A mutual authentication and key exchange scheme from bilinear pairings for low power computing devices

Yuh-Min Tseng*, Tsu-Yang Wu and Jui-Di Wu
Department of Mathematics, National Changhua University of Education, Chang-Hua City 500, Taiwan, R.O.C.
*Corresponding author’s E-mail: ymtseng@cc.ncue.edu.tw

Abstract

In a client-server network environment, a mutual authentication and key exchange scheme is an important security mechanism to provide two parties with the property that they can authenticate each other’s identity while they may construct a common session key. With rapid growth of mobile wireless networks, the computational cost on the client side with low power computing devices is a critical factor of the security scheme design. This paper presents a mutual authentication and key exchange scheme using bilinear pairings. Based on the computational Diffie-Hellman assumption and the random oracle model, we show that the proposed scheme is secure against passive attack, forgery attack and ID attack while it provides mutual authentication, implicit key confirmation and partial forward secrecy. A performance analysis demonstrates that our scheme is well suited for smart cards with limited computing capability.

1. Introduction

Now, handheld devices are popularly used by people and many mobile applications have rapidly developed such as wireless internet services, mobile access services and mobile e-commerce. If public-key based cryptographic schemes are designed for mobile users with handheld devices or smart cards, the computational cost on the user side is a critical issue in practical implementation because of their limited computing capability [1,2,3].

Recently, Boneh and Franklin [4,5] proposed a practical ID-based encryption system based on bilinear pairings. Bilinear 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 [6,7,8] and authenticated key agreement protocols [9,10,11,12].

Although ID-based authenticated key agreement protocols [9,10,11,12] can offer mutual authentication and key exchange, these protocols did not consider the computational cost on the user side with handheld devices. The computational cost on both sides in all these authenticated key agreement protocols requires at least one bilinear pairing operation. But, a bilinear pairing operation is more time-consuming than other operations [4,5].

In 2006, Das et al. [13] proposed an efficient ID-based remote user authentication scheme with smart cards using bilinear pairings. Unfortunately, Goriparthi et al. [14] showed that their scheme is insecure against forgery attack resulting in an adversary can always pass the authentication. Recently, Giri and Srivastava [15] 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 because it still requires one bilinear pairing operation. In additions, both schemes do not provide mutual authentication and key exchange between the user and the server.

In this paper, we propose a mutual authentication and key exchange scheme using bilinear pairings. Users with smart cards 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. We shift the computational burden to the powerful node and reduce the computational cost required by smart cards. Based on the computational Diffie-Hellman assumption [4,5] and the random oracle model [16], we show that the proposed scheme is secure against passive attack, forgery attack and ID attack while it provides implicit key confirmation and partial forward secrecy. We make a performance
analysis to demonstrate that our scheme is well suited for smart cards with limited computing capability. Compared to Das et al.’s scheme [13] and the Giri-Srivastava scheme [15], our scheme has the following merits: (1) Mutual authentication between the user and the server is achieved; (2) A secure session key is established; (3) The computational cost required by the user is reduced to be well suited for smart cards.

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 cryptosystems [4,5,7,8] for the following assumptions in details.

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 $E(F_q)$ 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 [4,5] 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 $a, b \in \mathbb{Z}_q^*$. 

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 $P, Q \in G_1$.

For proving the security of the proposed scheme, some important mathematical assumptions for bilinear pairings on elliptic curves are introduced. We refer to [4,5,7,8] for the following assumptions in details.

**Computational Diffie-Hellman (CDH) assumption:**

Given $P, xP, yP \in G_1$, it is hard to find $xyP$.

**Discrete Logarithm (DL) assumption:** Given $P, Q \in G_1$, finding an integer $x \in \mathbb{Z}_q^*$ such that $Q = xP$ is hard.

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

2.2. System setup of ID-based system

Without loss of generality, let $RS$ be a registration server and $U_i$ be a legal user of the registration server. The user $U_i$ wants to access services of the registration server $RS$ through an open network. The following system parameters and notations are used throughout the paper.

