediately terminates this phase. Note that a threshold value of the number of cumulative failed requests can be set to 3. Then, S generates a random number r_4 and computes $a=h(B^* \| h(h(y \| r_2))^* \| r_4)$. Finally, S sends $\{a, r_4\}$ to the smart card via a common channel.

Upon receiving $\{a, r_4\}$ from S, the smart card checks the freshness of r_4. If r_4 is fresh in an expected time interval, the smart card computes $a^*=h(B \| h(A) \| r_4)$ and compares a^* with a. If values a^* and a are the same, U_i authenticates S.

Password change phase: When U_i wants to change the password, U_i inserts the smart card into the card reader and keys in his/her identity ID_i, password PW_i, and a new password PW_{i_new}. Next, the smart card calculates values $M_{i\_new}=M_i\oplus h(PW_i \| r_1)\oplus h(PW_{i\_new} \| r_1)$ and $N_{i\_new}=N_i\oplus h(PW_i \| r_1) \oplus h(PW_{i\_new} \| r_1)$, and stores M_{i_new} and N_{i_new} in the smart card's memory. Note that a threshold value of the number of cumulative failed requests can be set to 3. If the number of failed requests exceeds 3, the smart card will be locked and only a specific user re-verification procedure can be used to unlock this smart card. This mechanism can be exploited to be secure against an offline password guessing attack as the number of instances of password guessing and testing is limited.

5 Formal analysis of our proposed authentication scheme

In this section, we present the formal analysis of our proposed authentication scheme based on Burrows *et al.* (1990), Bellare and Rogaway (1994), Blake-Wilson *et al.* (1997), Bellare *et al.* (2000), and Chang and Lee (2012).

5.1 Communication model

In the communication model, we assume that a user U_i intends to access a service provider S_j. For this goal, some concepts must be formally defined:

1. Protocol participants: there exist a set of users, called Client, and a set of service providers, called Server, in the protocol P in which the participant is either a user or a service provider. Each participant may possess several instances, called oracles, which are involved in distinctly concurrent executions of P. Here, $\prod_U^i$ is denoted as the instance i of a participant U.

2. Long-term secret keys: Each $S_j \in$ Server possesses secret values x and y as the long-term secret keys trusted by all U_i's $\in$ Client.

3. Acceptance and termination: There exist two states, $ACC\_\prod_U^i$ and $TERM\_\prod_U^i$, for oracle $\prod_U^i$. Normally, $ACC\_\prod_U^i$ is set to true when $\prod_U^i$ is able to make a valid authentication session with S_j.

Meanwhile, TERM$_\prod_U^i$ will be set to true when $\prod_U^i$ sends (or receives) the last message of the protocol, receives an unexpected message, or misses an expected message.

4. Session and partner identities: the session identity (sid) is used to represent each unique session. We define sid for oracles $\prod_{U_i}^i$ and $\prod_{S_j}^i$ in the execution of a protocol as sid$_\prod_{U_i}^i$ = sid$_\prod_{S_j}^i$ = {Flows$_{U_iS_j}$ | all flows that $\prod_{U_i}^i$ exchanges with $\prod_{S_j}^i$ in the execution of a protocol}. The partner identity pid is used to represent the participant with whom the oracle believes it has agreed via a valid authentication session. That is, pid$_\prod_{U_i}^i = S_j$ means that the oracle $\prod_{U_i}^i$ believes that it has just agreed via a valid authentication session with an oracle of participant S_j.

5.2 Adversary model

In this paper, we assume that the adversary is able to interact with the participants via oracle queries. The following major queries model the capabilities of the adversary:

Send($\prod_U^i$, m): This query sends a message m to an oracle $\prod_U^i$, and obtains the corresponding results. For instance, the adversary issues a Send($\prod_U^i$, start) to initialize the protocol, and thus obtains the initial flow that the initiator sends to the receiver.

Reveal($\prod_U^i$): This query returns the authenticated tokens of the oracle $\prod_U^i$.

Corrupt(U): This query returns the long-term secrets, such as the user's private password, of U.

Execute($\prod_{U_A}^i$, $\prod_{S_B}^j$): This query models passive attacks in which the adversary can obtain the messages exchanged during the honest execution of the protocol between two oracles $\prod_{U_A}^i$ and $\prod_{S_B}^j$.

