INFORMATICA, 2008, Vol. 19, No. 2, 285–302

A Pairing-Based User Authentication Scheme for Wireless Clients with Smart Cards *

Yuh-Min TSENG, Tsu-Yang WU, Jui-Di WU
Department of Mathematics, National Changhua University of Education
Jin-De Campus, Chang-Hua City 500, Taiwan, R.O.C.
e-mail: ymtseng@cc.ncue.edu.tw

Received: May 2007

Abstract. With rapid growth of mobile wireless networks, handheld devices are popularly used by people and many mobile applications have been rapidly developed. Considering the limited computing capability of smart cards or mobile devices, the security scheme design suitable for these mobile devices is a nontrivial challenge. A user authentication scheme is a mechanism to authenticate a remote user over an open network. In 2006, Das *et al*. proposed an identity (ID)-based remote user authentication scheme with smart cards using bilinear pairings. Unfortunately, their scheme is insecure against forgery attack. Recently, Giri and Srivastava proposed an improved scheme to overcome the forgery attack. The computational cost required by the Giri–Srivastava scheme is expensive, especially for smart cards with limited computing capability. In addition, the Giri–Srivastava scheme is unable to be used for a multi-server environment. This paper presents an efficient and secure ID-based remote user authentication scheme using bilinear pairings. Based on the computational Diffie–Hellman assumption, we show that the proposed scheme is secure against existential forgery on adaptively chosen-message and ID attack in the random oracle model. As compared with the recently proposed pairing-based authentication schemes, our scheme has better performance in term of the computational cost and it is suitable for a multi-server environment in distributed networks. Performance analysis and experimental data of related pairing operations on smartcards are given to demonstrate that our scheme is well suited for mobile devices with limited computing capability.

Keywords: authentication, wireless client, identity-based, smart card, bilinear pairing.

1. Introduction

Due to rapid growth in popularity of the Internet and wireless communications, many wireless E-commerce and business applications provide rapid and convenient resource accessing services to users. Now, handheld devices are popularly used by people and many mobile applications have been rapidly developed. Considering the limited computing capability of smart cards or mobile devices, the security scheme design based on traditional public-key systems is a nontrivial challenge because most cryptographic algorithms require many expensive computations. If public-key based cryptographic schemes

*This research was partially supported by National Science Council, Taiwan, R.O.C., under contract no. NSC95-2221-E-018-010.

are designed for mobile users with handheld devices or smart cards, the computational cost on the user side is a critical issue for implementation because of their limited computing capability (Tseng, 2006; Tseng, 2007).

A user authentication scheme is a mechanism to authenticate remote users over an open network. If a user wants to access services of servers in distributed networks, they must be authenticated by the servers before accessing their services. Traditionally, the user has to submit his identity and password to the server, then the server may validate his identity and password with the ones in its password table. In 1981, Lamport (1981) proposed a password-based remote user authentication scheme using a one-way hash function. The server uses a verification table to replace the password table for withstanding the threat of revealing the password table. It ensures the secrecy of the passwords even the verification table is disclosed. The existence of the verification table still incurs the cost of maintaining the table and could suffer from dictionary attacks (Jan and Chen, 1998; Chien *et al*., 2002). Afterwards, several user authentication schemes without the verification table have been proposed (Jan and Chen, 1998; Chien *et al*., 2002; Ku and Chang, 2005; Liaw *et al*., 2006). These schemes have a common property that each user's secret key is generated by the server using a one-way hash function. These secret keys are hard to memorize for users. For solving this problem, smart cards are issued to users for storing these secret keys. When a user with the smart card wants to access the server, he submits a login message to the server, while the server must keep the system secret to verify the user's login message. This enable that these hash-based schemes are not suited for a multi-server environment. Public-key based user authentication approach (Hwang and Li, 2000; Awasthi *et al*., 2003) with smart cards is alternative to solve the problem in a multi-server environment, but it requires expensive exponentiation operations.

In 1984, identity (ID)-based public-key system was first introduced by Shamir (1984). ID-based public-key system may simplify certificate management as compared to traditional public-key systems. However, Shamir's ID-based public-key system still suffers from many implementing problems, especially computational complexity. Recently, Boneh and Franklin (2001; 2003) proposed a practical ID-based encryption system based on bilinear pairings. Bilinear pairings (such as Weil pairings and Tate pairings) defined on elliptic curves offer an effective approach to reduce the computational cost of ID-based cryptographic schemes. Afterwards, many ID-based cryptographic schemes based on bilinear pairings have been proposed such as signature schemes (Paterson, 2002; Cha and Cheon, 2003) and authenticated key agreement protocols (Chen and Kudla, 2003; McCullagh and Barreto, 2005).

In 2006, Das *et al*. (2006) proposed an efficient ID-based remote user authentication scheme with smart cards using bilinear pairings. Unfortunately, Goriparthi *et al*. (2006) showed that their scheme is insecure against forgery attack resulting in an adversary can always pass the authentication. Recently, Giri and Srivastava (2006) proposed an improved scheme to withstand the forgery attack. The computational cost required by the Giri–Srivastava scheme is too expensive, especially for smart cards with limited computing capability. Their scheme is also not well suited for a multi-server environment because they adopt one public-key encryption.

In this paper, we propose an efficient pairing-based user authentication scheme with smart cards. Remote users apply smart cards to generate the login messages and send them to the server. The smart card is a low power computing device while a server is regarded as a powerful node. Attempt to shift the computational burden to the powerful node and reduce the computational cost required by smart cards is a flexible approach. Performance analysis and experimental data of related pairing operations are given to demonstrate that our scheme is well suited for mobile devices with limited computing capability. We show that the proposed scheme is secure against adaptively chosen-message attack and ID attack in the random oracle model (Bellare and Rogaway, 1993; Pointcheval and Stern, 2000). Our scheme has the following merits: (1) the computational cost required by the user is reduced to be well suited for smart cards with limited computing capability; (2) users can freely choose and change their password without any assistance from the server; (3) the scheme is suitable for a multi-server environment in distributed networks.