- $P$: a generator of the group $G_1$.
- $s$: the master private key of the registration server $RS$ in $\mathbb{Z}_q^*$.
- $P_{RS}$: the public key of the registration server $RS$ such that $P_{RS} = sP$.
- $H()$: a one-way hash function $\{0,1\}^k \rightarrow \{0,1\}^k$, where $k$ is the length of output. [17]
- $H_{ID}$: a map-to-point function $\{0,1\}^k \rightarrow G_1$.
- $ID_i$: the identity of the user $U_i$.
- $pw_i$: the password of the user $U_i$.
- $DID_i$: the secret key of the user $U_i$.
- $QID_i$: the public key of the user $U_i$ such that $QID_i = H_{ID}(ID_i)$.
- $T$: a current time stamp.
- $\oplus$: a simple XOR operation 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.

3. Proposed scheme

Here, we present a mutual authentication and key exchange scheme from bilinear pairings using smart cards. There are three entities in the proposed scheme, namely, the user, the user’s smart card and the registration server $RS$. The scheme consists of three phases: the registration phase, the mutual authentication 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 $RS$ for registration. The server then performs the following steps:

1. The registration server computes $W_i = pw_iP$ and $CW_i = H_{ID}(W_i)$.
2. The registration server computes $QID_i = H_{ID}(ID_i)$.
3. The registration server uses his master private key $s$ to compute $Reg_i = (sQID_i) \oplus W_i$.
4. The registration server loads $P, P_{RS}, CW_i, Reg_i, H_{ID}, QID_i$ 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.

### [Mutual authentication phase]

In the mutual authentication phase, the user $U_i$
communicates with the registration server RS. It provides mutual authentication between the user and the registration server while a session key is established. The user $U_i$ inserts his smart card into the terminal, and he enters his identity $ID_i$ and password $pw_i$. The communication steps between the smart card and the registration server are presented as follows:

1. The smart card computes $W_i = pw_iP$ and $CW_i = H_i(W_i)$. The smart card then checks $ID_i$ and $CW_i$. If they are correct, it continues the following steps.
2. The smart card computes $DID_i = Reg_iW_i$, where $DID_i$ is viewed as the secret key of the user $U_i$.
3. The smart card acquires the current time stamp $T$ and randomly selects an integer $r \in Z_q^*$. It then computes $U = rP$, $K_1 = rP_{RS}$, $h = H_i(ID_i, T, U)$ and $V = rQID_i + hDID_i$.
4. The smart card sends $(DID_i, T, U, V)$ to the registration server.
5. As receiving $(ID_i, T, U, V)$ at time $T'$, the server first checks the validity of $ID_i$. If $(T' - T) > \Delta T$, then the server rejects the request, where $\Delta T$ is the expected valid time for transmission delay. If two checks hold, the server performs the following Steps 6 and 7.
6. The server computes $QID_i = H_i(ID_i)$ and $h = H_i(ID_i, T, U)$. The server then verifies $e(QID_i, U + hP_{RS}) = e(P, V)$. If it holds, then the server accepts the request; otherwise, the server rejects it.
7. The server acquires the current time stamp $T''$, and computes $K_2 = sU$ and $Auth_i = H_i(P_{RS}, ID_i, T', U, V, K_2)$. The server then sends $(T', Auth_i)$ to the smart card.
8. As receiving $(T', Auth_i)$ at time $T''$ the smart card verifies the validity of the time interval between $T'$ and $T''$ for transmission delay. The smart card may authenticate the server by checking $Auth_i = H_i(P_{RS}, ID_i, T', U, V, K_2)$. It is obvious that $K_2 = sU = sP = rP_{RS}= K_i$.
9. After running the above steps, both the smart card and the server can compute a common session key $SK = H_i(Auth_i, T, T', U, V, K_i) = H_i(Auth_i, T', T, U, V, K_i)$.

Here, we present the correctness in Step 6.

\[e(QID_i, U + hP_{RS}) = e(QID_i, rP + hsP) = e(QID_i, (r + hs)P) = e(P, (r + hs)QID_i) = e(P, rQID_i + hsQID_i) = e(P, rQID_i + hDID_i) = e(P, V)\]

### [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 $W'_i = pw'_iP$ and $CW_i = H_i(W'_i)$. The smart card checks $ID_i$ and $CW_i$. If they are correct, it continues the following steps.
2. The smart card computes $W'_i = pw'_iP$ and $Reg'_i = Reg_iW'_i$.
3. The smart card stores new $CW'_i$ and $Reg'_i$.

### 4. Security analysis

Let us discuss the security of the proposed scheme. Based on the Computational Diffie-Hellman (CDH) assumption and the random oracle mode [16], we show that the proposed scheme offers mutual authentication, implicit key confirmation and partial forward secrecy, and it is secure against passive adversaries.

