Security Enhancement of a Remote User Authentication Scheme Using Bilinear Pairings and ECC

Duc-Liem Vo and Kwangjo Kim
International Research center for Information Security
Information and Communications University
119 Munji-ro, Yuseong-gu, Daejeon 305-732, Korea
{vdliem,kkj}@icu.ac.kr

Abstract

Remote authentication is an important mechanism to control user access to remote systems and a password-based authentication is a preferable method. With advances in elliptic curve cryptography, Jia et al. [10] proposed a remote user authentication scheme with a smart card. Their scheme utilized bilinear pairings and an elliptic curve ElGamal encryption scheme to provide a secure authentication mechanism. However, we show that their scheme is vulnerable to our impersonation attack which any adversary can be authenticated successfully with probability 1 at no extra cost. We also suggest our provably secure improvement scheme which is verified to be more efficient from the point of computational complexity than the original scheme.

1. Introduction

Remote authentication over insecure communications is an important application of cryptographic protocols. The first construction, proposed by Lamport [11] in 1981, can resist against a replaying attack but it will be vulnerable if the verifier, who is holding the password table, is compromised. To overcome this weakness, several schemes [4, 5, 9, 13] have eliminated the use of the password table and utilized a smart card as an authentication token for users. A smart card provides a low cost communication, computation and convenience for users. In 2006, Das et al. [6] have proposed a novel remote authentication scheme with a smart card using bilinear pairings. This scheme allows users to choose their password freely and requires no password table for verifying the legitimacy of users. Later, Chow et al. [3] have presented a possible impersonation attack on the scheme [6] and also have provided a solution to fix the scheme. But, Goriparthi et al. [8], again, have indicated that both Das’s and Chow’s schemes are vulnerable to forgery, replaying and insider attacks. Recently, Jia et al. [10] have utilized bilinear pairings along with the ElGamal version of elliptic curve cryptosystem in order to design a new remote authentication scheme withstanding the previous attacks. Nevertheless, in this paper, we show that Jia et al.’s authentication scheme is not secure by presenting our impersonation attack on their scheme. In addition, we point out the problem in Jia et al.’s construction and a method to fix it.

Organization: In the next section, we brief concepts of bilinear pairings and related security problems. We review Jia et al.’s scheme in Section 3 and propose an attack on the scheme in Section 4. An improvement scheme and its analysis are shown in Sections 5 and 6, respectively. Section 7 ends with concluding remarks.

2. Bilinear Pairings

Let $G_1$ and $G_2$ be additive and multiplicative groups of the same prime order $q$, respectively. Let $P$ be a generator of $G_1$. Assume that the discrete logarithm problems in both $G_1$ and $G_2$ are hard. Let $e : G_1 \times G_1 \rightarrow G_2$ be a pairing which satisfies the following properties:

- Bilinear: $e(aP, bP') = e(P, P')^{ab}$ for all $P, P' \in G_1$ and all $a, b \in \mathbb{Z}_q^*$.
- Non-degenerate: If $e(P, P') = 1 \forall P' \in G_1$ then $P = O$.
- Computable: There is an efficient algorithm to compute $e(P, P')$ for any $P, P' \in G_1$.

Under such group $G_1$, we can define the following hard cryptographic problems:

- **Discrete Logarithm (DL) Problem:** Given $P, P' \in G_1$, find an integer $n$ such that $P = nP'$ if such integer exists.
- **Computational Diffie-Hellman (CDH) Problem:** Given a triple $(P, aP, bP) \in G_1$ for $a, b \in \mathbb{Z}_q^*$, find $abP \in G_1$.
- **Decision Diffie-Hellman (DDH) Problem:** Given a quadruple $(P, aP, bP, cP) \in G_1$ for $a, b, c \in \mathbb{Z}_q^*$, decide
whether \( c = ab \pmod{q} \) or not.

The CDH assumption states that there is no polynomial time algorithm can solve the CDH problem with non-negligible probability. Details about bilinear pairings and related problems can be found in [1, 2, 7].

3. Review of Jia et al.’s scheme [10]

Jia et al.’s scheme [10] scheme consists of four phases: setup, registration, authentication and password change, which are described below.