Hash(m): The one-way hash function can be viewed as a random function with the appropriate range in the ideal hash model. The adversary can use this query to get the hash result. Note that, if m has never been queried before, it returns a truly random number r to the adversary and stores (r, m) in the hash table. Otherwise, it returns the previously generated result to the adversary.

Test($\prod_U^i$): This query models the security of the authentication session, i.e., whether the authenticated tokens can be distinguished from a random string or not. To answer this question, an unbiased coin b is flipped by the oracle $\prod_U^i$. When the adversary issues a single Test query to $\prod_U^i$, the adversary obtains either the real authenticated tokens if b=1 or a random string if b=0.

5.3 Security properties

This subsection describes the security required in the proposed authentication.

Freshness: An oracle $\prod_U^i$ is fresh if the following conditions hold:

(1) ACC$_\prod_U^i$ is set to true;

(2) No Corrupt query has been issued by the adversary before ACC$_\prod_U^i$ is set to true;

(3) Neither $\prod_U^i$ nor its partner has been issued a Reveal query.

In general, an authenticated token is fresh if, and only if, all oracles that participate in the current session are fresh.

Partnering: In the protocol P, two oracles $\prod_{U_i}^i$ and $\prod_{S_j}^j$ are partnered if the following conditions hold:

(1) Both ACC$_\prod_{U_i}^i$ and ACC$_\prod_{S_j}^j$ have been set to true;

(2) An authenticated token has been agreed via $\prod_{U_i}^i$ and $\prod_{S_j}^j$;

(3) sid$_\prod_{U_i}^i$ = sid$_\prod_{S_j}^j$;

(4) pid$_\prod_{U_i}^i = S_j$;

(5) pid$_\prod_{U_i}^i = U_i$.

Based on the above analysis, the security of each authentication session can thus be defined as follows:

Session-security (SS): The adversary tries to guess the hidden bit b involved in a Test query via a guess b'. We say that the adversary wins the game of breaking the SS of an authentication protocol P if the adversary issues Test queries to a fresh oracle $\prod_{U_i}^i$ and guesses the hidden bit b successfully. The probability that the adversary wins the game is Pr[b'=b].

In brief, the advantage of an adversary A in attacking protocol P can be defined as $\mathrm{Adv}_P^{\mathrm{SS}}(A)=\left|2\Pr[b'=b]-1\right|$. In brief, P is secure if $\mathrm{Adv}_P^{\mathrm{SS}}(A)$ is negligible.

5.4 Formal security analysis

In this subsection, we formally analyze the security of our proposed authentication protocol. We define T_A as the adversary's total running time, and q_{s} and q_{h} are the numbers of Send and Hash queries, respectively.

Theorem 1 Let A be an adversary attempting to break the SS of our proposed authentication protocol within a time bound T_A, with less than q_{s} Send queries with the communication entities, and asking q_{h} times Hash queries. Then,

$$\mathrm{Adv}_P^{\mathrm{SS}}(A)\le\frac{q_{\mathrm{s}}+q_{\mathrm{h}}+q_{\mathrm{h}}^2}{2^{l+1}}+\max\left(\frac{q_{\mathrm{h}}^2}{2^{l_1+1}},\frac{q_{\mathrm{h}}^2}{2^{l_2+1}}\right)+\frac{q_{\mathrm{s}}^2}{2^{l_3+1}},$$

where l, l_1, l_2, and l_3 are the bit-lengths of the output of the hash function, the server's two secrets, and the largest-size transmitted message in P, respectively.

Proof Let A be an adversary who intends to break the SS of protocol P within time T_A. We can construct an SS-attacker B from A to respond to all of A's queries. A sequence of game reductions is involved in the proof. We introduce a sequence of game reductions starting at the real game G_0.

