A Robust and Efficient Timestamp-based Remote User Authentication Scheme with Smart Card Lost Attack Resistance

Hong-Bin Tang[1], Xin-Song Liu[1], and Lei Jiang[2]
(Corresponding author: Hong-Bin Tang)

School of Computer Science and Engineering, University of Electronic Science and Technology of China[1]
Chengdu 610054, China
Sichuan Provincial People's Government Social Development Research Center, Chengdu 610054, China [2]
(Email: tanghongbin@uestc.edu.cn)

(Received Jan. 09, 2012; revised and accepted June 12, 2012)

Abstract

Password-based authentication scheme with smart card is an important part of security for accessing remote servers. In 2011, Awasthi et al. proposed an improved timestamp-based remote user authentication scheme to eliminate the attacks in Shen et al.'s. However, we find that their scheme is vulnerable to the privileged insider, the lost smart card, the password guessing, the replay, the modification, and the denial of service attacks. We propose a timestamp-based remote user authentication scheme using elliptic curve cryptography to fix such problems. Our scheme is based on elliptic curve discrete logarithm problem (ECDLP) and provides lost smart card attack resistance. The user can choose and change his or her password freely and the server need not maintain a password verifier table in its database in our scheme. Furthermore, our scheme is proved to be more secure than Awasthi et al.'s. And it is more efficient than the previous timestamp-based authentication schemes.

Keywords: Authentication, cryptography, password, smart card, protocol

1 Introduction

Authentication in essence is a process of verifying the authenticity of one's claim about its identity. It is one of the most important aspects of computer security, since other security services all depend upon it. Particularly, password-based authentication scheme with smart card is an important part of securely accessing remote server. A variety of schemes have been proposed to allow a legitimate user to log into a remote server and access the resources [1,3,4,5,6,7,8,9,11,12,15,16,17,18,22,23].

In 1999, Yang and Sheih [23] proposed a timestamp-based password remote authentication scheme using smart card to remove the needs of the password tables or the verification tables at the end of the server. However, it was inconvenient since the user had to send his/her smart card and a new password to the server to change his or her password. And since it was a unilateral authentication scheme, it may encounter the server spoofing attack. Meanwhile, the scheme was found to be vulnerable to the forged login attacks by Chan et al. [3], Fan et al. [6], and Shen et al. [22] independently.

In 2003, Shen et al. [22] proposed an improved mutual authentication scheme to resist the forged login attacks. However, in 2008, Liu et al. [18] pointed out that Shen et al.'s scheme did not resist the forged login attacks, since the attacker could intercept the legal user's login request and register the smart card to carry out the attacks.

In 2011, Awasthi et al. [1] pointed out that Shen et al.'s scheme suffered from the lost smart card and the forged login attacks, and suggested a timestamp-based improvement. Unfortunately, their scheme was still vulnerable to the lost smart card and other attacks.

In this paper, we point out that these previous reported timestamp-based authentication protocols are vulnerable to various attacks. Meanwhile, these schemes are inefficient since they use RSA cryptosystem and inconvenient since the user must send his/her smart card to the key information centre (KIC) to change his/her password. To resolve these problems, we propose a remote user authentication scheme using elliptic curve cryptography (ECC). It is based on timestamp and elliptic curve discrete logarithm problem. And it provides mutual authentication and is proved to resist the aforementioned attacks.

The remainder of this paper is organized as follows. In Section 2, we review the preliminaries. In Section 3, we give a brief review of Awasthi et al.'s scheme and discuss attacks against it. In Section 4, a timestamp-based mutual authentication scheme using smart card and ECC is proposed. In Section 5, we prove the security of the scheme. In Section 6, we evaluate the performance of the proposed scheme. In Section 7, we make a conclusion.

2 Preliminaries

2.1 RSA Cryptosystem

In 1983, Rivest et al. [21] proposed a public-key cryptosystem, namely RSA cryptosystem. RSA's mathematical hardness comes from the ease in calculating large numbers and the difficulty in finding the prime factors of those large numbers. The principle of RSA cryptosystem is described as follows:

A. To create a RSA public/private key pair, here are the basic steps:

1) Choose two large prime numbers, p and q, and calculate the modulus, $n = pq$.

2) Select a third number e that is relatively prime to the product $(p-1)(q-1)$ as the public exponent.

3) Calculate an integer d from the quotient $ed \equiv 1 \bmod [(p-1)(q-1)]$. The number d is the private exponent.

The public key is the number pair (n, e). Although these values are publicly known, it is computationally infeasible to determine d from n and e if p and q are large enough, e.g. 512 bits.

