Secure and Efficient Smart Card Based Remote User Password Authentication Scheme

Jianghong Wei, Wenfen Liu and Xuexian Hu
(Corresponding author: Jianghong Wei)

State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou, Henan Province 450002, China
(Email: jianghong.wei.xxgc@gmail.com)
(Received Dec. 6, 2014; revised and accepted Feb. 10 & Mar. 23, 2015)

Abstract

In distributed systems, the smart card based password authentication, as one of the most convenient and efficient two-factor authentication mechanisms, is widely used to ensure that the protected services are not available to unauthorized users. Recently, Li et al. demonstrated that the smart card based password authentication scheme proposed by Chen et al. cannot provide perfect forward secrecy as they claimed. In addition, the password change phase of the scheme is unfriendly and inefficient. Subsequently, Li et al. presented an enhanced smart card based password authentication scheme to overcome the above flaws existing in Chen et al.'s scheme. Furthermore, Kumari and Khan, and Jiang et al. demonstrated that Chen et al.'s scheme cannot resist off-line password guessing attacks, and also proposed an improved scheme, respectively. In this study, we first illustrate that Li et al.'s scheme, and Kumari and Khan's scheme both fail to achieve the basic security requirement of the smart card based password authentication, namely, once the private information stored in the smart card has been extracted, the schemes would be vulnerable to off-line password guessing attacks. We also point out that Jiang et al.'s scheme, as well as Kumari and Khan's scheme cannot provide perfect forward secrecy. Then, we introduce a new smart card based password authentication scheme. By presenting concrete analysis of security and performance, we show that the proposed scheme cannot only resist various well-known attacks, but also is more efficient than other related works, and thus is feasible for practical applications.

Keywords: Password, remote access, smart card, two-factor authentication

1 Introduction

Owing to information technology rapid progression, more and more resources are distributed in the form of network services provided and managed by servers in distributed systems. Remote user authentication schemes are used to ensure that these protected services are not available to unauthorized users. Most of early authentication mechanisms [1, 15, 18, 23] are solely based on the password. In these schemes, the remote server maintains a table to record the information about each user's password, and exploits it to verify the privilege of the corresponding user. However, while widely implemented in many real life applications (e.g., private corporations, banking systems, database management systems), password authentication schemes will inescapably suffer from several attacks, such as dictionary attacks, password table tampering, etc.

To conquer these attacks and improve the system security, Chang and Wu [2] introduced smart card based password authentication scheme, which has become one of the most convenient and commonly used two-factor authentication mechanisms. In the context of the smart card based password authentication scheme, each user possesses a password easy to remember and a smart card, which is issued by the remote server, and used to store some private data. The password and smart card of each user are bonded together by the remote server, that is, once successful mutual authentication requires the user to provide the correct password and corresponding smart card simultaneously. In order to evaluate the security of smart card based password authentication scheme, Xu et al. [24] suggested that there should be two assumptions of the adversary's capabilities explicitly made in this kind of authentication scheme:

1) The adversary has total control over the communication channel between the users and the remote server in the authentication phase, which means the adversary can intercept, insert, delete, or modify any message transmitted in the channel.

2) The adversary may either steal a user's smart card and then extract the information from it by the method introduced by Kocher et al. [13] and

Messerges et al. [19], or obtain a user's password, but not the both.

In fact, the first assumption is exactly the Dolev-Yao Threat Model [6], which has been widely accepted as the standard threat model for cryptographic protocols. The second assumption characters the basic security requirement of two-factor authentication scheme, that is, as long as the private information of the two authentication factors have not been disclosed simultaneously, the scheme should be still secure. This is also why the two-factor authentication scheme is more secure than the single-factor authentication scheme. The above two assumptions, which can also be considered as a security model for the smart card based password authentication scheme, have been widely approved, and the security analyses of the authentication schemes [3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 16, 17, 20, 21, 22, 24] are all based on them.

Since the introduction of smart card based password authentication, it has attracted many researcher's attention, and a lot of such schemes have been presented. However, most of them are flawed. Such examples are that, Xu et al.'s [24] scheme suffers from impersonation attacks, Das's [5] scheme is vulnerable to gateway node by-passing attack and privileged-insider attack.

Most recently, Chen et al. [4] illustrated that the schemes proposed by Song [20] and Sood et al. [21] still have various security flaws being ignored, and then proposed a robust smart card based remote user password authentication scheme. They claimed that their scheme can resist various attacks and provide perfect forward secrecy. However, Li et al. [17] pointed out that Chen et al.'s [4] scheme fails to ensure forward secrecy, and the password change phase of the scheme is unfriendly and inefficient. To overcome the problems mentioned above, Li et al. also introduced an enhanced smart card based remote user password authentication scheme. Furthermore, Kumari and Khan[14], as well as Jiang et al. [11] demonstrated that Chen et al.'s [4] scheme is even insecure against off-line password guessing attacks, and provided an improved scheme, respectively.

In this paper, we will demonstrate that Li et al.'s [17] scheme, and Kumari and Khan's [14] scheme are not secure under the assumptions (1) and (2). Specifically, the adversary can launch off-line password guessing attacks once the private data stored in the smart card have been extracted by the adversary. In addition, we point out that Kumari and Khan's [14] scheme is not correct in some case, and cannot provide perfect forward secrecy. We also note that Jiang et al.'s [11] scheme cannot provide perfect forward secrecy and friendly password change, since it inherits the main body of Chen et al.'s [4] scheme. Furthermore, to conquer these attacks and drawbacks, we propose a new smart card based password authentication scheme. Our scheme is not only secure against various well-known attacks (e.g., off-line password guessing attack, impersonation attack, replay attack, etc.) under the assumptions (1) and (2), but also is more efficient than previous schemes without loosing necessary security properties (e.g., forward secrecy, mutual authentication etc.).