Game G_0: This is the real attack game in the random oracle model. Several oracles are available for the adversary: all users and servers instances $\prod_{U_i}^i$ and $\prod_{S_j}^j$, and a public hash oracle, i.e., $h()$. For any game G_n, we define the event S_n as occurring if $b=b'$, where b is the binary bit involved in the Test query, and b' is the output of the adversary. By this definition, we have $\mathrm{Adv}_P^{\mathrm{SS}}(A)=\left|2\Pr[S_0]-1\right|$. In addition, if the adversary has not stopped playing the game after q_{s} Send queries last more than T_A, we terminate the game and choose a random bit b' as the output, where q_{s} and T_A are pre-defined upper bounds.

Game G_1: This game simulates the public hash oracle $h()$: $\{0,1\}^*\rightarrow\{0,1\}^l$ with hash list Λ_{h}. Note that all instances such as Send, Reveal, Corrupt, Execute, and Test queries can be simulated to imitate the behavior of real players. From this simulation, we know that this game is indistinguishable from a real attack unless the permutation properties of $h()$ do not hold. As a result, according to the birthday paradox, the probability of a collision occurring is at most $\left|\Pr[S_1-S_0]\right|\le q_{\mathrm{h}}^2/2^{l+1}$.

Game G_2: We avoid collisions amongst the hash queries asked by the adversary to the hashed values $h(\mathrm{ID}_i\|x)$ or $h(y\|r_2)$. Assume that the adversary maintains query list Λ_A. The adversary first chooses a random element $r\in\{0,1\}^k$, and checks if $(*,r)\in\Lambda_{\mathrm{h}}\cap\Lambda_A$ holds. If it holds, we abort this game. The games G_2 and G_1 are indistinguishable unless game G_2 aborts. Therefore, the game will be aborted with the probability bounded for $\left|\Pr[S_2-S_1]\right|\le(q_{\mathrm{s}}+q_{\mathrm{h}})/2^{l+1}$.

Game G_3: We avoid collisions amongst the hash queries asked by the adversary to the server's ephemeral secrets, i.e., x and y. Assume that no collision has been found by the adversary for the server's ephemeral secrets. Choose two random elements $r_1\in\{0,1\}^{l_1}$ and $r_2\in\{0,1\}^{l_2}$. If this query is directly asked by the adversary and $\{(*,r_1),(*,r_2)\}\in\Lambda_A$, then we abort the game. Note that Λ_A denotes the queried list of the adversary. The two games G_3 and G_2 are indistinguishable once the adversary causes the game to abort. Hence, we obtain

$$\left|\Pr[S_3-S_2]\right|\le\max\left(\frac{q_{\mathrm{h}}^2}{2^{l_1+1}},\frac{q_{\mathrm{h}}^2}{2^{l_2+1}}\right).$$

Game G_4: We modify the game so that the adversary may guess the correct authentic values $\{D, \mathrm{CID}_i, N_i', C\}$ and $\{a, r_4\}$ without hash queries. First, when A issues a Send query as a start command, B responds $\{D, \mathrm{CID}_i, N_i', C\}$ to A. Second, when A issues a Send query, B randomly chooses two integers c_1 and c_2 from $[1, q_{\mathrm{s}}]$. If $c_1\neq c_2$, B responds $\{a, r_4\}$ to A. Otherwise, B replaces $\{a, r_4\}$ with a random string RS, and responds RS to A. Finally, when A issues a Send query, B answers with a null string and then sets $\mathrm{ACC}\_\prod_{U_i}^i$, $\mathrm{ACC}\_\prod_{S_j}^j$, $\mathrm{TERM}\_\prod_{U_i}^i$, and $\mathrm{TERM}\_\prod_{S_j}^j$ to true. Thus, games G_4 and G_3 are indistinguishable, where the maximum bit-length between $\{D, \mathrm{CID}_i, N_i', C\}$ and $\{a, r_4\}$ is l_3. Games G_4 and G_3 are indistinguishable

unless game G_4 aborts. The probability of this game being aborted is at most $\left|\Pr[S_4 - S_3]\right| \le q_s^2 / 2^{l_3+1}$. □

Theorem 2 The proposed authentication protocol possesses mutual authentication.

Proof We prove mutual authentication for our protocol based on BAN logic (Burrows *et al.*, 1990). Basic constructs and logic postulates are defined as follows (Note that in this section the symbols P and Q range over principals, X and Y range over statements, and K ranges over encryption keys):

Constructs:

P believes X: The principal P believes that X is true.