B. To encrypt a message M with RSA, one should use a public encryption key (e, n). First, represent the message as an integer between 0 and n-1. Then encrypts the message by raising it to the eth power modulo n. That is, $C = M^e \bmod n$. On the other hand, to decrypt the cipher text, raises it to another power d, again modulo n. The encryption and decryption algorithms E and D are thus:

$$E(M) \equiv M^e \bmod n$$

$$D(C) \equiv C^d \bmod n.$$

2.2 Elliptic Curve Cryptography

The ECC [2, 13, 19] presents an attractive alternative cryptosystem. It is more efficient compared with the traditional exponential cryptosystem. It can offer levels of security with small keys comparable to RSA. Since the ECC key sizes are so much shorter than comparable RSA keys, the length of the public key and private key is much shorter in elliptic curve cryptosystems. This results into faster processing times, and lower demands on memory and bandwidth.

According to [2], that the server selects elliptic curve (EC) domain parameters over F_p are sextuple:

$$T=(p, a, b, P, n, h),$$

Consisting of an integer p specifying the finite field F_p, two elements $a, b \in F_p$, specifying an elliptic curve $E(F_p)$ defined by the equation:

$$E_{a,b} : y^2 = x^3 + ax + b$$

A base point P on $E(F_p)$, a prime n which is the order of P, and an integer h which is the cofactor $h = \# E(F_p)/n$.

2.3 Definitions

Definition 1. *The integer factorization problem (IFP) is the following: given a positive integer n, find its prime factorization; that is* $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ *, where the* p_i *are pair wise distinct primes and each* $e_i \geq 1$.

Definition 2. *The ECDLP is as follows: given a point on EC Q=aP, it is hard to compute secret value a.*

Definition 3. *A secure one-way hash function y=h(x) is one where given x to compute y is easy and given y to compute x is hard.*

3 Brief Review of Awasthi et al.'s Scheme

Awasthi et al.'s scheme consists of four phases: initialization, registration, login, and authentication phases. The details of their scheme are shown as follows.

3.1 Awasthi et al.'s Scheme

3.1.1 Notations

We first define the notations which are used in the whole paper.

U_i:	the ith user
ID_i:	the identity of U_i
PW_i:	the password of U_i
S:	the remote server
n:	the modulus of RSA cryptosystem
p, q:	big prime numbers, e.g. 512bits
$E(M)$:	encrypt message M with the public key e
$D(C)$:	decrypt the cipher text C with the secret key d
x:	the secret key of the server
$Q=x \cdot P$:	the public key of the server
$h(.)$:	a strong cryptographic one-way hash function
$\|\|$:	the string concatenation operation
$\oplus$:	the exclusive-or operation
$\Rightarrow$:	a secure channel
$\rightarrow$:	a common channel
$A?=B$:	compares whether A equals B
D:	a uniformly distributed dictionary of size $\|D\|$.

3.1.2 Initialization Phase

In Awasthi et al.'s scheme, KIC is a trusted authority which generates global parameters. KIC performs the following steps:

Step 1: Generate two large primes p and q and compute $n = pq$.

Step 2: Choose a prime number e and an integer d, such that $ed \bmod (p-1)(q-1) \equiv 1$, where e is the system's public key and d is the corresponding private key, which should be kept secret by the server.

Step 3: Find an integer g, which is a primitive element in both $GF(p)$ and $GF(q)$, and is the public information of the system.

3.1.3 Registration Phase

A new user U_i securely submits his identifier ID_i and password PW_i to the KIC. The KIC then performs the following steps:

Step 1: Computes $CID_i=h(ID_i \oplus d)$, $h_i \equiv g^{PW_i \cdot d} \bmod n$, and $S_i \equiv CID_i^d \bmod n$.

Step 2: Writes n, e, g, ID_i, S_i and h_i into a smart card and issues the smart card to U_i through a secure channel.

3.1.4 Login Phase

U_i performs the following steps:

Step 1: Choose a random number r_i and compute $X_i = g^{r_i \cdot PW_i}$ and $Y_i = S_i \cdot h_i^{r_i \cdot h(ID_i, T_c)}$.

Step 2: $U_i \rightarrow S$: $M=ID_i, X_i, Y_i, n, e, g, T_c$
U_i sends the login request message M to the remote server.

3.1.5 Authentication Phase

After receiving the login request message M from U_i, S will perform the following steps to verify the correctness of M.

Step 1: Verify that ID_i is a valid user identifier. Otherwise reject the login request.

Step 2: Check the validity of T_c. If $(T_s\text{-}T_c) > \Delta T$, the server rejects the login request, where T_s is the current timestamp on S; ΔT is the expected legitimate time interval for transmission delay.

Step 3: Compute $CID_i = h(ID_i \oplus d)$ and check the equation

$$Y_i^e \equiv CID_i \cdot X_i^{h(ID_i, T_c)} \bmod n.$$

If it holds, accept the login request, otherwise reject.