The remainder of the paper is structured as follows: we provide review and cryptanalysis of Li et al.'s [17] scheme and Kumari and Khan's [14] scheme in Section 2 and Section 3, respectively. And then a secure and efficient smart card based remote user password authentication scheme is proposed in Section 4. Section 5 discusses the performance and security of our proposal. Finally, we conclude in Section 6.

2 Review and Cryptanalysis of Li et al.'s Scheme

In this section, we first briefly review the remote user authentication scheme proposed by Li et al. [17], and then demonstrate that their scheme is vulnerable to off-line password guessing attack by presenting the concrete attack process. For convenience, we summarize the notations used throughout this paper in Table 1.

2.1 Review of Li et al.'s Scheme

Li et al.'s [17] scheme consists of four phases: initialization, registration, authentication, password change. The details of the scheme are given as follows.

2.1.1 Initialization Phase

To initialize, the remote server S selects large prim numbers p and q such that $p = 2q + 1$. S also chooses a random number $x \in Z_q^*$ as its master secret key, as well as a secure one-way hash function $h(\cdot) : \{0,1\}^* \rightarrow Z_p^*$.

2.1.2 Registration Phase

When a user U_i wants to register to become a legal user, he/she first selects a password PW_i and unique identity ID_i. Then, the registration procedure proceeds as follows:

1) U_i submits the registration request message $\{ID_i, PW_i\}$ to the remote server via a secure channel.

2) Upon receiving the request message, S computes $A_i = h(ID_i||PW_i)^{PW_i} \bmod p$ and $B_i = h(ID_i)^{x+PW_i} \bmod p$.

3) S stores $\{A_i, B_i, p, q, h(\cdot)\}$ into a smart card, and issues the smart card to U_i via a secure channel.

2.1.3 Authentication Phase

When a legal user U_i wants to login into the server for acquiring some services, he/she first attaches the smart card to a device reader, and inputs his/her identity ID_i and password PW_i. Then, the authentication mechanism performs as follows:

Table 1: The notations used throughout this paper

Symbol	Description
U_i	The ith user
S	The remote server
$\mathcal{A}$	The adversary
ID_i	The user U_i's identity
PW_i	The user U_i's password
x	The master secret key of the remote server
p and q	Two large prime numbers such that $p = 2q + 1$
T	The tiemstamp
ΔT	The maximum transmission delay
T_e	The running time for once modular exponentiation operation
T_m	The running time for once modular multiplication/inverse operation
T_h	The running time for once hash operation
T_s	The running time for once symmetric encryption/decryption operation
$h(\cdot)$	A secure one-way hash function
Z_q	The ring of integers modulo q
Z_q^*	The multiplicative group of Z_q
$\|\|$	The concatenation operation

1) The smart card computes $A_i' = h(ID_i||PW_i)^{PW_i} \bmod p$, and checks that whether A_i' is equal to A_i stored in the smart card. If not, the smart card terminates the session. Otherwise, the smart card performs the following steps.

2) The smart card chooses a random number $\alpha \in Z_q^*$, and then computes:

$$\begin{aligned} C_i &= B_i/h(ID_i)^{PW_i} \bmod p, \\ D_i &= h(ID_i)^{\alpha} \bmod p, \\ M_i &= h(ID_i||C_i||D_i||T_i), \end{aligned}$$

where T_i is the current timestamp. Finally, the smart card sends the authentication request message $\{ID_i, D_i, M_i, T_i\}$ to the remote server S.

3) Upon receiving the authentication request message, S checks if the identity ID_i is valid and $T_i' - T_i \leq \Delta T$, where T_i' is the current timestamp. If either or both are invalid, S rejects the authentication request.

4) By use of the received authentication request message, S first computes:

$$C_i' = h(ID_i)^x, \text{ and } M_i' = h(ID_i||C_i'||D_i||T_i).$$

Then, S compares M_i' with M_i. If they are equal, the user U_i is authenticated by the remote server S. Otherwise, S rejects the authentication request.

5) If the user is authenticated by the remote server S, the server first chooses a random number $\beta \in Z_q^*$, and computes $V_i = h(ID_i)^{\beta} \bmod p$. Then S sets the shared session key as $sk = D_i^{\beta} \bmod p$. Finally, S gets the current timestamp T_S, computes $M_S = h(ID_i||V_i||sk||T_S)$, and then sends the response message $\{ID_i, V_i, M_S, T_S\}$ to U_i.

6) On receiving the response message, U_i checks ID_i and compares T_S' with T_S, where T_S' is the time that the response message is received. If ID_i is valid and $T_S' - T_S \leq \Delta T$, U_i computes:

$$sk' = V_i^{\alpha} \bmod p,\ M_S' = h(ID_i||V_i||sk'||T_S).$$

Then, U_i checks that whether M_S' is equivalent to the received M_S. If not, the session is terminated. Otherwise, the remote server S is authenticated by the user U_i, and an agreed session key $sk = h(ID_i)^{\alpha\beta} \bmod p$ is shared between them.

2.1.4 Password Change Phase

When the user U_i wants to replace his/her password PW_i with a new password PW_i^{new}, he/she first inputs ID_i and PW_i into the smart card. Then, the smart card carries out the following steps:

1) The smart card computes $A_i' = h(ID_i||PW_i)^{PW_i} \bmod p$, and compares A_i' with the stored value A_i. If they are not equal, the request is rejected. Otherwise, the user is asked to input a new password PW_i^{new}.

2) After receiving the new password, the smart card computes:

$$\begin{aligned} A_i^{new} &= h(ID_i||PW_i^{new})^{PW_i^{new}} \bmod p, \\ B_i^{new} &= B_i \cdot h(ID_i)^{PW_i^{new}}/h(ID_i)^{PW_i} \bmod p. \end{aligned}$$

The smart card replaces A_i, B_i with A_i^{new}, B_i^{new}, respectively.