P sees X: Someone has sent a message containing X to P, who can read and repeat X (possibly after doing some decryption).

P said X: P has actually sent a message including statement X in the current session of the protocol or before.

P controls X: P has jurisdiction over X; i.e., the principal P is an authority on X and this matter should be trusted.

fresh(X): X has not been sent before the current session of the protocol.

$P \overset{K}{\longleftrightarrow} Q$: The key K is shared between the principals P and Q.

$P \overset{X}{\longleftrightarrow} Q$: The formula X is a secret known only to P and Q. Only P and Q may use X to prove their identities to each other.

$\{X\}_K$: This symbol represents the formula X encrypted or protected under the key K.

Logical postulates:

Rule 1 (Message-meaning rules) If P believes $P \overset{K}{\longleftrightarrow} Q$ and P sees $\{X\}_K$, then we postulate P believes Q said X.

Rule 2 (Nonce-verification rule) If P believes fresh(X) and P believes Q said X, then we postulate P believes Q believes X.

Rule 3 (Jurisdiction rule) If P believes Q controls X and P believes Q believes X, then we postulate P believes X.

Rule 4

(1) If P sees (X, Y) then P sees X.

(2) If P believes $P \overset{X}{\longleftrightarrow} Q$ and P sees $\{X\}_K$, then P sees X.

Rule 5 If one part of a formula is fresh, then the entire formula must also be fresh. If P believes fresh(X), then P believes fresh(X, Y).

Before analyzing the authentication scheme, the assumptions are given as follows:

Assumption 1 U_i, S_j believe $U_i \overset{h(y\|r_2),h(\mathrm{ID}_i\|x)}{\longleftrightarrow} S_j$.

Assumption 2 U_i, S_j believe fresh(r_3), fresh(r_4).

Assumption 3 S_j believes U_i controls r_3.

Assumption 4 U_i believes S_j controls r_4.

Our proposed authentication scheme is realized as follows:

Step 1: $U_i \rightarrow S_j$: $\{D, \mathrm{CID}_i, N_i', C\}$.

Step 2: $S_j \rightarrow U_i$: $\{a, r_4\}$.

The formal analysis of mutual authentication is as follows:

(1) S_j sees $\{D, \mathrm{CID}_i, N_i', C\}$.

(2) S_j believes $U_i \overset{h(y\|r_2),h(\mathrm{ID}_i\|x)}{\longleftrightarrow} S_j$ (from Assumption 1).

(3) S_j believes U_i said $\{D, \mathrm{CID}_i, N_i', C\}$ ((1) & (2), inferred by Rule 1).

(4) S_j believes fresh(r_3) (from Assumption 2).

(5) S_j believes U_i believes $\{D, \mathrm{CID}_i, N_i', C\}$ ((3) & (4), inferred by Rule 2).

(6) S_j believes U_i controls $\{r_3\}$ (from Assumption 3).

(7) S_j believes $\{D, \mathrm{CID}_i, N_i', C\}$ ((5) & (6), inferred by Rule 3).

(8) U_i sees $\{a, r_4\}$.

(9) U_i believes $U_i \overset{h(y\|r_2),h(\mathrm{ID}_i\|x)}{\longleftrightarrow} S_j$ (from Assumption 1).

(10) U_i believes S_j said $\{a, r_4\}$ ((8) & (9), inferred by Rule 1).

(11) U_i believes fresh(r_4) (from Assumption 2).

(12) U_i believes S_j believes $\{a, r_4\}$ ((10) & (11), inferred by Rule 2).

(13) U_i believes S_j controls $\{r_4\}$ (from Assumption 4).

(14) U_i believes $\{a, r_4\}$ ((12) & (13), inferred by Rule 3).

The final results are as follows:

S_j believes U_i believes $\{D, \mathrm{CID}_i, N_i', C\}$ (from (5)).

S_j believes $\{D, \mathrm{CID}_i, N_i', C\}$ (from (7)).

U_i believes S_j believes $\{a, r_4\}$ (from (12)).

U_i believes $\{a, r_4\}$ (from (14)).