Step 4: $S \rightarrow U_i$: $M'=(R', T'_s)$ where $R \equiv (h(ID_i, T'_s))^d \bmod n$ and T'_s is the current timestamp on S.

Step 5: Upon receiving the message M' from the server, U_i verifies the server as follows.

a) Check the time interval between T'_s and T'_c, where T'_c is the timestamp when the user U_i receives the message M'. If $T'_c\text{-}T'_s > \Delta T$, U_i rejects the remote server, where ΔT denotes the predetermined legitimate time interval of transmission delay.

b) Compute $R' \equiv R^e \bmod n$. If $R'= h(ID_i, T'_s)$, U_i accepts the server, otherwise rejects server and disconnects it.

3.2 Attacks on Awasthi et al.'s Scheme

In this section, we point out that Awasthi et al.'s scheme is vulnerable to the privileged insider, the lost smart card, the replay, the DoS, and the modification attacks.

3.2.1 Privileged Insider Attack

If the password of a user can be derived by the server in the registration protocol, it is called the privileged insider attack [10]. In the registration phase of the Awasthi et al.'s scheme, U_i sends his/her identity ID_i and password PW_i to S directly. The privileged insider of the server can get the user's password easily in this phase. He or she can use these passwords to access other servers with the same passwords if U_i registered himself/herself to other servers.

3.2.2 Lost Smart Card Attack

Kocher et al. [14] and Messerges et al. [20] have pointed out that all existent smart cards are vulnerable in that the confidential information stored in the device could be extracted by physically monitoring its power consumption; once a smart card is lost, all secrets in it may be revealed. Suppose that attacker Eve steals U_i's smart card. Eve then can log into the remote server successfully by performing the following steps:

Step 1: Eve gets n, e, g, hi from the lost smart card by using Kocher et al.'s or Messerges et al.'s extracting technique.

Step 2: Eve computes hie=($g^{pw_i \cdot d}$)e mod n = g^{pw_i} mod n.

Step 3: Eve selects a password candidate PW* from the dictionary D.

Step 4: Eve computes g^{PW^*} (mod n) and compares it with hie. If they are equal, Eve gets the correct password. Otherwise she goes to Step 3 and repeats this procedure until she finds the correct password. After getting the password, Eve can masquerade as U_i to log into the server successfully.

From aforementioned description, we know that Awasthi et al.'s scheme suffers from the lost smart card attack.

3.2.3 Replay Attack

Suppose Eve implants a Trojan horse software in the user's system. The software will intercept and modify the time request service message. When a session of Awasthi et al.'s protocol wants to get the timestamp T_c of the computer, the Trojan horse software intercepts this message and gets the timestamp from the system. After that, the software adds some time, e.g. twenty-four hours, to the timestamp and sends back the modified timestamp to the session. All these will be done underneath and nobody can find what has happened.

The smart card of U_i composes the message with this modified timestamp T_c and sends the message $M=(ID_i, X_i, Y_i, n, e, g, T_c)$ to the server. Eve records this message. On the other hand, after receiving the message, the server checks whether $(T_s - T_c) \leq \Delta T$. We know that the user's clock has been synchronized with the clock of the server, so T_s-T_c must be less than 0, i.e. T_s-T_c is a negative. The message $(ID_i, X_i, Y_i, n, e, g, T_c)$ passes the check successfully. Then the server sends back the response message with its timestamp to the user and the response message also passes the check successfully. All these steps go smoothly and nobody could find any errors in the procedure.

After the user leaves the system, Eve could replay the message $M=(ID_i, X_i, Y_i, n, e, g, T_c)$ at anytime within twenty-four hours. This message should pass the check of the server successfully within the limited time. The server will accept Eve as a valid user and maintains the session state waiting for Eve to proceed with the next step.

From the attack procedure mentioned above, we know that Awasthi et al.'s scheme is under the replay attack.

3.2.4 DoS Attack

In this section, we show that Awasthi et al.'s scheme is vulnerable to the DoS attack. The procedure is as follows.

Step 1: Eve registers herself as a valid user to the server and synchronizes her system clock with the server clock.

Step 2: Eve adjusts her system clock by adding twenty-four hours.

Step 3: Eve inserts her smart card into the card reader and runs the protocol. The smart card composes the message with this modified timestamp T_c and sends the message $M=(ID_i, X_i, Y_i, n, e, g, T_c)$ to the server. After receiving the message the server checks whether $(T_s - T_c) \le \Delta T$. We know that Eve's clock is twenty-four hours ahead of the server's clock, so T_s-T_c must be a negative and less than ΔT . The message $(ID_i, X_i, Y_i, n, e, g, T_c)$ passes the check successfully. After that, the server sends back the response message with its timestamp to Eve. On the receipt of the respond message, Eve simply discards it. Meanwhile, the server maintains the session state waiting for Eve to proceed with the next step.

