Efficient Two-Factor Authentication Protocol Using Password and Smart Card

Fenghua Liu
School of Mathematics and Physical Science of Xuzhou Institute of Technology,
221008, Jiangsu Province, China
E-mail:liufenghua@xzit.edu.cn

***Abstract*—Two-factor authentication using password and smart card could reduce the risk than the use of a password alone. Recently, Chen et al. proposed a two-factor remote user authentication protocol using password and smart card and provide the criteria of authentication protocols. They claimed their protocol is secure against certain known attacks. In this paper, the authors showed that Chen et al.'s scheme is still vulnerable to the off-line password guessing attack, privileged administrator attack, key control attack and lacks of forward security. To solve these security problems, we propose an efficient two-factor authentication and key agreement protocol..**

***Index Terms*—authentication, smart card, forward secrecy, off-line password guessing attack, Privileged administrator**

I. INTRODUCTION

In today's digital world, password authentication is the most simple and effective method for authentication in cyberspace [1]. Lamport [2] designed the first well-known password authentication protocol by employing a one-way hash chain. However, the remote server needs to maintain a password verification table for verifying the users. When more and more clients register to the server, the verification table will become a big burden to the system and hard to maintain. Another weakness is that the verification table may be modified, corrupted or stolen if an adversary breaks into the server.

Millions users are at risk of falling victim in the password authentication scheme. The password can be compromised by hardware keyboard logger, guessing, shoulder surfing and social engineering. Large number tools provide adversaries with easy ability to trap, extort, hack, or crack the users' passwords. When the password is disclosed, the adversary can impersonate the user to login the server.

As a remedy to the above problems, two-factor authentication protocol using password and smart card have been proposed. Two-factor authentication means that one entity authenticates another entity with two authentication factors. Two-factor authentication using password and smart card is more secure than the use of a password alone. However, many proposed schemes tend to be vulnerable to some attacks if a holder lost his smart card and an adversary can get the secret information of the smart card. Kocher et al. [3] and Messerges et al. [4] stated that the secret keys stored in smart cards could be extracted by monitoring the power consumption and analyzing the leaked information in the smart cards, respectively. In designing smart card based password authentication protocol, we should take into account stolen smart card problem.

In a real life, a user always chooses the same password to register with different application servers. The privileged administrator can obtain the user's password from the server. We should not ignore that the privileged administrator can impersonate the user to access the resource of other application servers [5].

Here, we list the known attacks on two-factor authentication protocols using password and smart card.

Impersonation attack: The adversary masquerades as the legal entity to communicate with another entity.

Man-in-the-middle attack: The adversary sites between entity A and entity B and makes them believe that they are communicating directly to each other, when in fact the entire conversation is controlled by the adversary.

Replay attack: The adversary intercepts the transmitted message and then re-sends it later.

Password guessing attack: Many users use short easy-to-remember passwords, such as the phone number, the email address, the birthday and the names of their children because the users are worried about forgetting their passwords. However, these easily-rememberable passwords are easy-to-guess for the adversary. The adversary can enumerate password from the password dictionary. Password guessing attack gives the adversary a chance to derive a password from verification table stored in the server or authentication data transmitted in a protocol run [6]. Password guessing attack can be classified as on-line dictionary attack or off-line dictionary attack. On-line guessing attack can be prevented by using CAPTCHAs [7]. It is desirable to design password-based protocol that withstands the off-line guessing attack and to limit the adversary only to the on-line guessing attack.

Stolen smart card problem: There are two cases. (1) The adversary could steal the user's smart card and

Project supported by The National Natural Science Funds of China (Grant NO.31270577)

doi:10.4304/jcp.8.12.3257-3263

extract the secret data stored in the smart card using physical methods. In this case, the user could revoke the lost smart card and re-apply a new card in registration phase. In some protocols, the adversary can employ this extracted information to launch off-line guessing attack or the impersonation attack. (2) The adversary steals the user's smart card temporarily and then returns it to the user. Since the adversary obtains the smart card only for a short period of time, we assume that the adversary has not enough time to extract the data stored in the smart card. In some protocols, the adversary can change the password and result in password de-synchronization between the user and the smart card. After the smart card is returned to the user, the user will be rejected by the server.