With the four results (5), (7), (12), and (14), the remote user U_i and the service provider S_j can be authenticated by each other. □

Claim 1 The proposed authentication scheme guarantees data security.

In our proposed authentication scheme, all transmitted messages $\{D, \mathrm{CID}_i, N_i', C\}$ and $\{a, r_4\}$ are well protected via high-entropy secrets x and y chosen by S. Without knowing the two secrets, attackers cannot obtain any useful information from transmitted ciphertexts. In addition, due to the irreversibility of the one-way hash function, it is difficult for attackers to derive any secrets such as random numbers and secret values. Therefore, data confidentiality can be ensured in our proposed authentication scheme.

Claim 2 The proposed authentication scheme guarantees user anonymity and resistance to replay attack.

In each session of the proposed authentication scheme, four random numbers r_1, r_2, r_3, and r_4 are utilized to randomize the messages transmitted between the user and the server. Without revealing the real identities in public, all the communicating entities need only to know whether the involved partners are legitimate or not. In more detailed terms, in our proposed authentication scheme all the identities are transmitted in cipher format instead of plaintext, and these identities will be randomized in each new session. As a result, our authentication scheme can guarantee the property of user anonymity and prevent user-traceability attack. On the other hand, in each session, we exploit random numbers, i.e., r_3 and r_4, to perform the computation of all transmitted messages. As these two random numbers are newly generated in each session, the resistance to replay attack is naturally embedded in our proposed authentication scheme. That is, owing to the freshness verification of r_3 and r_4, the replay attack can easily be detected and prevented.

Claim 3 The proposed authentication scheme guarantees resistance to man-in-the-middle based attacks such as server counterfeit attack, user impersonation attack, and man-in-the-middle attack.

An attacker may issue counterfeit messages to deceive the legal communication users or the server. However, without the knowledge of the two high-entropy secrets x and y, it is difficult for the attacker to compute legitimate request or response messages such as $\{D, \mathrm{CID}_i, N_i', C\}$ and $\{a, r_4\}$. Even if the attacker sends a previously eavesdropped message to a victim party, the verification of these old messages will fail. This is because the random numbers r_3 and r_4 will have already been used in a previous session. In addition, the verification procedures have been modified to help the communicating parties to prevent man-in-the-middle based attacks. That is, as the secret values $h(\mathrm{PW}_i \| r_1)$, $h(y \| r_2)$, and $h(\mathrm{ID}_i \| x)$ cannot be derived by an adversary, all the man-in-the-middle based attacks, such as user impersonation attack, server counterfeit attack, and man-in-the-middle attack, identified in the previous section, will not succeed.

Claim 4 The user can freely change his/her password.

In Chang *et al.*'s protocol, the password change phase always involves the server. This design impedes the property of user convenience in the authentication protocol operation. In the proposed protocol, we totally modify the password phase so that the user can freely change his/her password without the help of the server.

6 Security and performance comparison

To further investigate the advantages of our proposed authentication protocol, we compare the proposed scheme with three relevant authentication schemes (Tsai *et al.*, 2013; Chang *et al.*, 2014; Kumari and Khan, 2014) in terms of major security and efficiency features. From a robustness standpoint (Table 2), our proposed authentication scheme is superior to the other protocols by virtue of supporting all the major security features. It can be seen that our proposed authentication scheme possesses all the advantages and achieves the security requirements.

Performance evaluation is an important issue when designing a robust and efficient authentication scheme. This evaluation reflects the practicability of implementing the proposed authentication protocol in the real world. Hence, we also compare our proposed protocol with three of the most relevant proposals, i.e., Chang *et al.* (2014)'s scheme, Kumari and Khan (2014)'s scheme, and Tsai *et al.* (2013)'s scheme, in terms of protocol efficiency. The performance comparison among our proposed scheme and other schemes is listed in Table 3. The metrics are

hash function (HF), modular multiplication (MM), modular exponentiation (ME), elliptic curve cryptography (ECC) point multiplication (PM), XOR operation, and encryption/decryption (E/D). As the cost of performing one-way hash function and XOR operation is negligible in comparison with other heavy computation modules such as ME, PM, and E/D (Table 3), our proposed authentication scheme can be said to be efficient. It is obvious that our scheme can achieve the same order of computation complexity as Chang *et al.* (2014)'s scheme, and delivers better security robustness.