2.2 Cryptanalysis of Li et al.'s Scheme

Now, focus on Li et al.'s scheme [17], we present two kinds of off-line password guessing attacks once the private information stored in the smart card had been disclosed. To begin with the following discussions, by the assumption (1), we first suppose that the adversary $\mathcal{A}$ has recorded the messages $\{ID_i, D_i, M_i, T_i\}$ and $\{ID_i, V_i, M_S, T_S\}$, which are involved in some successful authentication completed between the user U_i and the server S. Then, by the assumption (2), the adversary $\mathcal{A}$ can obtain U_i's smart card, and extract the private data $\{A_i, B_i, p, q, h(\cdot)\}$ stored in the smart card by the method introduced by Kocher et al. [13] and Messerges et al. [19].

The adversary $\mathcal{A}$ launches the first kind of off-line guessing attacks as follows:

Step 1. $\mathcal{A}$ selects a candidate password PW_i^* from the dictionary space $\mathcal{D}$.

Step 2. $\mathcal{A}$ computes $A_i^* = h(ID_i||PW_i^*)^{PW_i^*} \bmod p$.

Step 3. $\mathcal{A}$ checks that whether A_i^* is equal to A_i. If yes, $\mathcal{A}$ can conclude that PW_i^* is correct. Otherwise, $\mathcal{A}$ repeats the above procedure until the correct password PW_i is yielded.

Furthermore, $\mathcal{A}$ can launch the second kind of off-line guessing attacks as follows:

Step 1. $\mathcal{A}$ selects a candidate password PW_i^* from the dictionary space $\mathcal{D}$.

Step 2. $\mathcal{A}$ computes $C_i^* = B_i/h(ID_i)^{PW_i^*} \bmod p$.

Step 3. $\mathcal{A}$ computes $M_i^* = h(ID_i||C_i^*||D_i||T_i)$. Note that if $PW_i = PW_i^*$ holds, so does $C_i = C_i^*$ and $M_i = M_i^*$.

Step 4. $\mathcal{A}$ checks that whether M_i^* is equal to M_i. If yes, $\mathcal{A}$ can conclude that PW_i^* is correct. Otherwise, $\mathcal{A}$ repeats the above procedure until the correct password PW_i is yielded.

Denote by $|\mathcal{D}|$ the number of passwords in the dictionary space $\mathcal{D}$. Then, the running time of the first attack procedure is $\mathcal{O}(T_e + T_h)$, and the running time of the second attack procedure is $\mathcal{O}(T_e + T_m + 2T_h)$. That means, regardless of which method to use, the time for the adversary to recover U_i's password is proportional to the size of the password space $\mathcal{D}$. Consequently, in practise, for a restricted password space, the adversary may recover the password in seconds on a PC.

3 Review and Cryptanalysis of Kumari and Khan's Scheme

In this section, we first briefly review the smart card based remote user password authentication scheme proposed by Kumari and Khan [14], and then provide a cryptanalysis of the scheme to demonstrate that the scheme is not correct in some case, suffers from off-line password guessing attack, and can not provide perfect forward secrecy.

3.1 Review of Kumari and Khan's Scheme

Similarly, Kumari and Khan's [14] scheme also consists of four phases, i.e., initialization phase, registration phase, authentication phase and password change phase. We briefly introduce the concrete scheme as follows.

3.1.1 Initialization Phase

For initialization, the remote server S chooses two large primes p and q such that $p = 2q+1$ and $n = pq$, and keeps p and q secret. S selects a random number $x \in Z_q^*$ as its long-term private key. S also picks up a secure one-way hash function $h(\cdot)$. In addition, S preserves a registration table R_GR to record registration information about all legal users, i.e., an unique tuple $(ID_i, T_r, x \cdot p \oplus (ID_i||T_r))$ for each registered user U_i, where T_r is the registration time of U_i.

3.1.2 Registration Phase

To become a legal user and obtain services provided by the remote server, one needs to register at S to get the corresponding privilege. The detailed registration procedure performs as follows:

1) A user U_i selects his/her identity ID_i, and submits the registration request message $\{ID_i\}$ to the remote server S through a secure channel.

2) After receiving the request message, S checks whether the received identity ID_i is in the table R_GR or not. If yes, S rejects the request; otherwise, S generates a tuple $(ID_i, T_r, x \cdot p \oplus (ID_i||T_r))$, and adds it into R_GR. Here, T_r is the timestamp that the user U_i registered to S, $\oplus$ is bitwise XOR operation.

3) S sets $A_i = h(ID_i)^{x+T_r+PW_0} \bmod n$, $B_i = (h(ID_i)^{x+T_r} \bmod n) \otimes PW_0 \otimes ID_i$, and generates a temporary identity $EID_i = E_{x+p}(ID_i||T_r)$ by encrypting ID_i and T_r with the private key $x + p$. Here, $\otimes$ is bitwise NOR operation. Then S stores $\{A_i, B_i, EID_i, n, h(\cdot), E_{key}(\cdot), D_{key}(\cdot)\}$ into a smart card, and issues the smart card to U_i through a secure channel.

4) Upon receiving the smart card, U_i chooses a new password PW_i, and replace the default password PW_0 with PW_i as described in Section 3.1.4.

3.1.3 Authentication Phase

If a registered user U_i wants to obtain the corresponding services provided by a legal remote server S, he/she needs to accomplish mutual authentication described as follows:

1) The user U_i first inserts his/her smart card to a device reader, and keys in ID_i and PW_i. Then, the smart card computes $C_i = (A_i/h(ID_i)^{PW_i}) \mod n$, $B_i^* = C_i \otimes PW_i \otimes ID_i$. The smart card checks whether B_i^* is equal to B_i. If not, the smart card terminates the authentication process; otherwise, the smart card chooses $\alpha \in Z_n^*$, and computes:

$$\begin{aligned} D_i &= h(ID_i)^{\alpha} \bmod n, \\ W_i &= C_i \cdot D_i \bmod n, \\ M_i &= h(ID_i||C_i||D_i||T_i), \end{aligned}$$

where T_i is the current timestamp. Finally, the smart card sends the authentication request message $\{EID_i, D_i, M_i, T_i\}$ to the server S through a public channel.