The remainder of this paper is organized as follows. The preliminaries for bilinear pairings and security definitions are given in the next section. In Section 3, we briefly review the Giri–Srivastava scheme. Section 4 describes a new ID-based remote user authentication scheme from bilinear pairings using smart cards. The security analysis and discussions of the proposed scheme are presented in Section 5. In Section 6, the performance comparison among the proposed scheme and the recently proposed schemes is presented. Section 7 gives our conclusions.

2. Preliminaries

In this section, we introduce the concepts of bilinear pairings, as well as the related mathematical assumptions. Bilinear pairings such as Weil pairing and Tate pairing defined on elliptic curves have been used to construct efficient ID-based public-key cryptosystems (Boneh and Franklin, 2001; Boneh and Franklin, 2003). This section also presents the system setup phase of an ID-based public-key system and some notations used throughout the paper. Meanwhile, the framework and security model of a user authentication scheme with smartcard are defined here.

2.1. *Bilinear Pairings*

Let G_1 be an additive cyclic group with a prime order q and G_2 be a multiplicative cyclic group with the same order q. G_1 is a subgroup of the group of points on an elliptic curve over a finite field $E(F_p)$ and G_2 is a subgroup of the multiplicative group over a finite field. Let P be a generator of G_1. We refer to (Boneh and Franklin, 2001; Boneh and Franklin, 2003) for a fuller description of how these groups, maps and other parameters should be selected in practice for efficiency and security. A bilinear pairing is a map e: $G_1 \times G_1 \rightarrow G_2$ and it satisfies the following properties:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.

(2) Non-degenerate: there exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.

(3) Computability: there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

A bilinear map satisfying the three properties above is said to be an admissible bilinear map.

2.2. *Related Mathematical Assumptions*

For proving the security of the proposed scheme, some important mathematical problems and assumptions for bilinear pairings on elliptic curves are introduced as follows:

Decision Diffie–Hellman (*DDH*) *problem*: Given $P, xP, yP, zP \in G_1$ for some $x, y, z \in Z_q^*$, it is easy to verify $e(xP, yP) = e(P, zP)$. That is, DDH problem in G_1 is easy.

Computational Diffie–Hellman (*CDH*) *problem*: Given $P, xP, yP \in G_1$, finding xyP.

Computational Diffie–Hellman (*CDH*) *assumption*: No probabilistic algorithm can solve the CDH problem with non-negligible advantage within polynomial time.

Bilinear Diffie–Hellman (*BDH*) *assumption*: Given (P, xP, yP, zP) for some $x, y, z \in Z_q^*$, computing $e(P, P)^{xyz} \in G_2$ is hard.

Since the Decision Diffie–Hellman (DDH) problem in G_1 is easy, we cannot use the DDH assumption to build ID-based systems by bilinear pairings on elliptic curves. Instead, the security of the bilinear pairing ID-based system is based on a variant of the computational Diffie–Hellman assumption called the bilinear Diffie–Hellman (BDH) assumption. We refer to (Boneh and Franklin, 2001; Boneh and Franklin, 2003; Cha and Cheon, 2003) for the above assumptions in details.

2.3. *System Setup of ID-Based Authentication Scheme*

Without loss of generality, let *RS* be a registration server, *SS* be a service server. U_i is a legal user of the registration server and the service server. The user U_i wants to access services of the service server *SS* through an open network. A remote user authentication scheme is designed to authenticate the remote user over an open network. The following system parameters and notations are used throughout the paper.

- ID_i: the identity of the user U_i.
- $IDss$: the identity of the service server *SS*.
- pw_i: the password of the user U_i.
- P: a generator of the group G_1.
- s: the master private key of the registration server RS in Z_q^*.
- P_{RS}: the public key of the registration server RS such that $P_{RS} = s \cdot P$.
- $H_1()$: a one-way hash function $\{0,1\}^* \rightarrow \{0,1\}^n$, where n is the length of output (NIST/NSA, 2005). Note that the input of $H_1()$ could be the concatenation of some integer values and points on an elliptic curve. If the input includes some points on an elliptic curve, each point could be viewed two integer values, i.e., the x-coordinates and the y-coordinates of the point.

- $H_2()$: a map-to-point function $\{0,1\}^* \rightarrow G_1$ (Boneh and Franklin, 2001).
- T: a current time stamp.
- $\oplus$: a simple XOR operation in G_1. If $P_1, P_2 \in G_1$, P_1 and P_2 are points on an elliptic curve over a finite field, the operation $P_1 \oplus P_2$ means that it performs the XOR operations of the x-coordinates and the y-coordinates of P_1 and P_2, respectively.

2.4. *Framework and Security Model*

In the subsection, we first present the framework for a pairing-based (ID-based) user authentication scheme with smart cards. Smart cards are used to aid users to memorize their secret keys and some public parameters. Meanwhile, smart cards perform some cryptographic operations to generate login messages. In the user authentication scheme, there are three entities involved in this scheme: the user with a smart card, the service server (i.e., the verifier) and the registration server. In the following, we present the formal definition of an ID-based user authentication scheme with smart cards.

DEFINITION 1. A pairing-based (ID-based) user authentication scheme with smart cards is made of the following four algorithms:

- The s*etup algorithm*. The *setup algorithm* is run by the registration server on a given security parameter k, and produces the public parameters and a master private key.
- The *registration algorithm*. Given the identity (ID) and the password (pw) of a user, the registration server performs the *registration algorithm* to generate the user's private key. The registration server then loads related information and the private key into a smart card, and then issues it to the user.
- The *login algorithm*. Given a pair (ID, pw) and a time stamp T, the smart card performs the *login algorithm* to produces a login message σ.
- The *verification algorithm*. Given a login message σ, the service server runs the *verification algorithm* to check whether the login message is valid. If the validation holds, the *verification algorithm* outputs "Accept". Otherwise, it outputs "Reject".

In the following, we define the security model for an ID-based user authentication scheme with smart cards. According to the *verification algorithm* in Definition 1, we say the user authentication scheme is secure if it satisfies the security definition as defined below.