#### 4.1. Providing mutual authentication

We show that the registration server can authenticate the user. In our scheme, the login messages $(ID_i, T, U, V)$ is viewed as a signature $(U, V)$ on the message $(ID_i, T)$ [7,8]. We prove that an adversary without knowing the secret key $DID_i$ of the user $U_i$ cannot forge a valid signature on the message $(ID_i, T)$. We rigorously prove the following theorem using the Forking Lemma in [18] and Lemma 1 in [8] under the random oracle mode.

**Theorem 1.** Under the random oracle model and the Computational Diffie-Hellman (CDH) assumption, an adversary $E$ without knowing the secret key $DID_i$ of any user $U_i$ cannot generate the valid message $(ID_i, T, U, V)$, so that the registration server can authenticate the user $U_i$.

**Proof.** In the random oracle model, let $A_0$ be an algorithm within running time $t_0$ and with advantage $\epsilon_0$ to perform an adaptive chosen message attack and an ID-attack to our scheme. Using Lemma 1 in [8], it implies that there is an algorithm $A_i$ for an adaptive chosen message attack and given fixed ID-attack which has running time $t_i \leq t_0$ and advantage $\epsilon_1 \leq \epsilon_0(1-1/q_2)$, where $q_2$ 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 $A_i$ with negligible advantage $\epsilon_1$, then it implies that an adversary $E$ without knowing the secret key $DID_i$ of the
legal user $U_i$ can use $A_i$ to solve the CDH problem. We assume that the adversary $E$ receives a random instance $(P, xP, yP)$ in $G_1$ and he wants to compute $xyP$. Let $P_{RS} = xP$ and $QID = H_A(ID_i) = yP$ are the system public key and user’s public key, respectively. Then $x$ simulates the master private key and is unknown to the adversary $E$. Following the Forking Lemma in [18], 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 $A_i$ with a non-negligible probability $\epsilon_1$ to generate a valid signature $(U, V)$ for the message $(ID_i, T)$, then the algorithm $A_i$ can generate two valid message signatures $(ID_i, T, U, V)$ and $(ID_i, T, U, V')$ with a non-negligible probability at least $\epsilon_2/2$ such that $e(QID_i, U+h\cdot P_{RS}) = e(P, V)$ and $e(QID_i, U+h'\cdot P_{RS}) = e(P, V')$, where $h$ and $h'$ are two hash values of $H(ID_i, T, U)$ and $h \neq h'$ under the random oracle model. Since $e(QID_i, U+h\cdot P_{RS}) = e(P, V)$ and $e(QID_i, U+h'\cdot P_{RS}) = e(P, V')$, we have

\[
e(yP, U+h-x\cdot P) = e(P, V) \text{ and } e(yP, U+h'+x\cdot P) = e(P, V').
\]

By the bilinear property, we have

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

Therefore, we have $y\cdot U+ h\cdot x\cdot P = V$ and $y\cdot U+ h'\cdot x\cdot P' = V'$. Then the adversary $E$ can easily obtain $xyP$ from $(V-V')(h-h')$. That is, adversary $E$ 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 $A_i$ with non-negligible advantage $\epsilon_1$ is invalid.

By the contradiction proof, since there exists no algorithm $A_i$ with the non-negligible advantage $\epsilon_1$, it implies that no algorithm $A_i$ within running time $t_i$ and with advantage $\epsilon_0$ to perform an adaptive 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 attack and ID attack under the random oracle model.

4.2. Other security properties

(1) Passive adversaries

Passive adversary is that if an attacker is unable to obtain the established session key by eavesdropping messages transmitted over the broadcast channel, the key exchange scheme is secure against passive adversaries.

**Theorem 3.** Under the Computational Diffie-Hellman assumption, the proposed scheme is secure against passive adversaries.

**Proof.** In the proposed scheme, the common session key $SK$ between the user and the registration server is computed by $H_1(Auth_i, T', U, V, K_j)$ and $H_2(Auth_i, T', U, V, K_j)$, respectively. Only the user with knowing $r$ and the registration server with knowing $s$ can compute correct $K_1$ and $K_2$. It is based on the difficulty of computing the Discrete Logarithm (DL) problem to compute $r$ and $s$ from $U$ and $P_{RS}$, respectively. By Theorem 2, we know that to compute $K_1$ or $K_2$ from the known messages $U$ and $P_{RS}$ is based on the difficulty of the Computational Diffie-Hellman problem. It is clear that the proposed scheme is secure against passive adversaries under the Computational Diffie-Hellman assumption.