2) After receiving the authentication request message from U_i, the server S first gets the current timestamp T_{S1}, and checks if $(T_i - T_{S1}) > \Delta T$. If yes, S terminates the authentication process; otherwise, S gets a tuple $(ID_i||T_r)$ through decrypting EID_i with its private key $x + p$.

3) If there exists a record corresponding to the tuple $(ID_i||T_r)$ in the table R_GR, S first computes:

$$\begin{aligned} C_i^* &= h(ID_i)^{x+T_r} \bmod n, \\ W_i^* &= C_i^* \cdot D_i \bmod n, \\ M_i^* &= h(ID_i||C_i^*||D_i||W_i^*||T_i). \end{aligned}$$

Then, S checks if $M_i^* = M_i$. If not, S rejects the authentication request; otherwise, S authenticates the user U_i.

4) S acquires the current timestamp T_{S2}, and computes the session key:

$$\begin{aligned} sk &= h(W_i^*||T_{S2}), \\ EID_i^* &= E_{x+p}(ID_i||T_r||T_{S2}), \\ M_S &= E_{C_i^*}(ID_i||EID_i^*||W_i^*||T_{S2}). \end{aligned}$$

Then S sends the response message $\{M_S\}$ to the user U_i, and replaces the value $x \cdot p \oplus h(ID_i||T_r)$ with $x \cdot p \oplus h(ID_i||T_r||T_{S2})$ in R_GR.

5) After receiving the response message from the server S, the smart card first obtains the tuple $(ID_i||EID_i^*||W_i^*||T_{S2})$ by decrypting M_S with its private key C_i^*. Then, the smart card checks the validity of ID_i, the freshness of T_{S2}, and verifies if $W_i^* = W_i$, $EID_i^* = EID_i$. If all of tests are passed, the smart card authenticates the remote server S; otherwise, it puts an end to the authentication process.

6) The smart card generates the session key $sk = h(W_i||T_{S2})$, and replaces the value EID_i with EID_i^*.

3.1.4 Password Change Phase

When a user U_i wants to update his/her password, (s)he inputs ID_i and PW_i followed with a new password PW_i^{new}. Then, the smart card proceeds as follows:

1) Compute $C_i = (A_i/h(ID_i)^{PW_i}) \mod n$, $B_i^* = C_i \otimes PW_i \otimes ID_i$, and check if $B_i^* = B_i$. If not, reject the request; otherwise, compute $A_i^{new} = C_i \cdot h(ID_i)^{PW_i^{new}} \mod n$, $B_i^{new} = C_i \otimes PW_i^{new} \otimes ID_i$.

2) Replace A_i and B_i with A_i^{new} and B_i^{new}, respectively.

3.2 Cryptanalysis of Kumari and Khan's Scheme

In this section, by presenting concrete analysis and attacks, we demonstrate that Kumari and Khan's [14] scheme is not correct in some case, suffers from off-line password guessing attack once the private information stored in the smart card has been extracted by the adversary by the method introduced by Kocher et al. [13] and Messerges et al. [19], and can not provide perfect forward secrecy.

3.2.1 Correctness

In the authentication phase of Kumari and Khan's scheme, we notice that the smart card need to compute $C_i = A_i \cdot 1/h(ID_i)^{PW_i} \mod n$. However, since $n = pq$ is a composite number, in some case (i.e., $gcd(n, h(ID_i)^{PW_i}) \neq 1$), $1/h(ID_i)^{PW_i} \mod n$ does not exist, and thus the smart card can not compute C_i. Although the probability that the aforementioned case occurs is less than $1 - \frac{\varphi(n)}{n} = \frac{p+q-1}{n}$, where $\varphi(\cdot)$ is Euler function, and is negligible when p and q are large enough, the essential point is that the correctness of Kumari and Khan's scheme is not perfect.

3.2.2 Off-line Password Guessing Attack

By the assumption (1), we first suppose that the adversary $\mathcal{A}$ has intercepted an authentication request message $\{EID_i^*, D_i, M_i, T_i\}$ and the associated response message $\{M_S = E_{C_i^*}(ID_i||EID_i^*||W_i^*||T_{S2})\}$ exchanged between the user U_i and the server S. Then, by the assumption (2), the adversary $\mathcal{A}$ can obtain U_i's smart card, and extracts the data $(A_i, B_i, h(\cdot), n)$. Subsequently, $\mathcal{A}$ can launch off-line password guessing attacks as follows:

Step 1. $\mathcal{A}$ picks up a candidate identity ID_i^* and a candidate password PW_i^* from two different dictionaries $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$, respectively.

Step 2. $\mathcal{A}$ computes $C_i^* = A_i/h(ID_i^*)^{PW_i^*} \mod n$, $B_i^* = C_i^* \otimes PW_i^* \otimes ID_i^*$. Note that if $ID_i^* = ID_i$ and $PW_i^* = PW_i$, then it holds that $C_i^* = C_i$ and $B_i^* = B_i$, which means that the adversary $\mathcal{A}$ can verify the validity of ID_i^* and PW_i^* by checking if $B_i^* = B_i$.

Step 3. If $B_i^* = B_i$, $\mathcal{A}$ concludes that ID_i^* and PW_i^* are correct identity and password, respectively. Otherwise, $\mathcal{A}$ repeats the above procedure until the correct identity and password are found.

In addition, similar to the above procedure, not only B_i, but also the recorded messages M_i and M_S can be used to verify the validity of candidate password and identity. We omit the details here.

In Kumari and Khan's scheme, we notice that the identity and password are both selected by the user him/hefself, which indicates that they are values easy to remember and guess, rather than random values with high entropy. The following analysis will show that the above attack can be finished in polynomial time, which is contrary to Kumari and Khan's [14] claim that "it is not possible to guess two correct values ID_i and PW_i simultaneously in polynomial time", and thus the attack is feasible in practice.