Stolen-verifier attack: In some password protocols, the server stores clear text passwords or hashed passwords. In this case, if the adversary obtains the password or password verifier and then he can use it directly to sign into the server.

Denial of service attack (DoS attack): In some password authentication protocols, the password de-synchronization attack in the password change phase can prevent the user from being authenticated by the server. There are two password de-synchronization cases. One is between the user and the smart card, the other is between the user/smart card and the server.

The two-factor authentication protocols using password and smart card should satisfy the following essential security requirements.

Mutual authentication: The user can authenticate the server, and vice versa. The adversary may be an outsider (unregistered user) or an insider (registered user). Mutual authentication can prevent the adversary from impersonating a valid user to access the resources of the server and vice versa.

Key secrecy: Only the authenticated participants know the session key.

Known-key security: In practical, the old session key can be compromised, either through user's negligence or the adversary's brute-force attack. The compromise of one session key should not allow an intruder to find out the keys of other sessions.

Perfect forward secrecy: If the adversary obtains all the long-term secret keys of the participants, the adversary could not able to derive the previously established session keys.

Privileged administrator resilience: Study reveals that many users use the same password just for the sake of convenience when they login to their email, shopping, instance message system, banking and social networking sites. Once one password has been compromised, the adversary will be able to gain access to the other accounts. Privileged administrator resilience means that the server's administrator should not be capable of possessing users' passwords.

Key control resilience: The session key should be established by both participants' key materials. Neither the user nor the server can predict the value of the session key.

Freely chosen password: Passwords are created either by the server or by the user. Randomly chosen passwords are usually hard-to-remember. In most computer system, humans are to choose a password. Freely chosen password means that the user should be able to choose or change his password freely.

Two-factor security: The two-factor authentication protocol is secure even if one authentication factor is compromised. The two-factor authentication protocol using smart card and password should prevent the adversary from joining to the server even if the adversary obtains the smart card or the password. Note that the adversary may either steal a user's smart card and then extract the information from it, or obtain a user's password, but not both.

Availability: A legal user should be able to login the server if he wants to gain access to the server.

A. Related Works

In 2000, Hwang and Li [8] proposed a two-factor authentication based password and smart card to overcome the pitfalls of Lamport's protocol. Since then, numerous smart card based password authentication protocols have been proposed. In 2009, Xu et al. [9] showed that [10][11] are vulnerable to the forgery attacks provided that the adversary steals the smart card. Xu et al. also suggested an improved protocol. However, [9] is vulnerable to internal impersonation attacks [12] and off-line dictionary attacks [13]. Recently, Chen et al. [14] found that [12] cannot resist stolen-smart-card and off-line guessing attacks and [13] does not achieve mutual authentication. To eliminate these flaws in [12][13], Chen et al. presented an improved two-factor protocol [14]. The protocols [9-14] use timestamp and require clock synchronization to resist replay attack. In this paper, we show that Chen et al.'s two-factor protocol doesn't achieve the desired security objectives.

B. Contributions

In a typical scenario, the smart card can be stolen. In this case, we analyze that Chen et al.'s protocol cannot withstand off-line dictionary attack. In addition, we assume that the server's long-term key can be promised. On this assumption, we show that Chen et al.'s protocol cannot achieve forward secrecy. Furthermore, we point that Chen et al.'s protocol cannot resist impersonation attack launched by a privileged administrator and key control attack by the user. Finally, we propose an efficient protocol to eliminate above mentioned flaws. In our protocol, no timestamps are used. Thus, the proposed protocol is not prone to the problems of time synchronization. For the best of our knowledge, the proposed smart card protocol is the most efficient protocol to provide perfect forward secrecy.