DEFINITION 2. A user authentication scheme with smart cards is secure against existential forgery on adaptively chosen-message and ID attack if no probabilistic polynomial-time adversary A has a non-negligible advantage in the following game played between a challenger C and the adversary A. Here, the challenger C plays both roles of the registration server and the service server.

- *Initialization*. The challenger C takes a security parameter k and runs the s*etup algorithm* to produce the public parameters and a master private key. Then C keeps the master private key and gives the public parameters to A.

- *Attack*. The adversary A may make a number of different types of queries in an adaptive manner as follows:
  - Hash query. Upon receiving the hash query, the challenger C computes the value of the hash function for the requested input and sends the hash value to the adversary A.
  - Registration query. Given the identity ID and the password pw, the challenger C uses the *registration algorithm* to return the private key corresponding to the pair (ID, pw) to the adversary A.
  - Login query. Given a pair (ID, pw) and a time stamp T, the challenger C produces a login message σ and sends it to the challenger A.
- *Forgery*. The adversary A outputs a forged login message σ^*, where σ^* did not appear in any login query. If the response of the *verification algorithm* on σ^* is "Accept", the adversary A wins the game. The advantage of the adversary A is defined as the probability that A wins.

3. Review of the Giri–Srivastava Scheme

Recently, Giri and Srivastava (2006) proposed an improved one on Das *et al*.'s scheme to withstand forgery attack. Because their scheme adopts the ID-based public-key cryptosystem to encrypt/decrypt a random secret r, the computational cost required by the Giri–Srivastava scheme is too expensive, especially for smart cards with limited computing capability.

For performance comparison, we briefly review the Giri–Srivastava scheme as follows. The scheme consists of four phases: the registration phase, the login phase, the verification phase and the password change phase.

[Registration phase]

In this phase, a user U_i securely submits his identity ID_i and password pw_i to the registration server *RS*.

1. The server computes $SP_i = pw_i \cdot P_{RS}$.
2. The server uses his master private key s to compute $Reg_i = s \cdot H_2(ID_i) + SP_i$.
3. The server loads P_{RS}, $SP_i, Reg_i, H_2()$ and ID_i into a smart card and issues the smart card to the user U_i.

[Login phase]

The user U_i inserts his smart card into the terminal, and he enters his identity ID_i and password pw_i. The smart card performs the following steps:

1. The smart card computes $A = pw_i \cdot P_{RS}$.
2. It computes $B = Reg_i - A$.
3. The smart card randomly selects an integer $r \in Z_q^*$. It then computes $C_i = E_{P_{RS}}(r)$ and $D_i = T \cdot B + r \cdot P_{RS}$, where $E_{P_{RS}}()$ is the ID-based public-key encryption function and T is the current time stamp.
4. Finally, the smart card sends the login message (ID_i, T, C_i, D_i) to the server.

Note that they refer to the Boneh–Franklin scheme (2001) for their ID-based public-key encryption and decryption.

[Verification phase]

As receiving the login message (ID_i, T, C_i, D_i) at time T', the server verifies the validity of the time interval between T' and T. If $(T' - T) > \Delta T$, then the server rejects the login request, where ΔT is the expected valid time for transmission delay. Otherwise, the server performs the following steps:

1. The server uses the master private key s to computes $X = E_s(C_i)$, where $E_s()$ is the corresponding public-key decryption function of $E_{P_{RS}}()$.
2. The server computes $Y = X \cdot P_{RS}$.
3. The server verifies $e(D_i - Y, P)? = e(H_2(ID_i), P_{RS})^T$. If it fails, then the server rejects the request; otherwise, the server accepts it.

Note that the server must keep the master private key s to decrypt X, it enable that their scheme is not well suited for a multi-server environment.

[Password change phase]

If the user U_i wants to change his password from pw_i to pw_i', he inserts his smart card into the terminal, and enters his identity ID_i, the old password pw_i and the new password pw_i'. The smart card performs the following steps:

1. The smart card computes $SP_i = pw_i \cdot P_{RS}$. The smart card checks the identity ID_i and SP_i. If they are correct, it continues the following steps.
2. The smart card computes $SP_i' = pw_i' \cdot P_{RS}$ and $Reg_i' = Reg_i - SP_i + SP_i'$.
3. The smart card stores new SP_i' and Reg_i' to replace SP_i and Reg_i.

4. Proposed Scheme

Here, we first present the multi-server environment. In the multi-server environment, there are a central registration server, n service servers and many legal users. The multi-server environment is depicted in Fig. 1. In many user authentication schemes (Jan and

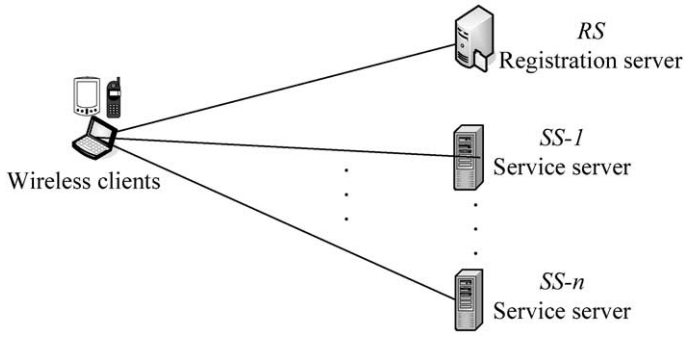

Fig. 1. The multi-server environment.

Chen, 1998; Chien *et al*., 2002; Ku and Chang, 2005; Liaw *et al*., 2006), the server must keep a system secret to verify the user's login message. The same problem is also found in the Giri–Srivastava scheme (2006) as reviewed in Section 3, in which the server must keep the master private key s to decrypt X. In fact, all these schemes are designed for single-server architecture. These schemes are not well suited for a multi-server environment. If a user wants to access multiple servers, the user must register with each server individually and remember several identifiers and the corresponding secrets.