Step 4: Now, Eve replays such requests from different machines as many as she could and the server could not detect such attacks. The resources of the server, e.g. CPU, memory, and bandwidth of the networks, will be exhausted.

From the attack procedure mentioned above, we know that Awasthi et al.'s scheme is under the DoS attack.

3.2.5 No Perfect Secrecy Property

Awasthi et al. claim that their scheme maintains perfect secrecy since nobody except S can recover $CID_i= h(ID_i \oplus d)$ from the transmitted messages. However, we will show that the attacker can easily recover U_i's CID_i during the protocol run.

Case 1:

Suppose Eve records one run of Awasthi et al.'s authentication scheme. Eve then performs the following steps to recover U_i's CID_i.

Step 1: Eve gets Xi, Yi, n, e, IDi, Tc from the recorded messages.

Step 2: Eve computes

$$CID_i= \frac{(Y_i)^e}{(X_i)^{h(ID_i, T_c)}} \bmod n$$

CID_i is a hash value and much less than the security parameter n in Awasthi et al.'s scheme. Thus CID_i which is computed above is equal to $h(ID_i \oplus d)$ in high probability. Therefore, anybody could recover this value easily. Thus Awasthi et al.'s scheme does not provide perfect secrecy property.

Case 2:

Suppose Eve gets n, e, S_i from the lost smart card. She can recover CID_i by computing $S_i^e \bmod n= (CID_i^d)^e \bmod n= CID_i \bmod n= CID_i$. In this case, Eve also can recover CID_i easily.

3.2.6 Modification Attack

Attack Eve can modify the message $M=(ID_i, X_i, Y_i, n, e, g, T_c)$ of U_i as she will. The modification attack procedure is as follows:

Step 1: Eve registers herself as a valid user to the server and gets a smart card, which contains n, e, g, ID_e, S_e and h_e, from the server.

Step 2: Eve extracts the parameters n, e, g, IDe, Se and he from her smart card.

Step 3: Eve intercepts the message M of U_i and computes

$$X'_i = X_i \cdot CID_e^{h^{-1}(ID_i, T_c)},$$
$$Y'_i = Y_i \cdot S_e.$$

After that, Eve sends the modification message $M'=(ID_i, X'_i, Y'_i, n, e, g, T_c)$ to the server. It is easy to verify that $M'=(ID_i, X'_i, Y'_i, n, e, g, T_c)$ is a valid login request. In fact,

$$\begin{aligned}(Y'_i)^e &= (Y_i \cdot S_e)^e \\ &= CID_i \cdot CID_e \cdot X_i^{h(ID_i, T_c)} \bmod n \\ &= CID_i \cdot (CID_e^{h^{-1}(ID_i, T_c)} \cdot X_i)^{h(ID_i, T_c)} \bmod n \\ &= CID_i \cdot X_i'^{h(ID_i, T_c)} \bmod n.\end{aligned}$$

From the description mentioned above, we know that Awasthi et al.'s scheme is susceptible to the modification attack.

3.3 Pitfalls in Awasthi et al.'s Scheme

It is inconvenient for a user to change his/her password since the scheme does not provide any password change mechanism for the user to change his/her password freely.

4 The Proposed Scheme

We propose a novel ECC-based authentication scheme in order to strength Awasthi et al.'s scheme in this section. Our scheme contains five phases, including system setup, registration, login, authentication, and password change phases.

4.1 System Setup Phase

All members and the server agree on EC parameters. The server selects a secret key x and computes $Q=x \cdot P$, keeps secret x and publishes the public parameters p, a, b, P, n, h, Q.

4.2 Registration Phase

Figure 1 shows the registration phase protocol of our scheme. When a user wants to log into the server, he/she must register to the remote server first. In this phase, the user communicates with the server through a secure channel. The details are described as follows.

Step 1: $U_i \Rightarrow S$: ID_i, $HPW= h(PW_i||N)$
U_i freely chooses his or her identity ID_i and password PW_i, selects a random number N, and computes $HPW= h(PW_i||N)$. After that U_i interactively sends them to S through a pre-established secure channel, such as virtual private network (VPN) or secure sockets layer (SSL).

Step 2: $S \Rightarrow U_i$: smart card
After receiving the message, S computes $V_i=h(ID_i||x) \oplus h(PW_i||N)$, stores ($V_i$, $h(.)$) in a smart card and issues the smart card to U_i through a secure channel. Finally, S maintains an ID table which contains (ID_i, status-bit).

Step 3: Upon receiving the smart card, U_i enters N into the smart card.