Denote by $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ the sizes of dictionary space $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$, respectively. Since the identity and password are human-remember and guessable, we can suppose that $|\mathcal{D}_{id}| = f_{id}(\lambda)$ and $|\mathcal{D}_{pw}| = f_{pw}(\lambda)$, where $f_{id}(\cdot)$ and $f_{pw}(\cdot)$ are polynomials, and λ is some fixed parameter. Roughly evaluating, the running time of the above attack is $\mathcal{O}(T_e + 2T_m + T_h)$. Thus, the time that the adversary gets the correct identity and password is at most $f_{id}(\lambda)f_{pw}(\lambda)\cdot\mathcal{O}(T_e+2T_m+T_h) = g(\lambda)\cdot\mathcal{O}(T_e+2T_m+T_h)$, where $g(\lambda) = f_{id}(\lambda)\cdot f_{pw}(\lambda)$, and is still a polynomial. That is, the adversary can recover the identity and password in polynomial time.

3.2.3 Perfect Forward Secrecy

Perfect forward secrecy ensures that previously established session keys are still secure even if the secret values of any participant involved in an authentication scheme are disclosed. Kumari and Khan [14] assumed that the secret value p could not be disclosed, and then claimed that their scheme could provide perfect forward secrecy. In fact, to complete once authentication process of their scheme, the server is required to possess the secret values x and $x + p$ simultaneously, where $x + p$ is used to generate a new temporary identity for the user by calling a symmetric encryption scheme. This suggests that the role of $x + p$ is the same with x. Thus, when considering perfect forward secrecy of their scheme, as well as x, $x+p$ should also be revealed.

Now, we illustrate that Kumari and Khan's [14] scheme can not provide perfect secrecy when the server's secret values x and $x + p$ are allowed to disclose. Suppose the adversary $\mathcal{A}$ has recorded an authentication request message $\{EID_i, D_i, M_i, T_i\}$ and the associated response message $\{M_S\}$, then $\mathcal{A}$ obtains (ID_i, T_r) by decrypting EID_i with $x + p$, and computes $C_i^* = h(ID_i)^{p+T_r}$. Furthermore, $\mathcal{A}$ can get $(ID_i, EID_i^*, W_i^*, T_{S2})$ by decrypting M_S with C_i^*, and retrieve the corresponding session key $sk = h(W_i^*||T_{S2})$. Thus, Kumari and Khan's [14] scheme can not provide perfect forward secrecy.

4 The Proposed Scheme

To conquer the security flaws existing in the schemes of Li et al. [17] and Kumari and Khan [14], we now propose a new smart card based remote user password authentication scheme. Our proposal also makes up of four phases, i.e., initialization phase, registration phase, authentication phase and password change phase.

4.1 Initialization Phase

Initially, the remote server S selects large prime numbers p and q such that $p = 2q + 1$. S also chooses its master secret key $x \in Z_q^*$, and a secure hash function $h(\cdot) : \{0,1\}^* \rightarrow Z_p^*$.

4.2 Registration Phase

As showed in Fig.1, when a user U_i wants to register to become a new legal user, the registration procedure is performed as follows:

1) U_i selects his/her identity ID_i and password PW_i, then submits the registration request message $\{ID_i, PW_i\}$ to the server S via a secure channel.

2) Upon receiving the registration request message, S checks that whether ID_i is valid or not. If not, S rejects the demand. Otherwise, S computes $B_i = h(x||ID_i)$, $A_i = B_i + h(PW_i||ID_i)$.

3) S stores $\{A_i, p, q, h(\cdot)\}$ into a smart card, and then issues the smart card to U_i via a secure channel.

4.3 Authentication Phase

When a user wishes to login into the server S for obtaining some services, he/she first attaches his/her smart card to a device reader, and inputs ID_i and PW_i. Then the authentication procedure, as illustrated in Fig.2, proceeds as follows:

1) The smart card first computes $B_i = A_i - h(PW_i||ID_i)$, and then selects a random number $\alpha \in Z_q^*$, and computes:

$$\begin{aligned} D_i &= h(ID_i)^{\alpha} \bmod p, \\ D_i^* &= D_i + B_i, \\ M_i &= h(ID_i||D_i^*||T_i), \end{aligned}$$

where T_i is the current time. Finally, the smart card sends the authentication request message $\{ID_i, D_i^*, M_i, T_i\}$ to the server.

2) On receiving the authentication request message, S checks if ID_i is valid and $T_i' - T_i \leq \Delta T$, where T_i' is the time that the message is received. If either or both are invalid, the request is rejected. Furthermore, S checks that whether $M_i' = h(ID_i||D_i^*||T_i)$ is equal to M_i or not. If not, the request is also rejected.

Registration phase

User U_i		Server S
Chooses ID_i and PW_i.	$\{ID_i, PW_i\}$ $\longrightarrow$	Computes: $B_i = h(x\|\|ID_i)$, $A_i = B_i + h(PW_i\|\|ID_i)$.
	$\longleftarrow$ Smart card	Smart card $\leftarrow \{A_i, p, q, h(\cdot)\}$.