In the following, we present our ID-based user authentication scheme with smart cards using bilinear parings. Our proposed scheme consists of four phases: the registration phase, the login phase, the verification phase and password change phase. In the registration phase, the user U_i registers with *RS* once. The *RS* then securely issues the user's ID to these service servers according to the access authorizations owned by the user. Each service server *SS* does not keep the system private key s and only stores the IDs of the legal users. Unlike other schemes (Jan and Chen, 1998; Chien *et al*., 2002; Ku and Chang, 2005; Liaw *et al*., 2006; Giri and Srivastava 2006), in our proposed scheme each service server does not keep the system private key s to authenticate users. Users do not need to register with each service server individually and remember several identifiers and the corresponding secrets. Thus, our proposed scheme is well suitable for the multi-server environment in distributed networks. The details of four phases in the proposed scheme are given as follows:

[Registration phase]

The registration phase is depicted in Fig. 2. A user U_i securely submits his identity ID_i and password pw_i to the registration server *RS* for registration. The registration server *RS* then performs the following steps:

1. *RS* computes $W_i = pw_i \cdot P$ and $CW_i = H_1(W_i)$.
2. *RS* computes $QID_i = H_2(ID_i)$.
3. The registration server *RS* uses his master private key s to compute $Reg_i = (s \cdot QID_i) \oplus W_i$.
4. The registration server *RS* loads $P, CW_i, Reg_i, H_1(), QID_i$ and ID_i into a smart card and issues the smart card to the user U_i. The server stores the ID_i into its database.

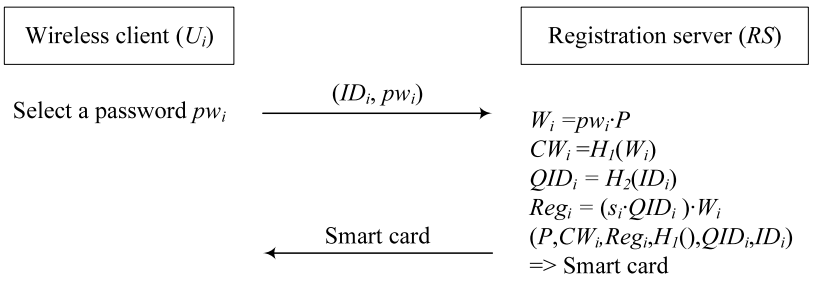

Fig. 2. The registration phase.

5. According to the authorized services of the user U_i, the registration server *RS* sends ID_i to the corresponding service servers securely.

[Login phase]

Without loss of generality, assume that the user U_i is a legal user of the service server *SS*. Fig. 3 depicts the login and verification phases between the user U_i and the service server *SS*. In the login phase, if the user U_i wants to access the server service *SS* with the identity $IDss$, the user U_i inserts his smart card into the terminal, and he then enters his identity ID_i and password pw_i as well as the service server identity $IDss$. The smart card performs the following steps:

1. The smart card computes $W_i = pw_i \cdot P$ and $CW_i = H_1(W_i)$. The smart card then checks ID_i and CW_i. If they are correct, it continues the following steps.
2. The smart card computes $DID_i = Reg_i \oplus W_i$, where DID_i is viewed as the secret key of U_i.
3. The smart card acquires the current time stamp T and randomly selects an integer $r \in Z_q^*$. It then computes $U = r \cdot QID_i$, $h = H_1(ID_i, IDss, T, U)$ and $V = (r + h) \cdot DID_i$.
4. Finally, the smart card sends the login message $(ID_i, IDss, T, U, V)$ to the service server *SS*, the login messages $(ID_i, IDss, T, U, V)$ can be viewed as a signature (U, V) on the message $(ID_i, IDss, T)$.

[Verification phase]

As receiving the login message $(ID_i, IDss, T, U, V)$ at time T', the service server *SS* first checks the validity of ID_i and verifies the validity of the time interval between T' and T for transmission delay. If two checks are correct, the service server *SS* performs the following steps:

1. The service server computes $QID_i = H_2(ID_i)$ and $h = H_1(ID_i, IDss, T, U)$.
2. The service server verifies $e(P_{RS}, U + h \cdot QID_i)? = e(P, V)$. If it holds, then the service server accepts the request; otherwise, the service server rejects it.

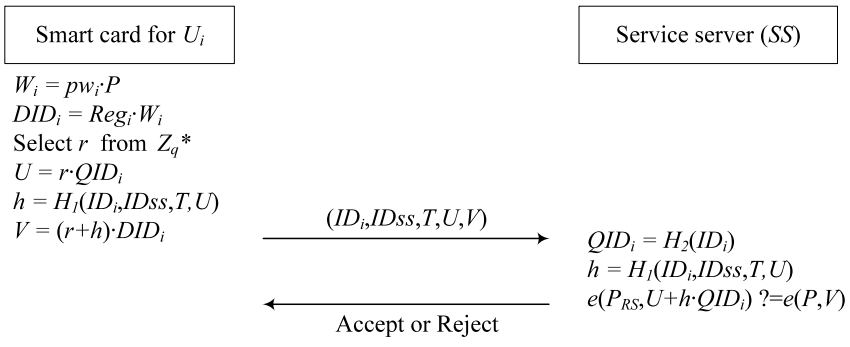

Fig. 3. The login and verification phases.

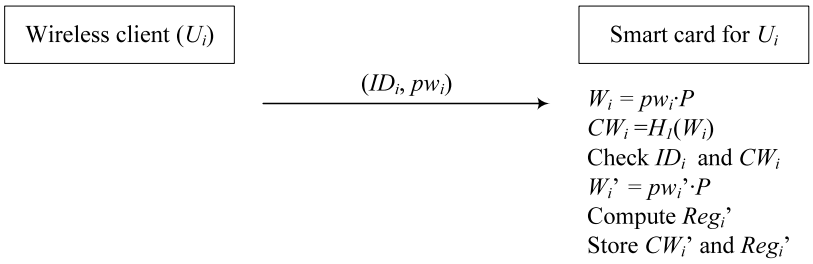

Fig. 4. The password change phase.