7 Conclusions

Designing a secure but lightweight authentication scheme is a particular challenge owing to the difficult trade-off between security requirements and computation efficiency. A recent pioneering study proposed by Chang *et al.* (2014) broke new ground in this interesting research area. However, their scheme still has room for improvement. In this paper, we have demonstrated that Chang *et al.*'s authentication scheme fails to provide adequate security, such as user-untraceability, and is subject to user

Table 2 Security comparison of our proposed protocol and other schemes

Performance	Proposed scheme	Chang *et al.*, 2014	Tsai *et al.*, 2013
Freedom to choose and change password	Yes	No	Yes
User anonymity	Yes	No	Yes
Mutual authentication	Yes	Yes	Yes
Resistance to user impersonation attack	Yes	No	No
Resistance to server counterfeit attack	Yes	No	Yes
Resistance to man-in-the middle attack	Yes	No	Yes
Resistance to replay attack	Yes	Yes	Yes

Table 3 Performance comparison of our proposed protocol and other schemes

Phase	Type of operation	Number of operations			
		Proposed scheme	Chang *et al.*, 2014	Kumari and Khan, 2014	Tsai *et al.*, 2013
Registration	HF	4	1	1	2
	MM	0	0	4	0
	ME	0	0	2	0
	PM	0	0	0	0
	XOR	2	1	3	1
	E/D	0	0	1	0
Login and authentication	HF	11	10	5	10
	MM	0	0	3	0
	ME	0	0	3	0
	PM	0	0	0	4
	XOR	8	5	4	6
	E/D	0	0	4	0
Password change	HF	4	12	1	2
	MM	0	0	2	0
	ME	0	0	2	0
	PM	0	0	0	0
	XOR	4	7	4	2
	E/D	0	0	0	0
Total		19 HF+14 XOR	23 HF+13 XOR	7 HF+9 MM+7 ME+ 11 XOR+5 E/D	14 HF+4 PM+ 9 XOR

HF: hash function; MM: modular multiplication; ME: modular exponentiation; PM: ECC point multiplication; E/D: encryption/decryption

impersonation attack, server counterfeit attack, and man-in-the-middle attack. A novel authentication protocol is introduced for security enhancement. Our formal analysis and performance comparison have established that the security robustness and computation efficiency of our proposed authentication protocol can be guaranteed. In brief, we believe that our proposed protocol is both practical and suitable for current service application architecture.

References

Bellare, M., Rogaway, P., 1994. Entity authentication and key distribution. *LNCS*, **773**:232-249.

Bellare, M., Pointcheval, D., Rogaway, P., 2000. Authenticated key exchange secure against dictionary attacks. Advances in Cryptology-EUROCRYPT, p.139-155.

Blake-Wilson, S., Johnson, D., Menezes, A., 1997. Key agreement protocols and their security analysis. 6th IMA Int. Conf. on Cryptography Coding, p.30-45.

Burrows, M., Abadi, M., Needham, R., 1990. A logic of authentication. *ACM Trans. Comput. Syst.*, **8**(1):18-36. [doi:10.1145/77648.77649]

Chang, C.C., Lee, C.Y., 2012. A secure single sign-on mechanism for distributed computer networks. *IEEE Trans. Ind. Electron.*, **59**(1):629-637. [doi:10.1109/TIE.2011.2130500]

Chang, Y.F., Tai, W.L., Chang, H.C., 2014. Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. *Int. J. Commun. Syst.*, **27**(11):3430-3440. [doi:10.1002/dac.2552]

He, D., Wu, S., 2012. Security flaws in a smart card based authentication scheme for multi-server environment. *Wirel. Pers. Commun.*, **70**(1):323-329. [doi:10.1007/s11277-012-0696-1]

Hsiang, C., Shih, W.K., 2009. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interf.*, **31**(6):1118-1123. [doi:10.1016/j.csi.2008.11.002]

Hsieh, W., Leu, J., 2012. Exploiting hash functions to intensify the remote user authentication scheme. *Comput. Secur.*, **31**(6):791-798. [doi:10.1016/j.cose.2012.06.001]