Phase 1: Setup

The remote server (RS) chooses an additive group \( G_1 \) and a multiplicative group \( G_2 \) of the same prime order \( q \). \( P \) is a generator of the group \( G_1 \). Let \( e : G_1 \times G_1 \rightarrow G_2 \) be a bilinear map and \( H(\cdot) : \{0, 1\}^* \rightarrow G_1 \) be a cryptographic hash function. The RS selects a private key \( s \in_R \mathbb{Z}_q^* \) and computes its corresponding public key \( P_{rs} = sP \). The server publishes the system parameters \( \{G_1, G_2, e, q, P, P_{rs}, H\} \) while keeping \( s \) secret.

Phase 2: Registration

Step 2-1. A user \( U_i \) submits his identity \( ID_i \) and password \( pw_i \) to the RS.

Step 2-2. Upon receiving a request from \( U_i \), the RS computes: \( Reg_i = sH(ID_i) + H(pw_i) \).

Step 2-3. The RS personalizes a smart card with the parameters: \( \{ID_i, Reg_i, H(\cdot), P, P_{rs}\} \) and distributes the card to \( U_i \) over a secure channel.

Phase 3: Authentication

This phase includes user’s login and RS’s verification.

Step 3-1. The user \( U_i \) inserts the smart card into the input device and enters his identity \( ID_i \) and password \( pw_i \). If the information matches with the data stored in the smart card, proceed to the next step, otherwise, reject.

Step 3-2. The smart card computes \( D_i = T \cdot Reg_i \) and \( V_i = T \cdot H(pw_i) \), where \( T \) is a current timestamp. The smart card picks a random integer \( k \) and computes: \( C_1 = kP; C_2 = (D_i - V_i) + kP_{rs} \). After that, the terminal sends a login message \( \{ID_i, C_1, C_2, T\} \) to the RS over a public channel.

Step 3-3. Receiving a login request at a timestamp \( T' \), the RS verifies if \( (T' - T) \geq \Delta T \), where \( \Delta T \) denotes the expected valid time interval for transmission delay, then the RS rejects the login request.

Step 3-4. The RS checks if the following equation holds:

\[
e(C_2 - sC_1, P) = e(H(ID_i), P_{rs})^T \tag{1}
\]

If Eq. (1) holds, the RS accepts the login request, otherwise rejects.

Phase 4: Password Change

The user \( U_i \) can change password without assistance from the RS. He performs the following steps:

Step 4-1. The user \( U_i \) inputs his \( ID_i \) and the old password \( pw_i \). The smart card checks validity by the equation:

\[
e(Reg_i, P) = e(H(ID_i), P_{rs})e(H(pw_i), P) \tag{2}
\]

If the equation holds, the smart card allows the user to change his password.

Step 4-2. The user inputs a new password \( pw_i' \).

Step 4-3. The smart card stores the new authentication information: \( Reg_i' = Reg_i - H(pw_i) + H(pw_i') = sH(ID_i) + H(pw_i') \).

4. Weakness of Jia et al.’s scheme

Jia et al. [10] claimed that the remote user authentication scheme is secure against the forgery attack by using the ElGamal encryption to provide confidentiality to the registration information \( D_i \) and \( V_i \). However, we show that their scheme could not sustain an impersonation attack. From eavesdropping on the login requests of a user, an attacker can produce a fake login request which helps the attacker bypass the authentication check of the RS as a legitimate user later. Our attack works as follows:

Assuming that an attacker succeeded to eavesdrop on a login request sent by the user \( U_i \) to the RS at time \( T_1 \) is \( \{ID_i, C_1', C_2', T_1\} \). The attacker modifies the login request to the new one which can be used to login at time \( T_2 \). He computes: \( C_1' = T_1^{-1}T_2C_1' \) and \( C_2' = T_1^{-1}T_2C_2' \). The new login request \( \{ID_i, C_1', C_2', T_2\} \) can pass the verification, Eq. (1), of the RS as shown below:

\[
e(C_2' - sC_1', P) = e(T_1^{-1}T_2C_2 - T_1^{-1}T_2sC_1, P) = \nonumber \\
e(T_1^{-1}T_2(C_2 - sC_1), P) = \nonumber \\
e(T_1^{-1}T_2D_i - V_i + kP_{rs} - skP, P) = \nonumber \\
e(T_1^{-1}T_2(D_i - V_i), P) = \nonumber \\
e(T_1^{-1}T_2(T_1(Reg_i - H(pw_i))), P) = \nonumber \\
e(T_2((Reg_i - H(pw_i)), P) = \nonumber \\
e(T_2((sH(ID_i) + H(pw_i) - H(pw_i)), P) = \nonumber \\
e(T_2sH(ID_i), P) = e(H(ID_i), P_{rs})T_2 
\]

By intercepting the communication line, the attacker sends this new login message and authenticates successfully with the RS and uses service freely.