In the following, we present the correctness of the verification equation in Step 2.

$$\begin{aligned} e(P_{RS}, U + h \cdot QID_i) &= e(s \cdot P, r \cdot QID_i + h \cdot QID_i) \\ &= e(s \cdot P, (r+h) \cdot QID_i) = e(P, (r+h) \cdot QID_i)^s \\ &= e(P, (r+h) \cdot s \cdot QID_i) = e(P, (r+h) \cdot DID_i) \\ &= e(P, V). \end{aligned}$$

[Password change phase]

If the user U_i wants to change his password from pw_i to pw_i', he inserts his smart card into the terminal, and enters his identity ID_i, the old password pw_i and the new password pw_i'. The password change phase is depicted in Fig. 4. The detailed steps of the smart card are presented as below:

1. The smart card computes $W_i = pw_i \cdot P$ and $CW_i = H_1(W_i)$. The smart card checks ID_i and CW_i. If they are correct, it continues the following steps.
2. The smart card computes $W_i' = pw_i' \cdot P$ and $Reg_i' = Reg_i \oplus W_i \oplus W_i'$.
3. The smart card stores new CW_i' and Reg_i'.

5. Security Analysis and Discussions

5.1. *Security Analysis*

Let us discuss the security of the proposed scheme. The security of the proposed scheme is based on the Computational Diffie–Hellman (CDH) assumption, i.e., given $P, xP, yP \in G_1$, finding xyP is hard. That is, no probabilistic algorithm can solve the CDH problem with non-negligible advantage within polynomial time (Boneh and Franklin, 2001; Boneh and Franklin, 2003; Cha and Cheon, 2003). Based on the Computational Diffie–Hellman (CDH) assumption, we show that our scheme is secure against forgery attack and ID attack in the random oracle model (Bellare and Rogaway, 1993; Pointcheval and Stern, 2000).

In our scheme, the login messages $(ID_i, IDss, T, U, V)$ can be viewed as a signature (U, V) on the message $(ID_i, IDss, T)$, where T is the current time stamp. If we

can prove that an adversary without knowing the secret key DID_i of the user U_i cannot forge a valid signature on the message $(ID_i, IDss, T)$, then our scheme is secure against forgery attack and ID attack. We rigorously prove the following theorem using the Forking Lemma in (Pointcheval and Stern, 2000) and Lemma 1 in (Cha and Cheon, 2003) in the random oracle mode (Bellare and Rogaway, 1993). Note that in the random oracle model, the hash function can be seen as an oracle that produces a random value for each new query.

Theorem 1. *In the random oracle model, assume that adversary A with a non-negligible advantage can forge a valid login message $\sigma = (ID_i, IDss, T, U, V)$ for an adaptively chosen-message attack and ID attack to the proposed user authentication scheme. Then, there exists an algorithm C with a non-negligible advantage that can solve the computing discrete logarithm problem modulo a large prime.*

Proof. In the random oracle model, let A_0 is an algorithm within running time t_0 and with advantage ε_0 to perform an adaptively chosen-message attack and an ID-attack to our scheme.

Using Lemma 1 in (Cha and Cheon, 2003), it implies that there is an algorithm A_1 for an adaptively chosen-message attack and given fixed ID-attack which has running time $t_1 \leqslant t_0$ and advantage $\varepsilon_1 \leqslant \varepsilon_0(1 - 1/q)/q_{H_2}$, where q_{H_2} is the maximum number of oracle queries to H_2 hash function asked by A_0. Without loss of generality, we refer the given fixed ID to the identity ID_i of a legal user U_i. If there exists the above algorithm A_1 with a non-negligible advantage ε_1, then it implies that an adversary C without knowing the secret key DID_i of the legal user U_i can use A_1 to solve the CDH problem.

We assume that the adversary C receives a random instance (P, xP, yP) in G_1 and he wants to compute xyP. Let $P_{RS} = xP$ and $QID_i = H_2(ID_i) = yP$ are the system public key and user's public key, respectively. Then x simulates the master private key and is unknown to the adversary C. The adversary C acts as a challenger in the game defined in Definition 2. C gives public parameters, ID_i and ID_{SS} to the adversary A_1. C needs to maintain two lists L_1 and L_2 that are initially empty and are used to keep track of answers to $H_1()$ and login queries, respectively. The adversary C is responsible to answer the different queries of the adversary A_1 as follows:

- $H_1()$ queries. Upon receiving the hash query $H_1(\tau)(\tau = (ID_i, ID_{SS}, T, U)$ from the adversary A_1, the adversary C searches a pair (τ, R_h) in the list L_1. If such a pair is found, C returns R_h. Otherwise, he returns a random value $R_h \in_R \{0,1\}^n$, where n is the fixed length of the hash function $H_1()$. In order to avoid collisions on $H_1()$, no entry $(., R_h)$ exists in L_1. Then, C adds (τ, R_h) into the list L_1.
- Login query. The adversary A_1 chooses a time stamp T and sends it to the adversary C. The adversary C first generates two random value r_T and x_T from Z_q^* and produces a login message $\sigma = (ID_i, ID_{SS}, T, U, V)$ as follows: $U = r_T \cdot QID_i$, and $V = (r_T + h) \cdot x_T \cdot P$, where h is the simulated value of $H_1()$ query as mentioned above.. Then, C adds σ into the list L_2. Finally, C returns σ to the adversary A_1 as the answer.

Following the Forking Lemma in (Pointcheval and Stern, 2000), this lemma adopts the "oracle replay attack" using a polynomial replay of the attack with the same random tape and a different oracle. If there is an algorithm A_1 with a non-negligible probability ε_1 to generate a valid login message $\sigma = (ID_i, ID_{SS}, T, U, V)$, then the algorithm A_1 can generate two valid message $\sigma = (ID_i, ID_{SS}, T, U, V)$ and $\sigma' = (ID_i, ID_{SS}, T, U, V')$ with a non-negligible probability at least $\varepsilon_1/2$ such that $e(P_{RS}, U + h \cdot QID_i) = e(P, V)$ and $e(P_{RS}, U + h' \cdot QID_i) = e(P, V')$, where h and h' are two hash values of $H_1(ID_i, ID_{SS}, T, U)$ and $h \neq h'$ in the random oracle model. Since

$$e(P_{RS}, U + h \cdot QID_i) = e(P, V) \text{ and } e(P_{RS}, U + h' \cdot QID_i) = e(P, V'),$$

we have

$$e(x \cdot P, U + h \cdot y \cdot P) = e(P, V) \text{ and } e(xP, U + h' \cdot y \cdot P) = e(P, V').$$