(2) Implicit key confirmation

A key exchange scheme offers implicit key confirmation, if a party $U_i$ is assured that $U_i$ can compute the session key and no one other than $U_i$ can compute the session key.

**Theorem 4.** Under the random oracle model and the Computational Diffie-Hellman (CDH) assumption, the proposed scheme provides implicit key confirmation.

**Proof.** By Theorems 1 and 2, we have shown that the user and the registration server can authenticate with each other under the random oracle model and the Computational Diffie-Hellman (CDH) assumption. By Theorem 3, we know that no one than both the user and the registration server can compute the session key $SK$ between the user and the registration server. Therefore,
the proposed scheme provides implicit key confirmation. □

(3) Partial forward secrecy

A key exchange scheme offers forward secrecy if compromise of a long-term key cannot result in the compromise of previously established session keys. Obviously, if the secret key $s$ of the registration server is compromised by an attacker, then the attacker can obtain the previous session key. On the other hand, if the secret key $DID_i$ of a user $U_i$ is compromised by an attacker, he tries to compute $r$ from $V = r\cdot QID_i + h\cdot DID_i$. It is based on the difficulty of computing the Discrete Logarithm (DL) problem in $G_1$. Therefore, the proposed scheme offers only partial forward secrecy.

5. Discussions and performance analysis

5.1. Discussions

In this subsection, we discuss 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.

(2) Clock synchronization problem

To resist replay attacks, 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 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) The security of smart card

In several literals [19,20], they discussed the security of smart cards. They assumed that the secret keys stored in a smart card can be breached, so that they presented some attacks such as poor reparable or insider attacks [19,20]. Here, we assume that the secret keys stored in smart cards cannot be revealed by attackers. We use smart cards to aid users to memorize their secret keys. When users obtain their smart cards in the registration phase, they should immediately change their passwords by running the password change phase in our proposed protocol. Meanwhile, one self-protected mechanism [21] should be provided to securely store these secret keys on the smart card.

5.2. Performance analysis

For convenience, the following notations are used to analyze the computational cost. We ignore some light-weight operations including modular addition in $Z_q$, point XOR on the group $G_1$. 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 $G_1$.
- $TG_{Hf}$: the time of executing the map-to-point hash function $H_f()$.
- $TG_{add}$: the time for point addition on the group $G_1$.
- $T_{mul}$: the time for modular multiplication in $Z_q$.
- $T_H$: the time of executing the hash function $H_1()$.

As we all know, a bilinear pairing operation ($TG_e$) is more time-consuming than other operations [4,5]. Table 1 summarizes the performance result of the proposed scheme in terms of the computational costs for the registration phase, the mutual authentication phase and the password change phase, respectively. From Table 1, we know that the device on the user side does not require expensive bilinear pairing operation. Some previous implementations [22,23] of elliptic curve cryptographic primitives on smart cards or microprocessors can give an evidence to demonstrate that the proposed scheme is well suited for smart cards with limited computing capability.

Table 1 Performance evaluation of the proposed scheme.

<table>
<thead>
<tr>
<th></th>
<th>User</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration</td>
<td>0</td>
<td>$2TG_{hmul} + TG_{Hf}$</td>
</tr>
<tr>
<td>Mutual</td>
<td>$5TG_{add} + 4T_{Hf}$</td>
<td>$2TG_{Hf} + 2TG_{mul} + TG_{Hf} + TG_{add} + 3T_{Hf}$</td>
</tr>
<tr>
<td>Authentication</td>
<td>2$T_{Hf} + 2TG_{add}$</td>
<td>0</td>
</tr>
</tbody>
</table>

6. Conclusions

In this paper, we have proposed an ID-based mutual authentication and key exchange scheme using bilinear pairings. We have shown that the proposed
scheme is secure against passive attack, forgery attack and ID attack under the random oracle model and the computational Diffie-Hellman assumption. Our scheme offers mutual authentication, implicit key confirmation and partial forward secrecy. In the proposed protocol, we shift the computational burden to the server and reduce the computational cost required by the user. As a result, the computational cost required by the user is reduced to be well suited for smart cards. As compared to Das et al.’s scheme and the Giri-Srivastava scheme, our scheme offers mutual authentication and key exchange between the user and the server.

Acknowledgements

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

References