Figure 1: Registration phase of the proposed scheme

Authentication phase

$U_i(ID_i, PW_i,$ Smart card)		$S(x)$
Inputs ID_i, PW_i, $B_i = A_i - h(ID_i\|\|PW_i)$. Selects $\alpha \in_R Z_q^*$, computes: $D_i = h(ID_i)^\alpha \bmod p$, $D_i^* = D_i + B_i$, $M_i = h(ID_i\|\|D_i^*\|\|T_i)$.	$\{ID_i, D_i^*, M_i, T_i\}$ $\longrightarrow$	Verifies ID_i and T_i. $M_i' = h(ID_i\|\|D_i^*\|\|T_i)$?= M_i. Selects $\beta \in_R Z_q^*$, computes: $V_i = h(ID_i)^\beta$, $B_i' = h(x\|\|ID_i)$, $D_i' = D_i^* - B_i'$, $Z_i = (D_i')^\beta \bmod p$, $M_S = h(ID_i\|\|D_i'\|\|V_i\|\|Z_i\|\|T_S)$.
	$\longleftarrow$ $\{ID_i, V_i.M_S, T_S\}$	
Checks ID_i and T_S. $Z_i' = V_i^\alpha \bmod p$, $M_S' = h(ID_i\|\|D_i\|\|V_i\|\|Z_i'\|\|T_S)$?= M_S. $sk = h(ID_i\|\|D_i\|\|V_i\|\|Z_i')$ $R_i = h(ID_i\|\|D_i^*\|\|V_i\|\|Z_i'\|\|T_i^{new})$	$\{ID_i, R_i, T_i^{new}\}$ $\longrightarrow$	Verifies ID_i and T_i^{new} $R_i' = h(ID_i\|\|D_i^*\|\|V_i\|\|Z_i'\|\|T_i^{new})$?= R_i $sk' = h(ID_i\|\|D_i'\|\|V_i\|\|Z_i)$

Figure 2: Authentication phase of the proposed scheme

3) S selects a random $\beta \in Z_q^*$, and first computes $V_i = h(ID_i)^\beta \bmod p$, $Z_i = (D_i^*)^\beta \bmod p$, and then sets:

$$\begin{aligned} B_i' &= h(x||ID_i), \\ D_i' &= D_i^* - B_i', \\ M_S &= h(ID_i||D_i'||V_i||Z_i||T_S), \end{aligned}$$

where T_S is the current time. Finally, S sends the message $\{ID_i, V_i, M_S, T_S\}$ to U_i.

4) After receiving the message, the smart card checks ID_i and compares T_S with T_S', where T_S' is the time that the message is received. If ID_i is valid and $T_S' - T_S \leq \Delta T$, S computes $Z_i' = V_i^\alpha \bmod p$, $M_S' = h(ID_i||D_i||V_i||Z_i'||T_S)$. If $M_S' \neq M_S$, the session is terminated. Otherwise, the server S is authenticated by the user U_i, and the shared session key is set as $sk = h(ID_i||D_i||V_i||Z_i')$. Furthermore, U_i gets the current time T_i^{new}, and generates a response message $R_i = h(ID_i||D_i||V_i||Z_i'||T_i^{new})$, and then sends the message $\{ID_i, R_i, T_i^{new}\}$ to S.

5) Upon receiving the response message, S checks ID_i and T_i^{new}. If they are valid, S computes $R_i' = h(ID_i||D_i'||V_i||Z_i||T_i^{new})$. If $R_i' \neq R_i$, S terminates the session. Otherwise, U_i is authenticated by S, and the shared session key is set as $sk' = h(ID_i||D_i'||V_i||Z_i)$. Finally, an agreed session key $sk = sk'$ is established between the user and the server.

4.4 Password Change Phase

This phase is invoked whenever a user U_i wants to replace his/her password PW_i with a new password PW_i^{new}. The specified procedure is performed as follows:

1) U_i attaches his/her smart card to a device reader, and inputs ID_i and PW_i, followed with the new password PW_i^{new}.

2) The smart card computes $A_i^* = A_i - h(PW_i||ID_i) + h(PW_i^{new}||ID_i)$. Then the smart card replaces A_i with A_i^*.

5 Security Analysis and Performance Comparisons

In this section, we present the security and performance analysis of our proposal, and compare it with other related schemes.

5.1 Resist Off-line Password Guessing Attacks

In this kind of attack, an adversary $\mathcal{A}$ is supposed to be able to get the private data $\{A_i, p, q, h(\cdot)\}$ stored in the user U_i's smart card, where $A_i = h(x||ID_i) + h(PW_i||ID_i)$. The adversary $\mathcal{A}$ may select a candidate password PW_i^* and compute $h(PW_i^*||ID_i)$, but he/she can not exploit A_i to verify the correctness of PW_i^* if he/she does not have the master secret key x. Furthermore, $\mathcal{A}$ can get the transmitted messages $\{ID_i, D_i^*, M_i, T_i\}$, $\{ID_i, V_i, M_S, T_S\}$, $\{ID_i, R_i, T_i^{new}\}$. Note that $\mathcal{A}$ can also not exploit D_i^* and R_i, which contain the information about the password PW_i, to verify the correctness of PW_i^*, since he/she does not know the values of $D_i = h(ID_i)^\alpha \bmod p$ and $Z_i' = h(ID_i)^{\alpha\beta} \bmod p$. This also makes off-line password guessing attacks impossible for a passive attacker, who can only obtain the exchanged messages. Therefore, our scheme is secure against off-line password guessing attacks, even the private data stored in the smart card are disclosed.

5.2 Resist Replay Attacks

Replay attacks mean that the adversary interferes with a protocol run by the insertion of a message, or part of a message, that has been sent previously in any protocol run. Our scheme exploits timestamp and secure one-way hash function to guard against replay attacks during the authentication phase. Suppose that the adversary has recorded the messages $\{ID_i, D_i', M_i, T_i\}$, $\{ID_i, V_i, M_S, T_S\}$ and $\{ID_i, R_i, T_i^{new}\}$, which would be used to replay. However, note that the timestamps T_i, T_S and T_i^{new} are contained in these messages, thus the replayed messages can be quickly detected by checking these timestamps. Furthermore, if the adversary replaces the timestamps T_S and T_i^{mew} with the current timestamps, the messages cannot pass the verification of the hash function. Therefore, our proposal is secure against replay attacks.

5.3 Resist Impersonation Attacks

If the adversary wants to launch the impersonation attacks, he/she has to generate a correct value R_i, which is difficult without the knowledge of D_i and Z_i. In order to get the values D_i and Z_i, the adversary must either hold the server's secret key x (i.e., the adversary has impersonated the server), or possess the private data A_i stored in U_i's smart card and the password PW_i simultaneously. It is obvious that such impersonation attack is trivial in the above two settings. Hence, our proposal is free from impersonation attacks.