The rest of the paper is organized as follows: Section 2 defines the notations, Section 3 reviews Chen et al.'s protocol. Section 4 elaborates the cryptanalysis of Chen et al.'s protocol. Section 5 proposes an enhanced two-factor authentication and key agreement protocol. Section 6 analyzes the security of the new efficient protocol. At the end, we conclude this paper.

II. NOTATIONS

The notations used throughout this paper are described as in the following.

- U: the user.
- ID: the identity of U.
- PW: the password of U.
- b: a random value b of the smart card
- S: the remote server.
- x: the permanent secret key of S.
- $h(\ )$: a cryptographic un-keyed hash function.
- p and q : two large prime numbers such that $p = 2q + 1$.
- g: a generator of order q in Z_p^*.
- r: a random number $r \in Z_q^*$ generated by the smart card.
- w: a random number $w \in Z_q^*$ generated by S.
- T_U: the timestamp of U.
- T_S: the timestamp of S.
- PW_{new}: the new password.
- $||$: string concatenation operation.

III. REVIEW OF CHEN ET AL.'S PROTOCOL

Chen et al.'s protocol can be described as in the following.

A. Registration phase

The server S chooses two large primes p and q where $p = 2q + 1$, its random secret key $x \in Z_q^*$, and cryptographic hash function $h()$.

1. The user U sends his identity ID, the password PW to S secretly.

2. Upon receiving ID and PW, the server S computes $B=h(ID)^{x+PW} \bmod p$. S stores $\{B, h, p, q\}$ into a smart card. At last, S sends the smart card to U.

The Figure 1 describes the registration phase of Chen et al.'s protocol.

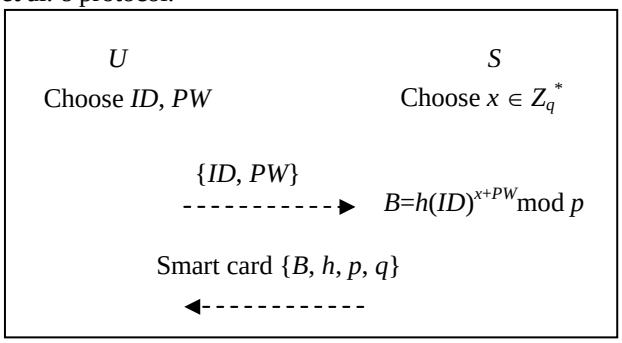

Figure 1. Registration phase of Chen et al.'s protocol

B. Login Phase

In login phase, the user U logins the server S by sending a login request message. The Figure 2 describes the login and authentication phases of Chen et al.'s protocol.

1. U keys in his ID and PW after he inserts the smart card.

2. The smart card selects a random number $r \in Z_q^*$ and computes.

$C = B/h(ID)^{PW} \bmod p$,

$D = h(ID)^r \bmod p$,

$W = CD \bmod p$, and

$M = h(ID||C||D||W||T_U)$, where T_U is U's timestamp.

