# Impersonation Attack of a Mutual Authentication Scheme Using Smart Card and its Improvement

Duc-Liem Vo* and Kwangjo Kim*

*International Research center for Information Security (IRIS),

Information and Communications University (ICU)

## Abstract

In 2006, Jeon *et al.* [2] proposed a mutual authentication using smart cards and claimed that their scheme is secure against various attacks including impersonation attack. However, we point out that Jeon et al.'s scheme is vulnerable to impersonation attack in which an attacker, without knowing any secret information of either users or the server, can identify himself as a legitimate user or an authentic server, respectively. Finding out the weakness of Jeon et al.'s scheme, 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.

## I. Introduction

In many applications, *e.g.* financial transactions or e-commerce applications, unilateral authentication only does not provide guaranteed security. Although a remote user is authenticated with the remote systems, the remote user is not assured if the remote systems are genuine ones or not and in the worst case, the user's information can be stolen by a fake system. Therefore, it is necessary to provide a mutual authentication scheme where each party has to authenticate each other. In 2006, Jeon *et al.* [2], by improving Das *et al.*'s scheme [1], proposed a mutual authentication scheme using pairings that provided security as well as efficiency.

However, in this paper, we point out that Jeon *et al.*'s scheme, denoted as JKKLY06 scheme for short, is not secure as stated by the authors [8] and show appropriate impersonation attacks on their scheme. An adversary can mount these attacks and pretend either the remote user or the remote system successfully. We also construct a secure improvement mutual authentication scheme which is more efficient than the original scheme.

**Organization.** We brief concepts of bilinear pairings and related security problems in the next section. We review and demonstrate our attacks on JKKLY06 scheme in Section 3. Section 4 is followed by our revised improvement. We analyze our improvement scheme in terms of security and performance in Section 5. Section 6 ends with concluding remarks.

## II. Bilinear Pairings

Let $G_1$ and $G_2$ be additive and multiplicative groups of the same prime order $q$. Let $P$ be a generator of $G_1$. Let $e:G_1\times G_1\rightarrow G_2$ be a pairing which satisfies the following properties:

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

2. Non-degenerate: For all $P'\in G_1$ if $e(P,P')=1$ then $P=O$.

3. 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 mathematical problems:

- Discrete Logarithm (DL) Problem: Given $P$, $P'\in G_1$, find an integer $n$ such that $P=nP'$ whenever such integer exists.

- Computational Diffie-Hellman (CDH) Problem: Given a triple $(P, aP, bP)\in G_1$ for $a, b\in Z_q^*$, find the element $abP$.

- Decision Diffie-Hellman (DDH) Problem:

Given a quadruple $(P, aP, bP, cP) \in G_1$ for $a, b, c \in Z_q^*$, decide whether $c=ab \pmod q$ or not.

The CDH assumption states that there is no polynomial time algorithm can solve CDH problem with non-negligible probability.

# III. Review of JKKLY06 [2]

## 1. Summary of JKKLY06

JKKLY06 mutual authentication scheme consists of three phases: setup, registration, and authentication and. briefly described as follows:

**Setup phase**: The remote server (RS) selects a private key $s$ then computes the corresponding public key $P_S=sP$. RS chooses two cryptographic hash functions $H_1:\{0,1\}^* \rightarrow G_1$ and $H_2:\{0,1\}^* \rightarrow Z_q^*$; then publishes the system parameters $\{G_1, G_2, e, q, P, P_S, H_1, H_2\}$.

**Registration phase**:

**R1.** A user $U_i$ submits his identity $ID_i$ and password $pw_i$ to RS for a registration request.

**R2.** Upon receiving the request from the user, the RS computes $Reg_{IDi}=sH_1(ID_i)+H_1(pw_i)$.

**R3.** The RS personalizes a smart card with the parameter $\{ID_i, Reg_{IDi}, H_1, H_2\}$ and sends it to $U_i$ over a secure channel.