5.4 Resist Parallel Attacks

To launch this kind of attack, the adversary $\mathcal{A}$ is required to create a valid authentication message by use of these intercepted authentication messages. However, we note that the authentication request message and the corresponding response message in our scheme are different in terms of structure and associated with timestamps. In addition, our scheme exploits hash values to ensure the

Table 2: Performance comparisons with previous related works

	User side	Server side	Total
Song [20]	T_s+4T_h	T_e+T_m+4T_h	T_e+2T_s+8T_h
Sood et al. [21]	3T_e+2T_m+3T_h	2T_e+T_m+3T_h	5T_e+3T_m+6T_h
Chen et al. [4]	2T_e+2T_m+4T_h	T_e+T_m+4T_h	3T_e+3T_m+8T_h
Li et al. [17]	4T_e+T_m+4T_h	3T_e+3T_h	7T_e+T_m+7T_h
Kumari and Khan [14]	2T_e+3T_m+2T_h+T_s	T_e+T_m+2T_h+3T_s	3T_e+4T_m+4T_h+4T_s
Jiang et al. [11]	3T_e+T_m+3T_h	2T_e+3T_h	5T_e+T_m+6T_h
Ours	2T_e+6T_h	2T_e+6T_h	4T_e+12T_h

authenticity. Thus, our scheme is secure against parallel attacks.

5.5 Perfect Forward Secrecy

Similar to Li et al.'s [17] scheme, by means of the intractability of the discrete logarithm problem, our scheme can also provide perfect forward secrecy. Specifically, in the case that both the user's password and the server's master secret key are disclosed, if the adversary wants to recover a previous session key $sk = h(ID_i||D_i'||V_i||Z_i)$ which is independent of the password and the master secret key, he/she must compute $Z_i = h(ID_i)^{\alpha\beta} \bmod p$. This means that the adversary has to compute α from $D_i' = h(ID_i)^{\alpha} \bmod p$ or β from $V_i = h(ID_i)^{\beta} \bmod p$. However, the discrete logarithm problem is widely believed to be difficult. Therefore, our proposal can ensure perfect forward secrecy.

5.6 Known-key Security

Known-key security means that the corrupted session keys have no effect on the security of those uncorrupted session keys. In our proposal, the shared session key is derived from $D_i = h(ID_i)^{\alpha} \bmod p$, $V_i = h(ID_i)^{\beta} \bmod p$ and $Z_i = h(ID_i)^{\alpha\beta} \bmod p$, where α and β are *randomly* chosen from Z_q^*. Thus, for another session of which session key is derived from $D_i' = h(ID_i)^{\alpha'} \bmod p$, $V_i' = h(ID_i)^{\beta'} \bmod p$ and $Z_i' = h(ID_i)^{\alpha'\beta'} \bmod p$, α' and β' are independent of α and β, which means that $h(ID_i||D_i||V_i||Z_i)$ is also independent of $h(ID_i||D_i'||V_i'||Z_i')$. Therefore, our scheme can provide known-key security.

5.7 Mutual Authentication and Key Agreement

To achieve mutual authentication, our scheme provides a mechanism that allows the user to verify the server in Step 4 of the authentication phase, and that allows the server to verify the user by Step 5 of the authentication phase. Furthermore, after they authenticated each other correctly, a shared session key, which is derived by the user and server as a function of information contributed by each of them such that no party can predetermine the resulting value, is established among the user and server, and then is used to provide a secure channel for subsequent communications.

5.8 Performance and Functionality Comparisons

In this section, we evaluate our scheme in terms of performance and functionality, and compare it with other related schemes as summarized in Table 5.1 and Table 5.7.

Typically, time complexity associated with these cryptographic operations, i.e., modular exponentiation operation, modular multiplication/inverse operation, hash operation and symmetric encryption/decryption, can be roughly expressed as $T_e \gg T_m \gg T_s \approx T_h$. Thus, the running time of all modular exponentiation operations, which are executed by the smart card and the remote server simultaneously, accounts for the major part of the running time of the entire authentication phase. In addition, computation ability of the smart card is usually limited. Therefore, to reduce the authentication delay, the smart card (i.e., user side) should execute as few modular exponentiation operations as possible, while the essential security properties of smart card based password authentication scheme are not compromised. From this perspective, Table 5.1 shows that our scheme is more efficient than these schemes [21], [4], [17], [14] and [11], since we have

$$\begin{aligned} 2T_e + 6T_h \text{ (Ours)} &< 2T_e + 2T_m + 4T_h \text{ ([4])} \\ &< 3T_e + T_m + 3T_h \text{ ([11])} \\ &< 2T_e + 3T_m + 2T_h + T_s \text{ ([14])} \\ &< 3T_e + 2T_m + 3T_h \text{ ([21])} \\ &< 4T_e + T_m + 4T_h \text{ ([17])}. \end{aligned}$$

Besides, in the aspect of the total computation cost, our scheme is also more efficient than schemes of Sood et al. [21], Li et al. [17] and Jiang et al. [11]. Although the remote server involved in our scheme needs once additional modular exponentiation operation when compared with Chen et al.'s [4] scheme and Kumari and Khan's [14] scheme respectively, we can consider the total computation cost of our scheme to be nearly the same with the two schemes, since the remote server possesses powerful

Table 3: Functionality comparisons with previous related works

	Song [20]	Sood [21]	Chen [4]	Li [17]	Kumari [14]	Jiang [11]	Ours
Off-line password guessing attacks	No	Yes	No	No	No	Yes	Yes
Impersonation attacks	No	Yes	No	Yes	Yes	Yes	Yes
Replay attacks	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel attacks	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Forgery attacks	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Man-in-the-middle attacks	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Known-key security	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Perfect forward secrecy	No	No	No	Yes	No	No	Yes
Mutual authentication	Yes	No	Yea	Yes	Yes	Yes	Yes
Session key agreement	Yes	No	Yes	Yes	Yes	Yes	Yes
Quickly detect wrong password	No	No	No	Yes	Yes	No	No
Friendly password change	No	No	No	Yes	Yes	No	Yes
Perfect correctness	Yes	Yes	Yes	Yes	No	Yes	Yes

capacity of computation and storage, and then the time difference of once modular exponentiation operation may be ignored.