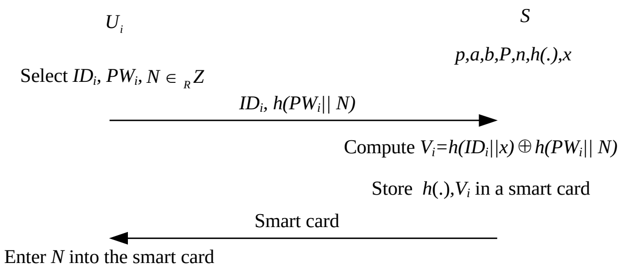

Figure 1: Registration phase

4.3 Login Phase

Figure 2 shows the login phase of our scheme. In this phase, U_i communicates with S through a common channel. When U_i wants to log into a remote server, he/she keys his or her identity ID_i and password PW_i. The smart card performs the following steps to execute the protocol.

Step 1: The smart card computes $s=V_i \oplus \mathrm{h}(PW_i||N)$. Then it selects a random nonce $r_1 \in Z_n^*$ and computes $R_1=r_1 \cdot P$, $R_2=r_1 \cdot Q$ and $V_1=\mathrm{h}(ID_i||R_1||R_2||s||T_c)$, where T_c is the timestamp at the login device.

Step 2: $U_i \rightarrow S$: $M_1= (ID_i, R_1, V_1, T_c)$
The smart card sends message $M_1= (ID_i, R_1, V_1, T_c)$ to the server S.

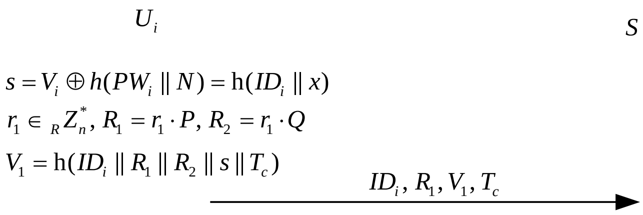

Figure 2: Login phase

4.4 Authentication Phase

After receiving the login request message M_1 from U_i, the remote server will perform the following steps to verify the correctness of M_1.

Step 1: S checks the validity of the identity ID_i. If ID_i is not in the database, S aborts the session and informs the user about it. Otherwise, S checks the status-bit of ID_i in the ID table. If the status-bit is equal to one, S will reject the login message and inform the user about it. Otherwise S sets the status-bit to one and checks the validity of T_c. If (T_s-T_c)<=0 or $(T_s-T_c) > \Delta T$, the server rejects the login request, where T_s is the current timestamp at the remote server; ΔT is the expected legitimate time interval for transmission delay. Otherwise S goes to the next step.

Step 2: $S \rightarrow U_i$: $M_2= (V_2, T'_s)$
S computes $R'_2=x \cdot R_1= x \cdot r_1 \cdot P= r_1 \cdot Q$ and $s'= h(ID_i||x)$. After getting R'_2 and s', S checks whether V_1 is equal to $h(ID_i||R_1||R'_2||s'||T_c)$. If they are not equal, S rejects the login request and informs the user about it. On the other hand, S authenticates U_i and gets the timestamp T'_s, computes $V_2=\mathrm{h}(S||ID_i||R'_2||R_1||s'|| T'_s)$, and sends the message $M_2= (V_2, T'_s)$ to the user.

Step 3: Upon receiving the message M_2 from the server, U_i checks the validity of T'_s. If $(T'_c-T'_s)$<=0 or $(T'_c-T'_s) > \Delta T$, then the server rejects the login request, where T'_c is the current timestamp at the user system; ΔT is the expected legitimate time interval for the transmission delay. Otherwise U_i goes to the next step.

Step 4: U_i checks whether V_2 is equal to $h(S||ID_i||R_2||R_1||s||T'_s)$, If they are not equal, U_i aborts the session. Otherwise, U_i authenticates the server by this message.

The server sets status-bit to zero when the session finishes.

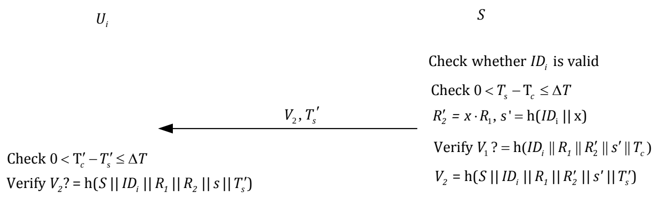

Figure 3: Authentication phase

4.5 Password Change Phase

Figure 4 shows the password change phase of our scheme. When a user doubts that his or her password has been stolen, he or she can change the password freely in this phase. U_i needs to key his or her identity ID_i and password PW_i first.