**Authentication phase:**

*Login phase:* The user $U_i$ inserts the smart card into a terminal and inputs $ID_i$ and $pw_i$. The smart card performs:

**L1.** Check if $ID_i$ is identical to the one which is stored in the smart card.

**L2.** Generate a nonce $n_i \in Z_q^*$.

**L3.** Compute $V_i=n_iP$; $t=H_2(T||V_i^x||V_i^y)$ where $T$ is the user system's timestamp, $V_i^x$ and $V_i^y$ are $x$ and $y$ coordinates of $V_i$, respectively.

**L4.** Compute $D_{IDi}=n_i^{-1}(Reg_{IDi} - H_1(pw_i) + tP)$.

**L5.** Send the login request $\{ID_i, D_{IDi}, V_i, T\}$ to the RS over a public channel.

*User authentication phase:* On receiving the login request $\{ID_i, D_{IDi}, V_i, T\}$ from the user at time $T^*$, the RS performs following operations:

**U1.** Verify the valid time interval $\Delta T \geq T^*-T$.

**U2.** Check if $e(D_{IDi},V_i)=e(H_1(ID_i),P_S)e(P,P)^t$. If this equation holds, the RS accepts the login request. Otherwise, reject it.

**U3.** Compute $V_R=sT_RH_1(ID_i)$ and send $\{V_R, T_R\}$ to the user where $T_R$ is RS's timestamp.

*Server Authentication*: On receiving the message $\{V_R, T_R\}$, the user performs:

**S1** Verify the valid time interval $\Delta T \geq T^*-T_R$.

**S2.** Check if $T_R(Reg_{IDi}-H_1(pw_i))=V_R$. If this holds, the user is assured of the authentic RS.

## 2. Our Attacks

Suppose that an attacker eavesdrops on the genuine authentication messages and captures $\{ID_i, D_{IDi}, V_i, T\}$ from a user and $\{V_R, T_R\}$ from RS. From RS's message, he can computes:

$$S_{IDi}=T_R^{-1}V_R=T_R^{-1}sT_RH_1(ID_i)=sH_1(ID_i). \quad (1)$$

With the secret value $S_{IDi}$, an attacker can mount impersonation attacks to pretend either a legitimate user or an authentic server to the specific user. The detailes are given below:

**User Impersonation Attack**: an attacker can masquerade as a legitimate user by computing a new login request at time $T'$ as follows:

- Choose $n_i' \in Z_q^*$ and compute $V_i'=n_i'P$.
- Compute $t'=H_2(T'||V_i^{x\prime}||V_i^{y\prime})$ and $D_{IDi}'= n_i'^{-1}(S_{IDi} + t'P)$; then send $\{ID_i,D_{IDi}',V_i',T'\}$ to RS.

This login message passes the checking process done by the RS in step **U2** as usual:

$e(D_{IDi}',V_i') = e(n_i'^{-1}(S_{IDi}+t'P),n_i'^{-1}P) = e(sH_1(ID_i)+ t'P, P) = e(sH_1(ID_i),P)e(t'P, P)=e(H_1(ID_i),P_S)e(P, P)^{t'}$

**Server Impersonation Attack:** an attacker pretends the legitimate server to $U_i$ by computing a new server authentication message:

- Pick up an appropriate timestamp value $T_R'$.
- Compute $V_R'=T_R'S_{IDi}=T_R'H_1(ID_i)$.
- Send $\{V_R', T_R'\}$ to the user $U_i$.

This server authentication message passes the checks done by $U_i$ in step **S2** as usual:

$T_R'(Reg_{IDi}-H_1(pw_i)) = T_R'(sH_1(ID_i) + H_1(pw_i) - H_1(pw_i)) = sT_R'H_1(ID_i) = V_R'$

## IV. Our Revised Scheme

Our scheme includes four phases which is described details as below:

**Initialization phase:** This phase is same as **Setup** phase of JKKLY06 scheme.

**Registration phase:** A user $U_i$ registers with the RS by the following steps:

**R'1.** $U_i$ submits his identity $ID_i$ and $w_i = H_1(pw_i)$ to RS, where $pw_i$ is his password.

**R'2.** Receiving a request from $U_i$, RS computes: $Reg_{IDi} = sH_1(ID_i) \oplus w_i$ and $V_i = sH_2(w_i)^{-1}H_1(ID_i)$.