By the bilinear property, we have

$$e(P, x \cdot U + h \cdot x \cdot y \cdot P) = e(P, V) \text{ and } e(P, x \cdot U + h' \cdot x \cdot y \cdot P) = e(P, V').$$

Therefore, we have $x \cdot U + h \cdot x \cdot y \cdot P = V$ and $x \cdot U + h' \cdot x \cdot y \cdot P = V'$. Then the adversary C can easily obtain xyP from $(V - V')/(h - h')$. That is, adversary C can compute the CDH problem from the random instance (P, xP, yP) in G_1, which is a contradiction for the Computational Diffie–Hellman (CDH) assumption. Therefore, we say that the assumption for the existence of algorithm A_1 with non-negligible advantage ε_1 is invalid.

By the contradiction proof, since there exists no algorithm A_1 with non-negligible advantage ε_1, it implies that no algorithm A_0 within running time t_0 and with advantage ε_0 to perform an adaptively chosen-message attack and an ID-attack to our scheme. Therefore, based on the Computational Diffie–Hellman (CDH) assumption, the proposed scheme is secure against forgery attacks and ID attacks in the random oracle model.

5.2. *Discussions*

In this subsection, we discuss several implementation issues of the proposed scheme.

(1) Eviction mechanism

For all user authentication schemes without the verification table, obviously the server does not store the password or verification table to authenticate the login user. However, when a user is revoked to access the services of some servers, there should be a mechanism that can process the situation. There are two practical approaches for the eviction mechanism. One is that the server stores a black ID list to record all revoked users. Another approach is that the server keeps a positive list containing all authorized users. In our scheme, we adopt the second approach. Each service server keeps only a positive ID list containing all authorized users. If a user is revoked to access the service of some server, the server has to delete his ID from the positive ID list.

(2) Clock synchronization problem

To resist replay attacks, in our scheme the smart card acquires the current time stamp T to generate the login message. As we all know, all authentication schemes resisting the replay attack with time stamp will suffer from the clock synchronization problem potentially. If the clock synchronization between the server and the user is not achieved, then the smart card should acquire a time stamp or a random challenge from the server. Nevertheless, it will increase extra transmission between the user and server but it does not affect the computational cost required by the smart card.

(3) Smart card security

In several literals (Ku *et al*., 2005; Ku and Chen, 2005), they discussed the security about smart cards. They assumed that the secrets stored in a smart card may be breached, so that they presented some weaknesses or attacks such as poor reparability or insider attack (Ku *et al*., 2005; Ku and Chen, 2005). In this article, we do not focus on the security about smart cards. We use smart cards to aid users to memorize their secret keys. Because these secret keys generated by the system authority in ID-based public key system are hard to memorized, we offers a simple mechanism to protect the users' secret keys. Certainly, the best approach is that each user can memorize these secret keys in their brains. We suggest that when users obtain their smart cards in the registration phase, they should immediately change their passwords by running the password change phase. Certainly, one self-protected mechanism (Rankl *et al*., 2000) should be provided to securely store these messages on the smart card. For example, the smart card should be invalided automatically if the user enters three times of invalid passwords. In additions, once a user loses his smart card, he should report it to his corresponding registration server.

6. Performance Comparisons

For convenience, the following notations are used to analyze the computational cost. We evaluate the computational time of the costly operations. We ignore some light-weight operations including modular addition in Z_q, point XOR on the group G_1. As we all know, they are much smaller than the following costly operations.

- TG_e: the time of executing the bilinear pairing operation e: $G_1 \times G_1 \rightarrow G_2$.
- TG_{mul}: the time for point scalar multiplication on the group G_1.
- TG_H: the time of executing the map-to-point hash function $H_2()$.
- TG_{add}: the time for point addition on the group G_1.
- T_H: the time of executing the one way hash function $H_1()$.
- T_{mul}: the time for modular multiplication in Z_q.

As we all know, a bilinear pairing operation (TG_e) is very time-consuming than other operations (Boneh and Franklin, 2001; Boneh and Franklin, 2003). As mentioned in Section 2, G_1 is an additive cyclic group with a prime order q and G_2 is a multiplicative cyclic group with the same order q. G_1 is a subgroup of the group of points on an elliptic curve over a finite field $E(F_p)$. Computational costs for the above operations with

a 160-bit prime q and a 512-bit prime p were given in (Cui *et al*., 2006). Performance simulation results show that one TG_e is about 7 times of one TG_{mul} and 15 times of TG_H, respectively. In additional, TG_{add}, T_H and T_{mul} are trivial in comparison with TG_e, TG_{mul} and TG_H. The Giri–Srivastava scheme (2006) reviewed in Section 3 uses an ID-based public-key encryption and decryption in login phase and verification phase, respectively. Here, we refer to the Boneh–Franklin scheme (2001) for their ID-based public-key encryption and decryption. In this case, an ID-based public-key encryption requires $\mathrm{TG}_e + 2\mathrm{TG}_{mul} + \mathrm{T}_H$ and an ID-based public-key decryption requires $\mathrm{TG}_e + \mathrm{T}_H$.

Table 1 demonstrates the performance comparisons among our scheme and the recently proposed ID-based user authentication schemes (Das *et al*., 2006; Giri and Srivastava, 2006) in terms of the computational costs for the registration phase, the login phase, the verification phase and the password change phase, respectively. From Table 1, it is obvious that our scheme has better performance in comparison with the recently proposed schemes. Although Das *et al*.'s scheme (2006) is also an efficient scheme, but their scheme is insecure against forgery attack (Goriparthi *et al*., 2006).

Since three ID-based authentication schemes are applied to authenticate users with smart cards, the computational cost of the login phase performed by smart cards is critical because smart cards are low power computing devices. Obviously, our scheme is better than the Giri–Srivastava scheme (2006), especially the login phase. The login phase on the wireless client in our scheme requires only 3 scalar multiplication operations on an elliptic curve and 2 one-way hash function operations over a finite field. The password phase requires only 2 scalar multiplication operations on an elliptic curve and one one-way hash function operations over a finite field.