Step 1: U_i needs to go through the above login and authentication procedures and let the server authenticate him or her first with his or her old password PW_i. After receiving the successful authentication confirmation from the server, U_i inputs the new password PW^*_i.

Step 2: The smart card selects a random number N' and then computes $V'_i = V_i \oplus h(PW_i||N) \oplus h(PW^*_i||N')$. After finishing the computation, the smart card replaces V_i and N with the new values V'_i and N', respectively. Finally, the smart card sends back the successful message to U_i.

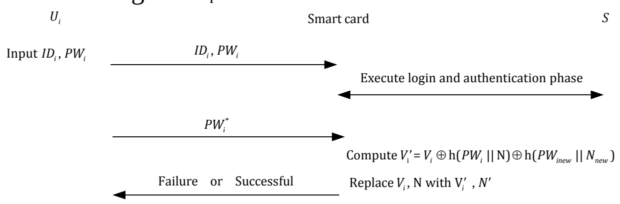

Figure 4: Password change phase

5 Security Analysis

The following propositions are used to analyze the security properties of the proposed scheme.

Proposition 1. *The proposed scheme resists the privileged insider attack.*

Proof. The privileged insider of the server cannot derive the password of the user from $h(PW_i||N)$ since N is a high entropy random number and not guessable. At the same time, the secure one-way hash function cannot be inversed. So the proposed scheme has the privileged insider attack resistance property. □

Proposition 2. *The proposed scheme resists the stolen-verifier attack.*

Proof. When the attacker Eve steals verifiers from the database of the server, she cannot get the password of the user or the secret key of the server since we stores only (ID_i, status-bit) in the server's database. So our scheme is secure against the stolen-verifier attack. □

Proposition 3. *The proposed scheme resists the lost smart card attack.*

Proof. After stealing the smart card of U_i, Eve uses Kocher et al.'s technique to extract value N and $V_i=h(ID_i||x) \oplus h(PW_i||N)$ from the smart card. She may try the dictionary attack on our scheme. The only way is as follows:

Step 1: Eve selects a candidate password PW^* from the dictionary D.

Step 2: Eve computes $s^*=V_i \oplus h(PW^*||N)$.

Step 3: In the next step, Eve wants to check the correctness of s^* by comparing it with some values. In our scheme, she only can compare it with V_1 or V_2. However, she must compute R_2 or R'_2 correctly after extracting r_1 from R_1. It is impossible since she has to solve ECDLP.

From the description above, we prove that our scheme resists the lost smart card attack. □

Proposition 4. *The proposed scheme resists the impersonation attack.*

Proof. Eve cannot impersonate the user to cheat the server, because she cannot construct the message $V_1=h(ID_i||R_1||R_2||s||T_c)$ without the knowledge of s and R_2.

Eve also cannot masquerade as the server to cheat the user, since she cannot compute the response message $R'_2= x \cdot R_1$ and $V_2=h(S||ID_i||R'_2||R_1||s'||T'_s)$ correctly without the secret key of the server. So our scheme resists the impersonation attack. □

Proposition 5. *The proposed scheme resists the replay attack.*

Proof. Eve cannot start a replay attack against our scheme because of the timestamp mechanism. We verify the timestamp by checking whether $0<(T_s-T_c) \leq \Delta T$ holds. If it does not hold, the server identifies the replay message immediately and rejects the login request. If Eve wants to launch the replay attack successfully, she must compute and modify $V_1=h(ID_i||R_1||R_2||s||T_c)$ correctly. But she only knows ID_i, R_1 and T_c, and she cannot compute $R_2= r_1 \cdot Q$ because she has to extract r_1 from R_1 first. Again it is impossible since she must face ECDLP. Meanwhile, she does not know the secret value s. Thus, Eve cannot launch the replay attack in our scheme. □

Proposition 6. *The proposed scheme resists the password guessing attack.*

Proof. Here we only consider the off-line password guessing attack, since the online dictionary attack can be easily detected and confined by checking the correctness of V_1 and V_2.

We only use the password to calculate s in the login phase. If Eve wants to launch the password guessing attack, she has to steal the smart card from the user first. In this case, Eve cannot get the user's password. The details could be referred to Proposition 3.