Huang, X., Chen, X., Li, J., *et al*., 2013. Further observations on smart-card-based password-authenticated key agreement in distributed systems. *IEEE Trans. Parall. Distr. Syst.*, **25**(7):1767-1775. [doi:10.1109/TPDS.2013.230]

Juang, W.S., Chen, S.T., Liaw, H.T., 2008. Robust and efficient password-authenticated key agreement using smart cards. *IEEE Trans. Ind. Electron.*, **55**(6):2551-2556. [doi:10.1109/TIE.2008.921677]

Kumari, S., Khan, M.K., 2014. Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme. *Int. J. Commun. Syst.*, **27**(12):3939-3955. [doi:10.1002/dac.2590]

Lamport, L., 1981. Password authentication with insecure communication. *Commun. ACM*, **24**(11):770-772. [doi:10.1145/358790.358797]

Li, C.T., Lee, C.C., Liu, C.J., *et al*., 2011. A robust remote user authentication scheme against smart card security breach. 25th Annual IFIP WG 11.3 Conf., p.231-238.

Li, X., Qiu, W., Zheng, D., *et al*., 2010. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Trans. Ind. Electron.*, **57**(2):793-800. [doi:10.1109/TIE.2009.2028351]

Li, X., Xiong, Y., Ma, J., *et al*., 2012. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *J. Network Comput. Appl.*, **35**(2):763-769. [doi:10.1016/j.jnca.2011. 11.009]

Liao, Y.P., Wang, S.S., 2009. A secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interf.*, **31**(1):24-29. [doi:10.1016/j.csi.2007.10.007]

Sood, S.K., Sarje, A.K., Singh, K., 2011. A secure dynamic identity based authentication protocol for multi-server architecture. *J. Network Comput. Appl.*, **34**(2):609-618. [doi:10.1016/j.jnca.2010.11.011]

Sun, D.Z., Huai, J.P., Sun, J.Z., *et al*., 2009. Improvements of Juang *et al.*'s password-authenticated key agreement scheme using smart cards. *IEEE Trans. Ind. Electron.*, **56**(6):2284-2291. [doi:10.1109/TIE.2009.2016508]

Tsai, J.L., Lo, N.W., Wu, T.C., 2013. Novel anonymous authentication scheme using smart cards. *IEEE Trans. Ind. Inform.*, **9**(4):2004-2013. [doi:10.1109/TII.2012.2230639]

Wang, D., Ma, C.G., 2012. Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards. *J. China Univ. Posts Telecommun.*, **19**(5):104-114. [doi:10.1016/S1005-8885(11)60307-5]

Wang, D., Wang, P., 2013. Offline dictionary attack on password authentication schemes using smart cards. 16th Information Security Conf., p.1-16.

Wang, D., Wang, P., 2014. On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions. *Comput. Networks*, **73**:41-57. [doi:10.1016/j.comnet.2014.07.010]

Wang, D., Ma, C., Wang, P., *et al*., 2012a. iPass: privacy preserving two-factor authentication scheme against smart card loss problem. Cryptology ePrint Archive, **439**:1-35.

Wang, D., Ma, C., Wang, P., 2012b. Secure password-based remote user authentication scheme with non-tamper resistant smart cards. 26th Annual IFIP Conf. on Data and Applications Security and Privacy, p.114-121.

Wang, D., He, D., Wang, P., *et al*., 2014. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. *IEEE Trans. Depend. Secure Comput.*, in press. [doi:10.1109/TDSC.2014.2355850]

Wang, G., Yu, J., Xie, Q., 2013. Security analysis of a single sign-on mechanism for distributed computer networks. *IEEE Trans. Ind. Inform.*, **9**(1):294-302. [doi:10.1109/TII.2012.2215877]

Wang, Y., 2012. Password protected smart card and memory stick authentication against off-line dictionary attacks. 27th IFIP TC 11 Information Security and Privacy Conf., p.489-500.

Yeh, K.H., Lo, N.W., Li, Y., 2011. Cryptanalysis of Hsiang-Shih's authentication scheme for multi-server architecture. *Int. J. Commun. Syst.*, **24**(7):829-836. [doi:10.1002/dac.1184]