Some previous implementations (Gupta *et al*., 2004; Gura *et al*., 2004; Han *et al*., 2002) of elliptic curve cryptographic primitives on smart cards or microprocessors have been developed. Recently, there are some implementations (Bertoni *et al*., 2006; Scott, 2005; Scott *et al*., 2006) of pairing operations on smartcards that have been reported. Here, we summary their implementation results. In (Scott *et al*., 2006), the processor on a Philips HiPersmart card offers a maximum clock speed of 36MHz and 16K RAM mem-

Table 1

Performance comparisons among the new scheme and the recently proposed schemes

	[1]Das *et al*.'s scheme (2006)	[2]Giri–Srivastava scheme (2006)	New scheme
Registration (Server)	$\mathrm{TG}_{mul} + 2\mathrm{TG}_H$	$2\mathrm{TG}_{mul}+\mathrm{TG}_H+\mathrm{TG}_{add}$	$2\mathrm{TG}_{mul}+\mathrm{TG}_H+\mathrm{T}_H$
Login (Smart card)	$2\mathrm{TG}_{mul}+\mathrm{TG}_H$	$\mathrm{TG}_e + 5\mathrm{TG}_{mul}$ $+2\mathrm{TG}_{add}+\mathrm{T}_H$	$3\mathrm{TG}_{mul} + 2\mathrm{T}_H$
Verification (Server)	$2\mathrm{TG}_e+\mathrm{TG}_{mul}$ $+\mathrm{TG}_H+\mathrm{TG}_{add}$	$3\mathrm{TG}_e + 2\mathrm{TG}_{mul}$ $+\mathrm{TG}_H+\mathrm{TG}_{add}$	$2\mathrm{TG}_e+\mathrm{TG}_{mul}+\mathrm{TG}_H$ $+\mathrm{TG}_{add}+\mathrm{T}_H$
Password change (Smart card)	$2\mathrm{TG}_H + 2\mathrm{TG}_{add}$	$2\mathrm{TG}_H + 2\mathrm{TG}_{add}$	$2\mathrm{TG}_{mul}+\mathrm{T}_H$

[1] Das *et al*.'s scheme (2006) was shown that it is insecure against forgery attack.

[2] Giri–Srivastava scheme (2006) is not suit for a multi-server environment.

Table 2
Experimental costs of operations on smart cards

	TG_{mul}	T_H (2048-bit block)	Login phase ($3TG_{mul} + 2T_H$)	Password change phase ($2TG_{mul}+T_H$)
HiPerSmart^{TM} (20MHz)	440 ms	About 1 ms	1.32 seconds	0.88 seconds
HiPerSmart^{TM} (36MHz)	270 ms	About 1 ms	0.81 seconds	0.54 seconds

ory. Some experimental data of one scalar multiplication and one hash function based on various clock speeds are given in Table 2. In which, G_1 is a subgroup of order q on an elliptic curve over a finite field $E(F_p)$, where p is a 512-bit prime and q is a 160-bit prime. Since the computational costs of the login phase and the password phase on wireless client sides require only $3TG_{mul} + 2T_H$ and $2TG_{mul}+T_H$, respectively. Thus, the required costs of both phases are about 1 second. It is obvious that our proposed scheme is well suited for smart cards with limited computing capability.

7. Conclusions

In this paper, we have proposed a pairing-based remote user authentication scheme using smart cards. We have shown that the proposed scheme is secure against forgery attack and ID attack under the computational Diffie–Hellman assumption. As compared with the recently proposed schemes, our scheme has better performance in term of the computational cost. Experimental data of related pairing operations on smartcards are given to demonstrate that our scheme is well suited for mobile devices with limited computing capability Our scheme is also well suitable for a multi-server environment in distributed networks. As we all know, some pairing-based authenticated key agreement protocols offering mutual authentication and session key establishment still requires some expensive computations. The design of ID-based mutual authentication and session key establishment schemes with low power computing devices using bilinear pairings is a critical research issue.

Acknowledgements

The authors would like to thank the referees for their valuable comments and suggestions.

References

Awasthi, A.K., and S. Lal (2003). A remote user authentication scheme using smart cards with forward secrecy. *IEEE Trans. on Consumer Electron.*, **49**(4), 1246–1248.

Bellare, M., and P. Rogaway (1993). Random oracles are practical: a paradigm for designing efficient protocols. In *Proc. 1st Annual ACM Conference on Computer and Communications Security* (*ACM CCS'93*). pp. 62–73.

Bertoni, G.M., L. Chen, P. Fragneto, K.A. Harrison and G. Pelosi (2005). Computing tate pairing on smartcards. White Paper, STMicroelectronics. Available:
http://www.st.com/stonline/products/families/smartcard/ches2005_v4.pdf

Boneh, D., and M. Franklin (2001). Identity-based Encryption from the Weil pairing. In *Advances in Cryptology-CRYPTO 2001*, *LNCS*, vol. 2139. pp. 213–229.

Boneh, D., and M. Franklin (2003). Identity based encryption from the Weil pairing. *SIAM J. of Computing*, **32**(3), 586–615.

Cha, J.C., and J.H. Cheon (2003). An identity-based signature from gap Diffie–Hellman groups. In *PKC 2003*, *LNCS*, vol. 2567. pp. 18–30.

Chen, L., and C. Kudla (2003). Identity based authenticated key agreement from pairings. In *IEEE Computer Security Foundations Workshop*. pp. 219–233.

Chien, H.Y., J.K. Jan and Y.M. Tseng (2002). An efficient and practical solution to remote authentication: smart card. *Computers and Security*, **21**(4), 372–375.

Cui, S., P. Duan and C.W. Chan (2006). An efficient identity-based signature scheme with batch verifications. In *Proceedings of the First International Conference on Scalable Information Systems* (*INFOSCALE'06*).

Das, M.L., A. Saxena, V.P. Gulati and D.B. Phatak (2006). A novel remote user authentication scheme using bilinear pairings. *Computers and Security*, **25**(3), 184–189.