From the aforementioned description, our scheme is proved to resist the password guessing attack. □

Proposition 7. *The proposed scheme resists the modification attack.*

Proof. Eve cannot modify the message $M_1= (ID_i, R_1, V_1, T_c)$ and $M_2= (V_2, T_s)$, because the user and the server always detect them by checking the correctness of V_1 and V_2, respectively. □

Proposition 8. *The proposed scheme resists the oracle attack.*

Proof. There is not decryption oracle in our scheme and our scheme resists the oracle attack. □

Proposition 9. *The proposed scheme resists the man-in-the-middle attack.*

Proof. The password of U_i and the secret key of S are used to resist the man-in-the-middle attack. Eve cannot pretend to be U_i to cheat the server, since she does not obtain the password of the user or the secret key of the server. On the other hand, Eve also cannot masquerade as the server to cheat the user. □

Proposition 10. *The proposed scheme resists the DoS attack.*

Proof. Firstly, Eve cannot start the DoS attack during the password change phase since our scheme changes the user's password locally in the smart card. At the same time, we check the correctness of the old password by executing the login and authentication phase first. Secondly, we set the status-bit to maintain only one session per user. Therefore, our scheme resists the DoS attack. □

Proposition 11. *The proposed scheme provides mutual authentication.*

Proof. Mutual authentication means that both the user and the server are authenticated to each other within the same protocol. The server can authenticate the user by checking whether V_1 is correct since only the valid user can construct the request message correctly. The user U_i can authenticate the identity of the server if V_2 is correct, since only the server can make a correct response to the user's challenge. □

6 Performance Analysis and Comparison

6.1 Performance Analysis

We focus on the computation cost of the login and authentication phases since they are the main body of the authentication scheme. Since exclusion-or operation requires very low computation costs, it is usually neglected. The most processor-hungry computation is modular exponentiation, it consumes more time than the elliptic curve scale multiplication operation.

Table 1 shows the main computation cost of our scheme. We define the notation PM and H as the time complexity for elliptic curve scale multiplication and hash function operation, respectively. Computation costs of the user and the server are 2PM+ 3H and 1PM+ 3H, respectively. The total cost is 3PM + 6H. And during the protocol run, we need to send 1 identity, 2 hash values, 2 timestamp values, and an element of additive group G. The total communication cost is about 5×128+ 160=800 bits.

6.2 Comparison

The security property comparison of previous related schemes [1, 22, 23] and our scheme is summarized in the Table 2.

Table 1: Computation cost and communication cost of the login and authentication phases

	Computation cost	Communication cost
User side	2PM+ 3H	
Server side	1PM+ 3H	
Total	3PM+ 6H	5×128+ 160

Total computation cost is 3 PM+ 6 H and the total communication cost is 800 bits.
Assumed identity, hash value, and the timestamp are the same length, 128 bits; an element of additive group G is 160 bits.

Table 2: Comparison of security properties

	Yang [23]	Shen [22]	Awasthi [1]	Ours
A1	IS	IS	IS	S
A2	IS	IS	IS	S
A3	IS	IS	IS	S
A4	IS	IS	IS	S
A5	IS	IS	IS	S
A6	IS	IS	IS	S
P1	NP	P	P	P

P-provided; NP-not provided; IS- insecure; S-secure
A1-privileged insider attack; A2-lost smart card attack; A3-password guessing attack; A4-replay attack; A5-forged login attack; A6-modification attack; P1- mutual authentication;

Table 3: Comparison of computation cost

	Yang [23]	Shen [22]	Awasthi [1]	Ours
No. of Exp	4	6	5	0
No. of PM	0	0	0	3
No. of MM	2	2	2	0
No. of H	2	5	5	6
Communication cost (bits)	5504	6656	6528	800
Security	IFP	IFP	IFP	ECDLP

Exp-modular exponentiation; MM-modular multiplication; PM-scale multiplication; H-hash function
Suppose identity, timestamp and hash length are the same length, 128 bit; RSA parameter is 1024 bit; ECC parameter is 160 bit

The computational cost of our proposed authentication scheme and that of the previous related schemes [1, 22, 23] are summarized in Table 3.

As shown in the Table 3, Yang et al.'s scheme needs 4 modular exponentiations, 2 modular multiplications, and 2 hash functions. Meanwhile, it is only a unilateral authentication scheme. Shen et al.'s scheme needs to compute 6 modular exponentiations, 2 modular multiplications, and 5 hash functions. Awasthi et al.'s scheme needs 5 modular exponentiations, 2 modular multiplications, and 5 hash functions. Meanwhile, all these timestamp-based authentication schemes are vulnerable to the privileged insider, lost smart card, dictionary, replay, modification, and DoS attacks. However, our scheme only needs 3 elliptic curve scale multiplications and 6 hash function operations.

In summary, compared with these protocols, the communication cost of our scheme is much lower. At the same time, our scheme is proved to be secure against aforementioned attacks. We can make a conclusion that our scheme is more secure and efficient than these timestamp-based authentication schemes.