5. Our Improvement Scheme

The problem of Jia et al. [10] scheme is that they did not guarantee the integrity of the login message, thus, an attacker just modifies an eavesdropped login message and can impersonate a legitimate user successfully. We present an improvement scheme which overcomes this problem.
The setup and registration phases are the same as Jia et al.’s scheme except that in the registration phase, the RS employs an additional operation, a hash function $h : \{0,1\}^* \to \mathbb{Z}_q^*$. The authentication phase is described as follows:

**Phase 3: Authentication**

As in the original scheme, the user first passes the smart card’s identification then sends a login request to the RS. The RS checks the login request to verify if that user is legitimate or not. The user $U_i$ logs in by the following steps:

**Step 3-1.** $U_i$ inserts the smart card into the input device and enters his identity $ID_i$ and password $pw_i$. The smart card ensures this is an actual user by checking the equation:

$$e(Reg_i - H(pw_i), P) = e(H(ID_i), P_{rs})$$  \hspace{1cm} (3)

If the equation holds, $U_i$ is a legitimate user and the smart card goes to the next step, otherwise, it cancels.

**Step 3-2.** The smart card selects a random integer $k$ and performs calculation: $C_1 = kP, C_2 = h(ID_i, T)(Reg_i - H(pw_i)) + kP_{rs}$. Here, $T$ is the timestamp at the computation. After that, the terminal sends a login message $\{ID_1, C_1, C_2, T\}$ to the RS over a public channel.

**Step 3-3.** Receiving a login request at a timestamp $T'$, the RS verifies if $(T' - T) \geq \Delta T$, where $\Delta T$ is the expected valid transmission delay interval, then the RS rejects the login request. Otherwise, the RS proceeds the next step.

**Step 3-4.** The RS checks the following equation:

$$e(C_2 - sC_1, P) = e(h(C_1, T)H(ID_i), P_{rs})$$  \hspace{1cm} (4)

If Eq. (4) holds, the login is accepted, otherwise rejected.

The correctness of Eq. (4) can be checked easily:

$$e(C_2 - sC_1, P) = e(h(C_1, T)(Reg_i - H(pw_i) + kP_{rs}) - skP, P) = e(h(C_1, T)(Reg_i - H(pw_i)), P) = e(h(C_1, T)sH(ID_i), P) = e(h(C_1, T)H(ID_i), P_{rs})$$

**Phase 4: Password Change**

The steps to change password are the same as the original scheme except how the user identifies himself to the smart card. The smart card will validate the user by checking Eq. (3) instead of Eq. (2). Other steps keep unchanged.

**Remarks:** For simpler implementation to the smart card, we can adopt an exclusive OR, $\oplus$, instead of using the hash function $h$. Utilizing $\oplus$ operation in the scheme will minimize the change in the smart card implementation. Usually, this is a basic operation having very low computational complexity. In this case, the only quantity $h(C_1, T)$ will be changed to $C_1(x) \oplus T$, where $C_1(x)$ is $x$-coordinate of the $C_1$. The remaining part of the scheme is unchanged. The security of the scheme is maintained.

### 6. Discussion

#### 6.1. Security Analysis

In our improvement scheme, the replaying attack can be avoided due to using a timestamp technique as in the original scheme. Given that the adversary recorded a login message $\{ID_1, C_1, C_2, T\}$, if he wants to authenticate at the later time, he needs to recompute values $C_1$ and $C_2$ to pass the verification in Eq. (4). This cannot be done without knowing the secret value $Reg_i$ and $pw_i$.

For the impersonation attack, one can consider $\{C_1, C_2\}$ is a signature on the timestamp value $T$ with randomness $C_1$ embedded. Hence, if an adversary forges these values successfully, it can be shown that there exists an algorithm which can break the CDH problem in the group $G_1$.

**Theorem 1.** If an adversary can perform an impersonation attack on the authentication scheme, there exists an algorithm which can break the CDH problem in the group $G_1$.