Giri, D., and P.D. Srivastava (2006). An improved remote user authentication scheme with smart cards using bilinear pairings. In *Cryptology ePrint Archive*. Available:
http://eprint.iacr.org/2006/274.pdf

Goriparthi, T., M.L. Das, A. Negi and A. Saxena (2006). Cryptanalysis of recently proposed Remote User Authentication Schemes. In *Cryptology ePrint Archive*. Available:
http://eprint.iacr.org/2006/028.pdf

Gupta, V., D. Stebila and S. Fung (2004). Speeding up secure web transactions using elliptic curve cryptography. In *Proceedings of 11th Network and Distributed Systems Security Symposium*. pp. 231–239.

Gura, N., A. Patel, A. Wander, H. Eberle and S.C. Shantz (2004). Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proceedings of Cryptographic Hardware and Embedded Systems*. pp. 119–132.

Han, J.H., Y.J. Kim, S.I. Jun, K.I. Chung and C.H. Seo (2002). Implementation of ECC/ECDSA cryptography algorithms based on Java card. In *Proceedings of 22nd International Conference on Distributed Computing Systems Workshops*. pp. 272–276.

Hwang, M.S., and L.H. Li (2000). A new remote user authentication scheme using smart cards. *IEEE Trans. Consum. Electron.*, **46**(1), 28–30.

Jan, J.K., and Y.Y. Chen (1998). Paramita wisdom: password authentication scheme without verification tables. *Journal of Systems and Software*, **42**, 45–47.

Ku, W.C., and S.T. Chang (2005). Impersonation attack on a dynamic id-based remote user authentication scheme using smart cards. *IEICE Transactions on Communications*, **E88-B 5**(5), 2165–2167.

Ku, W.C., and S.M. Chen (2005). Cryptanalysis of a flexible remote user authentication scheme using smart cards. *ACM Operating Systems Review*, **39**(1), 90–96.

Ku, W.C., M.H. Chiang and S.T. Chang (2005). Weaknesses of Yoon–Ryu–Yoo's hash-based password authentication scheme. *ACM Operating Systems Review*, **39**(1), 85–89.

Lamport, L. (1981). Password authentication with insecure communication. *Commun. of ACM*, **24**, 770–772.

Liaw, H.T., J.F. Lin and W.C. Wu (2006). An efficient and complete remote user authentication scheme using smart cards. *Mathematical and Computer Modelling*, **44**, 223–228.

McCullagh, N., and P.S.L.M. Barreto (2005). A new two-party identity-based authenticated key agreement. In *Proceedings of CT-RSA 2005*, *LNCS*, vol. 3376. pp. 262–274.

NIST/NSA FIPS 180-2 (2005). *Secure Hash Standard* (*SHS*). NIST/NSA, Gaithersburg, MD, USA.

Paterson, K. (2002). ID-based signatures from pairings on elliptic curves. *Electronics Letters*, **38**(18), 1025–1026.

Pointcheval, D., and J. Stern (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptography*, **13**, 361–396.

Rankl, W., W. Effing and R. Wolfgang (2000). *Smart Card Handbook'*, 2nd ed. John Wiley & Sons.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – Crypto '84*, *LNCS*, vol. 196. pp. 47–53.

Scott, M. (2005). Computing the tate pairing. In *CT-RSA*, *LNCS*, vol. 3376. Springer-Verlag. pp. 293–304.

Scott, M., N. Costigan and W. Abdulwahab (2006). Implementing cryptographic pairings on smartcards. In

Cryptology ePrint Archive. Available: `http://eprint.iacr.org/2006/144.pdf`

Tseng, Y.M. (2006). GPRS/UMTS-aided authentication protocol for wireless LANs. *IEE Proceedings – Communications*, **153**(6), 810–817.

Tseng, Y.M. (2007). A resource-constrained group key agreement protocol for imbalanced wireless networks. *Computers & Security*, **26**(4), 331–337.

Y.-M. Tseng received the BS degree in computer science and engineering from National Chiao Tung University, Taiwan, Republic of China, in 1988; and the MS degree in computer and information engineering from National Taiwan University in 1990 and the PhD degree in applied mathematics from National Chung-Hsing University in 1999. He is currently a professor in the Department of Mathematics, National Changhua University of Education, Taiwan. He is a member of IEEE Communications Society and the Chinese Cryptology and Information Security Association (CCISA). His research interests include cryptography, information security, network security, computer network and mobile communications. In 2006, his paper obtained the Wilkes Award from *The British Computer Society*. He is also editors of three international Journals: *Computer Standards & Interfaces*, *International Journal of Security and Its Applications*, and *Journal of Security Engineering.*

T.-Y. Wu received the BS and the MS degrees in Department of Applied Mathematics, Tatung University, Taiwan, in 2003 and 2005, respectively. He is currently a PhD candidate in Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include applied cryptography and pairing-based cryptography.

J.-D. Wu received the MS degree in Department of Mathematics, National Changhua University of Education, Taiwan in 2007. His research interests include communication security, wireless and mobile communications.

Bevielio tinklo intelektualiųjų kortelių vartotojų tapatybės nustatymo porinis metodas

Yuh-Min TSENG, Tsu-Yang WU, Jui-Di WU

Straipsnyje nagrinėjamas efektyvus ir saugus vartotojų tapatybės nustatymo bitiesinis porinis metodas. Remiantis Diffie–Hellman prielaidomis parodyta, kad metodas yra saugus esant adaptyviai parinktam pranešimo tekstui. Palyginus su neseniai pasiūlytais tapatybės nustatymo poriniais metodais, nagrinėjamas metodas yra pranašesnis skaičiavimo kainos prasme ir gali būti naudojamas paskirstytųjų tinklų daugiaserverinėje aplinkoje. Pateikta metodo kokybinė analizė bei eksperimentų duomenys su intelektualiųjų kortelių porinėmis operacijomis, kurių tikslas – parodyti, kad nagrinėjamas metodas gerai tinka intelektualiosioms kortelėms ir mobiliesiems prietaisams, turintiems ribotas skaičiavimo galimybes.