**R'3.** The RS personalizes a smart card with the registration information: $\{ID_i, Reg_i, V_i, H_1, H_2, P, P_S\}$ and sends the card to $U_i$ securely.

**User authentication phase:** The user $U_i$ inserts the smart card into the terminal and inputs $ID_i$ and $pw_i$.

**U'1.** The smart card checks for the legitimate user by: $Reg_i \oplus H_1(pw_i) = H_2(H_1(pw_i))V_i$.

**U'2.** The smart card picks a random number $r \in Z_q^*$ and computes: $C_1 = rP$, $C_2 = h(T, C_1)(Reg_i \oplus H_1(pw_i)) + rP_S$, where $T$ is a current timestamp; then sends a login message $\{ID_i, C_1, C_2, T\}$ to RS.

**U'3.** Receiving the login message from $U_i$, at $T^*$, the RS checks the valid time interval $\Delta T \geq T^* - T$; then verifies if $U_i$ is real by equation:

$$e(C_2, P) = e(H_2(T, C_1)H_1(ID_i) + C_1, P_S) \quad (2)$$

**U'4.** The RS gets a timestamp value $T$ and computes $U = H_2(T, sC_1)$. The RS sends a server authentication message $\{T, U\}$ to the user.

**Server authentication phase:** Receiving $\{T, U\}$ from RS at time $T^*$, $U_i$ performs:

**S'1.** Check valid time interval $\Delta T \geq T^* - T$

**S'2.** Verify if $U = H_2(T, rP_S)$. RS is authenticated if the equation holds, otherwise, it fails.

**Password change capability:** Users can change password without help from the RS:

**P'1.** The user $U_i$ inserts the smart card into the terminal and chooses the password change function. He is required to input $ID_i$ and $pw_i$. The smart card checks for the legitimate user by $Reg_i \oplus H_1(pw_i) = H_2(H_1(pw_i))V_i$. Only if it holds, the smart card allows $U_i$ to change password:

**P'2.** The user enters a new password $pw_i^*$.

**P'3.** The smart card updates $Reg_i$ and $V_i$ by:

$$Reg_i^* = Reg_i \oplus H_1(pw_i) \oplus H_1(pw_i^*) = sH_1(ID_i) \oplus H_1(pw_i^*)$$

$$V_i^* = H_2(H_1(pw_i^*))^{-1}H_2(H_1(pw_i))V_i = H_2(H_1(pw_i^*))^{-1}sH_1(ID_i)$$

**Correctness:** The correctness of Eq. (2) in the user authentication phase is shown below:

$$e(C_2, P) = e(H_2(C_1, T)((Reg_i \oplus H_1(pw_i)) + rP_S, P) = e(H_2(C_1, T)((sH_1(ID_i) \oplus H_1(pw_i) \oplus H_1(pw_i)) + rP_S, P) = e(H_2(C_1, T)sH_1(ID_i) + rsP, P) = e(H_2(C_1, T)H_1(ID_i) + rP, PS) = e(H_2(C_1, T)H_1(ID_i) + C_1, P_S)$$

The correctness of the server authentication message in step **S'2** is obvious:

$$U^* = H_2(T^*, sC_1) = H_2(T^*, srP) = H_2(T^*, rsP) = H_2(T^*, rP_S).$$

## V. Discussion

### 3. Security Analysis

We assume that a smart card is physically secure so that it is impossible to extract information from the smart card. Under this assumption, we examine security of our scheme against various attacks. The replay attack is impossible since the revised scheme utilizes timestamp technique. Suppose an adversary recorded a login message $\{ID_i, C_1, C_2, T\}$ and a server authentication message $\{T^*, U^*\}$. If the adversary wants to apply a new timestamp, he needs to recompute values $C_1$ and $C_2$ to pass the verification in Eq. (2) or the verification in step **S'2**. This cannot be done without knowing the secret value $Reg_i$, $pw_i$, and $r$ simultaneously for the user authentication case, and the RS's private key $s$ for the server authentication case. The revised scheme also allows the users to change their passwords without dependence on the RS. For the user impersonation attack, one can consider $\{C_1, C_2\}$ as a signature on the timestamp value $T$ with randomness $C_1$ embedded. We have a theorem:

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

*Proof (Sketch)*: We use the same proving technique in [4]. 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 $\varepsilon$. **A** can query $H_1$, $H_2$, and login request for at most $q_{H1}$, $q_{H2}$ and $q_P$ times. Assume that $\varepsilon \geq 10(q_P+1)(q_{H2}+q_P)/q$. We can construct an algorithm **B** which breaks the CDH problem $(P, aP, bP)$ in $G_1$.