**Proof (Sketch):** We use the same proving technique in [14]. Suppose that there exists an adversary $A$ performs the impersonation attack successfully on the authentication scheme for a given $ID$ within a time bound $t$ with the probability $\epsilon$. $A$ can query $h, H$, and user authentication for at most $q_h, q_H$ and $q_P$ times, respectively. Assume that $\epsilon \geq 10(q_P + 1)(q_h + q_P)/q$. We can construct an algorithm $B$ which breaks the CDH problem ($P, aP, bP$) in $G_1$ as follows: The algorithm $B$ sets $P_{rs} = aP$. For any identity $ID_i$ other than the identity of the impersonated user $ID$, output the hash query $H(ID) \equiv x_iP$ for $x_i \in R \mathbb{Z}_q^*$. If $ID = ID_i$, set $H(ID) = bP$. The output of the query to $h(\cdot, h_j$, where $j = 1, 2, \ldots, q_P$, is chosen randomly from $\mathbb{Z}_q^*$. At the registration phase, the algorithm $B$ also can answer the registration request from $A$ by returning $Reg_i = x_iP_{rs} + H(pw_i)$, where $pw_i$ is given by $A$ and $H(pw_i)$ can be chosen randomly from the group $G_1$. $A$ is not allowed to ask this type of queries for $ID_i = ID$.

$B$ also provides login requests $\{ID_{ij}, C_{1ij}, C_{2ij}, T_{ij}\}$ to $A$ for verification. $B$ selects $r_j \in R \mathbb{Z}_q$ and computes the login request $\{ID_{ij}, C_{1ij}, C_{2ij}, T_{ij}\}$ as follows:

$$C_{1ij} = r_jP - h_jH(ID_{ij})$$ and $C_{2ij} = r_jP_{rs}$.

$A$ verifies the correctness of the login request by the following equation which is equivalent to Eq. (4):

$$e(C_{2ij}, P) = e(C_{1ij} + h(C_{1ij}, T_{ij})H(ID_{ij}), P_{rs})$$  \hspace{1cm} (5)
Later, $A$ outputs a valid login request for the user $ID$ as $(ID, C_1, C_2, T)$. By replaying $A$ with the same random tape but a different hash function $h'$ and using the forking lemma in [12], $B$ comes up with two different login requests $(ID, C_1, C_2, T)$ and $(ID, C_1, C_2', T)$ such that $h'(C_1, T) \neq h(C_1, T)$, hence $C_2 \neq C_2'$, with probability $\geq 1/9$. Finally, $B$ obtains $abP$ from

$$abP = \frac{C_2' - C_2}{h'(C_1, T) - h(C_1, T)}$$

The total running time $t'$ of $B$ is bounded by $23q_{h,t}/\epsilon$ as per the forking lemma [12].

Regarding the password change capability, users perform changing their password without intervention of the RS. Moreover, this will prevent a malicious RS from using their identities information illegally. The risk of the storing password also is avoided.

6.2. Performance

In Table 1, $H$ and $h$ are hash operations, $P$ is pairing computation, $A$ and $S$ are elliptic curve point addition and scalar multiplication, respectively. $E$ is exponentiation of pairing value.

<table>
<thead>
<tr>
<th>Phases</th>
<th>Improvement</th>
<th>Jia et al.[10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg.</td>
<td>$2H + S$</td>
<td>$2H + S$</td>
</tr>
<tr>
<td>Login</td>
<td>$A + 3S + H + h$</td>
<td>$A + 4S + H$</td>
</tr>
<tr>
<td>Verfy.</td>
<td>$2S + A + 2P + H + h$</td>
<td>$S + A + E + 2P + H$</td>
</tr>
<tr>
<td>Pwd.Chg.</td>
<td>$A + 2P$</td>
<td>$3P$</td>
</tr>
</tbody>
</table>

Table 1. Computational Complexity.

Our improvement does not degrade the performance of the original scheme. In fact, by changing the calculation slightly, our scheme is more efficient than the original scheme. In the original scheme, to validate a user, the smart card has to evaluate Eq. (2), consuming 3 pairing operations. In our scheme, this verification is done by Eq. (3) which has only 2 pairing operations. The additional point adding operation in this equation is much cheaper than 1 pairing operation. The similarity is done in the computation of $C_1$ and $C_2$, saving 1 scalar multiplication. The detailed comparison is given in Table 1.

7. Concluding Remarks

In this paper, we reviewed the authentication scheme using bilinear pairings proposed by Jia et al. and showed a proper impersonation attack on this scheme. Fortunately, we are able to fix the scheme and provide an improvement scheme with provable security. The new authentication scheme is more secure as well as more efficient than the original scheme. For further work, we consider to develop a new authentication scheme which is supporting mutual authentication for preventing malicious remote servers from deriving information of registered users to use illegally.

References