Smart card based password authentication should enjoy two-factor security, namely, even when either the private data stored in the smart card or the corresponding password are compromised (not the both), the scheme should be still secure. As illustrated in Table 3, when compared with the schemes of Song [20], Sood et al. [21], Chen et al. [4], Kumari and Khan [14], and Li et al. [17], only our scheme can resist password guessing attacks when the private data stored in the smart card is disclosed. Although schemes of Sood et al. [21] and Jiang et al. [11] are also free from off-line password guessing attacks, nevertheless they cannot provide perfect forward secrecy and friendly password change. We also note that when compared with other schemes, only the correctness of Kumari and Khan's scheme is not perfect, since a composite number is used as the modular number in their scheme.

The essential point is that Li et al.'s [17] scheme and Kumari and Khan's [14] scheme enjoy the functionality of quickly detecting wrong password through storing the verification information about the corresponding password into the smart card. However, as indicated by off-line password guessing attacks presented in Section 2.2 and Section 3.2, once the private information stored in the smart card has been disclosed, the adversary would exploit the verification information to check the validity of each candidate password, and launch off-line password guessing attacks. Thus, we suggest that the smart card should not contain any information that can be directly used to verify the validity of the corresponding password. Nevertheless, when the smart card can not detect the wrong password, which is the case in our scheme, inputting wrong password will produce one round additional communication between the user and the remote server, since only the remote server can check the correctness of the password.

6 Conclusions

In this study, we first examined the smart card based password authentication schemes proposed by Li et al. [17] and Kumari and Khan [14], respectively. Our cryptanalysis showed that the schemes would be vulnerable to off-line password guessing attacks once the private information stored in the smart card has been disclosed. In addition, we also pointed out that Kumari and Khan's [14] scheme cannot provide perfect forward secrecy and perfect correctness. Subsequently, to overcome the defects existing in the above two schemes, we proposed a new smart card based password authentication scheme. By presenting the concrete analysis of security and performance, we demonstrated that our proposal is not only free from various well-known attacks, but also is more efficient than other previous related works. Thus, our scheme is more feasible for practical applications.

Acknowledgements

This work was supported in part by the National Key Basic Research Program (973 program) under Grant 2012CB315905, in part by the National Nature Science Foundation of China under Grant 61379150, and in part by Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-14-004.

References

[1] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology (Eurocrypt'00)*, pp. 139–155, Springer, 2000.

[2] C. C. Chang and C. S. Laih, "Remote password authentication with smart cards," *IEE Proceedings E: Computers and Digital Techniques*, vol. 138, no. 3, pp. 165-168, 1992.

[3] K. Chatterjee, A. De, and D. Gupta, "Mutual authentication protocol using hyperelliptic curve cryptosystem in constrained devices," *International Journal of Network Security*, vol. 15, no. 1, pp. 9–15, 2013.

[4] B. L. Chen, W. C. Kuo, and L. C. Wuu, "Robust smart-card-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 27, no. 2, pp. 377–389, 2014.

[5] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[6] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[7] T. H. Feng, C. H. Ling, and M. S. Hwang, "Cryptanalysis of Tan's improvement on a password authentication scheme for multi-server environments," *International Journal of Network Security*, vol. 16, no. 4, pp. 318–321, 2014.

[8] D. Giri, T. Maitra, R. Amin, and P. D. Srivastava, "An efficient and robust rsa-based remote user authentication for telecare medical information systems," *Journal of Medical Systems*, vol. 39, no. 1, pp. 1–9, 2015.

[9] M. S. Hwang, S. K. Chong, and Te-Yu Chen, "Dos-resistant id-based password authentication scheme using smart cards," *Journal of Systems and Software*, vol. 83, no. 1, pp. 163–172, 2010.

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

[11] Qi Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383–393, 2015.

[12] W. S. Juang and J. L. Wu, "Two efficient two-factor authenticated key exchange protocols in public wireless lans," *Computers & Electrical Engineering*, vol. 35, no. 1, pp. 33–40, 2009.

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology (CRYPTO'99)*, pp. 388–397, Springer, 1999.

[14] S. Kumari and M. K. Khan, "Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 27, no. 12, pp. 3939–3955, 2014.

[15] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[16] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol.," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[17] X. Li, J. Niu, M. K. Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1365–1371, 2013.

[18] I-En Liao, C. C. Lee, and M. S. Hwang, "A password authentication scheme over insecure networks," *Journal of Computer and System Sciences*, vol. 72, no. 4, pp. 727–740, 2006.

[19] 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.

[20] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.

[21] S. K. Sood, A. K. Sarje, and K. Singh, "An improvement of Xu et al.'s authentication scheme using smart cards," in *Proceedings of the Third Annual ACM Bangalore Conference*, p. 15, 2010.

[22] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[23] T. D. Wu, "The secure remote password protocol," in *Network & Distributed System Security*, vol. 98, pp. 97–111, 1998.

[24] J. Xu, W. T. Zhu, and D. G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

Jianghong Wei received the B.S. degree in Information Security from Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2009. He is currently a PhD student in State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China. His research interests include applied cryptography and network security.

Wenfen Liu received the PhD degree in Mathematics from Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 1995. She is a full professor in the State Key Laboratory of Mathematical Engineering and Advanced Computing, and severs as head of probability statistics. Her research interests include probability statistics, network communications and information security.

Xuexian Hu is a lecturer in the State Key Laboratory of Mathematical Engineering and Advanced Computing. He received the PhD degree in Information Security from Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2009. His current research interests include applied cryptography, network security.