The algorithm **B** sets $P_S=aP$. **B** can answer the hash queries, registration queries and login request queries since it controls the hash oracles. **B** also makes a special hash for an identity $ID_i=ID$, *i.e.* $H(ID_i)=bP$. **A** is not allowed to ask the registration query for $ID_i=ID$. After interacting with **B**, at sometime **A** outputs a valid login request for the user $ID$. By replaying **A** with the same random tape but a different hash function $H_2'$ and using the forking lemma [3], **B** can obtain $abP$ with probability 1/9 and time bounded by $23q_{H2}t/\varepsilon$ as per the forking lemma [3]. ■

Our scheme is also secure against a server impersonation attack. To forge a server authentication message $\{T^*, U^*\}$, the adversary must produce $U^*=rP_S=rsP$. Clearly, $\{C_1, P_S, U^*\}$ is a CDH tuple. If the adversary outputs $U^*$, he must solve the CDH problem in $G_1$ properly. Under the CDH assumption, the adversary is not able to perform this attack to our scheme.

## 4. Performance

In Tables 1 and 2, H is a hash-to-point operation and h is an integer hash operation. P is pairing computation, A and M are elliptic curve point addition and scalar multiplication operations. E is an exponentiation of a pairing value. X is an exclusive OR operation.

Table 1: Computation at the RS side

| Algorithm | Our scheme | JKKLY06 [2] |
|---|---|---|
| Registration | 2M+I+X+H+h | M+A+2H |
| User Authen. | A+M+2P+H+h | A+3P+E+H |
| ServerAuthen. | M+h | 2M+H |

As can be seen in Table 1, the computation at the RS is more efficient than JKKLY06 scheme, especially we can save a pairing and an exponentiation computation.

Table 2: Computation at the user side

| Algorithm | Our scheme | JKKLY06 [2] |
|---|---|---|
| Registration | H | |
| User Authen. | X+A+4M+H+2h | 3M+2A+I+H+h |
| ServerAuthen. | h | M+A |

Table 2 shows comparison of the total computation at the user side. Our scheme requires just one more scalar multiplication but provides a strong method for smart card to check for a correct user before authentication.

## VI. Concluding Remarks

Mutual authentication has become more important in growing up online applications such as financial transactions, e-commerce, and government services. In this paper, we analyzed a mutual authentication scheme using smart cards JKKLY06 [2] and demonstrated the impersonation attacks on their scheme in both user and server sides. We also suggested a revised construction which is superior to JKKLY06 in both security and performance aspects. Our revised scheme is secure against various known attacks while providing beneficially computational cost which is suitable for resource constraint devices.

## References


[1] M.L. Das, A. Saxena, V.P. Gulati, and D.B. Phatak, *A Novel Remote User Authentication Scheme Using Bilinear Pairings*, Computers & Security, Vol. 25, 3 (2006), pp. 184-189.

[2] Jun-Cheol Jeon, Byung-Heon Kang, Se-Min Kim, Wan-Soo Lee, and Kee-Young Yoo, *An Improvement of Remote User Authentication Scheme Using Smart Cards*, The 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006), Dec. 2006, Hong Kong, China, LNCS 4325, pp. 416-432, 2006.

[3] D. Pointcheval and J. Stern, *Security Argument for Digital Signatures and Blind Signatures*, Journal of Cryptology, Springer-Verlag, Vol. 13 No. 3, pp. 361-396, 2000.

[4] HyoJin Yoon, Jung Hee Cheon, and Yongdae Kim, *Batch Verifications with ID-Based Signatures*, In Proc. of the 7th International Conference on Information Security and Cryptology - ICISC 2004, Seoul, Korea, LNCS 3506, pp. 233-248, 2005.