7 Conclusions

We have demonstrated the vulnerability of Yang et al.'s, Shen et al.'s, and Awasthi et al.'s schemes to the privileged insider, lost smart card, replay, modification, and DoS attacks in this paper. In order to overcome the shortcomings and improve the efficiency of these schemes, we propose a robust and efficient timestamp-based remote user mutual authentication scheme based on smart card and ECC. Our new scheme resists the above attacks and only needs to compute 3 elliptic curve scale multiplications and 6 hash function operations during a protocol run. The new scheme has been proved to be more secure and efficient than the aforementioned timestamp authentication schemes.

References

[1] A. K. Awasthi, K. Srivastava and R. C. Mittal, "An improved timestamp-based remote user authentication scheme," *Computers and Electrical Engineering*, vol. 37, no. 6, pp. 869-874, 2011.

[2] Certicom research, "Standard for efficient cryptography, sec 1, 2000: EC Cryptography," Ver. 1.0, 2000.

[3] C. K. Chan and L. M. Cheng, "Cryptanalysis of timestampbased password authentication scheme," *Computer & Security*, vol. 21, no. 1, pp. 74-76, 2002.

[4] C. C. Chang and W. Y. Liao, "A remote password authentication scheme based upon elgamal's signature scheme," *Computer & Security*, vol. 13, no. 2, pp. 137-144, 1994.

[5] K. M. Cheng, T. Y. Chang and J. W. Lo, "Cryptanalysis of security enhancement for a modified authenticated key agreement protocol," *International Journal of Network Security*, vol. 11, no. 1, pp. 55-57, 2010.

[6] L. Fan, J. H. Li, and H. W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Computers and Security*, vol. 21, no. 7, pp. 665-667, 2002.

[7] D. B. He, J. H. Chen and J. Hu, "Weaknesses of a remote user password authentication scheme using smart card, " *International Journal of Network Security*, vol. 13, no. 1, pp. 58-60, 2011.

[8] D. B. He, S. H. Wu and J. H. Chen, "Note on 'Design of improved password authentication and update scheme based on elliptic curve cryptography'," *Mathematical and Computer Modelling*, vol. 55, no. 3-4, pp. 1661-1664, 2012.

[9] D. B. He, J. H. Chen and R. Zhang, "An efficient and provably-secure certificateless signature scheme," International Journal of Communication Systems, vol. 25, no. 11, pp. 1432-1442, Nov. 2012.

[10] D. B. He, J. H. Chen and R. Zhang, "A more secure authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 36, no. 3, pp. 1989-1995, 2012.

[11] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, 2000.

[12] W. Juang, "Efficient password authenticated key agreement using smart card," *Computer & Security*, vol. 23, pp. 67-173, 2004.

[13] N. Koblitz, "Elliptic curve cryptosystems," *M athematics of Computation*, vol. 48, pp. 203-209, 1987.

[14] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," in *Proceedings of Advances in Cryptology*, Santa Barbara, CA, U.S.A., 1999.

[15] W. Ku and S. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50 no. 1, pp. 204-207, 2004.

[16] M. Kumar, "An enhanced remote user authentication scheme with smart card," *International Journal of Network Security*, vol. 10, no. 3, pp. 175-184, 2010.

[17] M. Kumar, M. K. Gupta and S. kumari, "An improved efficient remote password authentication scheme with smart card over insecure networks," *International Journal of Network Security*, vol. 13, no. 3, pp. 167-177, 2011.

[18] J. Y. Liu, A. M. Zhou and M. X. Gao, " A new mutual authentication scheme based on nonce and smart cards," *Computer Communications*, vol. 31, pp. 2205-2209, 2008.

[19] A. J. Menezes, P. C. Oorschot and S. A. Vanstone, *Handbook of Applied Cryptograph*, Crc press, New York, 1997.

[20] T. S. Messerges, E. A. Dabbish and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541-552, 2002

[21] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM - Special 25th Anniversary Issue*, vol. 26, no. 1, DOI:10.1145/357980.358017, 1983.

[22] J. J. Shen, C. W. Lin and M. S. Hwang, "Security enhancement for the timestamp-based password authentication," *Computers and Security*, vol. 22, no. 7, pp. 591-595, 2003..

[23] W. H. Yang and S. P. Shieh, "Password authentication scheme with smard cards," *Computers and Security*, vol. 18, no. 8, pp. 727-733, 1999.

Hong-Bin Tang received his MS degree in School of Mathematics and Statistics from Wuhan University in 2006 and BS degree from Huazhong University of Science and Technology in 1996. He is currently a Ph.D. candidate at School of Computer Science and Engineering of University of Electronic Science and Technology of China. His research interests include distributed system, authentication protocol, and cryptography.

Xin-Song Liu is a professor at University of Electronic Science and Technology of China. His research interests include distributed operating system and distributed database.

Lei Jiang received his master of engineering degree in 2009. He is currently an engineer in Sichuan Provincial People's Government Society Development Research Center. His research interests include information security and cryptography.