3. The smart card submits the login request message $\{ID, D, M, T_U\}$to S.

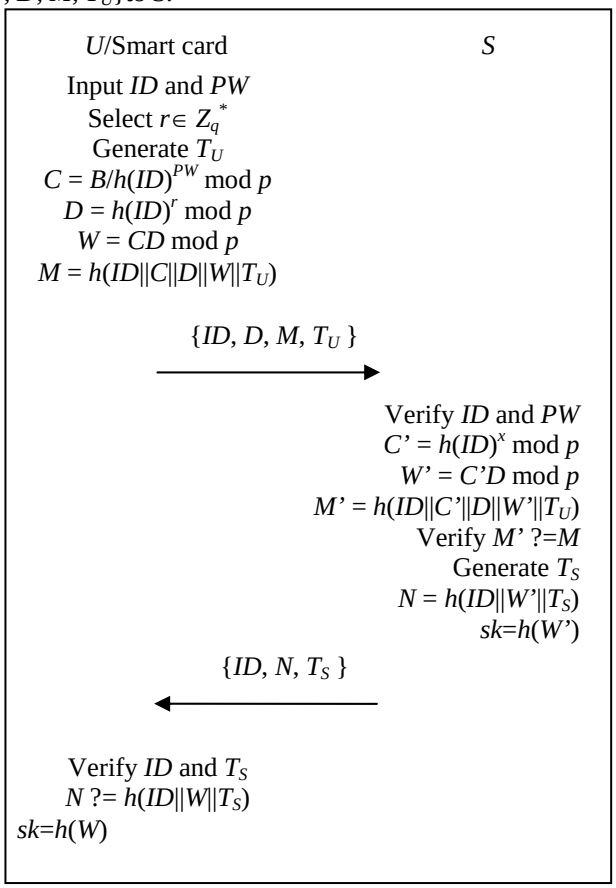

Figure 2. Login and authentication phases of Chen et al.'s protocol

C. Authentication Phase

In authentication phase, S and the smart card execute the following operations to authenticate each other.

1. The server S checks the identity ID and the timestamp TU. If ID or TU is invalid, then S rejects the login request.

2. S computes:

$C' = h(ID)^x \bmod p$,

$W' = C'D \bmod p$, and

$M' = h(ID||C'||D||W'||T_U)$.

3. S compares M' ?=M. If they are equal, the server authenticate the user and computes $sk=h(W')$ as the session key, otherwise, the server terminates the protocol run.

4. S computes $N = h(ID||W'||T_S)$ and sends ID, N, and T_S to U where T_S is the timestamp generated by S.

5. Upon receiving the message $\{ID, N, T_S\}$, U checks ID, T_S and N ?= $h(ID||W||T_S)$. If they are all valid, U successfully authenticates S and computes $sk=h(W)$ as the session key, otherwise U terminates the protocol run.

D. Password Change Phase

In password change phase, the user updates his password with a new one.

1. U inserts his smart card into the reader, enters ID, original password PW, and new password PW_{new}.

2. After authentication by the server to check the validity of PW, the smart card replaces B with $(B/h(ID)^{PW})h(ID)^{PW_{\text{new}}}$. Now, new password is successfully updated.

IV. Cryptanalysis of Chen Et Al.'s Protocol

A. Off-line Dictionary Attack

Unfortunately, Chen et al.'s scheme does not achieve two-factor security. We assume that the adversary has obtained the U's smart card and extracted the data stored in it and intercepted the transmitted message over the communication channel. Then the adversary can launch the off-line password guessing attack as follows.

1. The adversary obtains the values $\{B, h, p, q\}$ stored in U's smart card by physical attack.

2. The adversary intercepted the message $\{ID, D, M, T_U\}$ from a early session.

3. The adversary guesses a password PW' and calculate $C = B/h(ID)^{PW'} \bmod p$, $W = CD \bmod p$ and $M' = h(ID||C||D||W||T_U)$. Then the adversary compare M' with M. If M' is equal to W, the adversary obtains the correct password PW', otherwise, the adversary guesses another password.

B. Lack of Forward Secrecy

Chen et al. [6] claimed that their scheme provides forward secrecy, which means that disclose of long-term private keys of one or more participants will not affect the previous session keys. However, we show that Chen et al.'s scheme cannot provide forward secrecy. Suppose that an adversary obtains the server's long-term private key x. Then, he can compute the $C' = h(ID)^x \bmod p$ and $W' = C'D \bmod p$, where D is eavesdropped in the previous protocol run. Then, he can derive the previous session key $sk=h(W')$ which is established by the long-term private key x. It means that Chen et al.'s scheme cannot provide forward secrecy.

C. Privileged Administrator Attack

A user might use the same password because it's too difficult to remember numerous different passwords for a person. If the server's administrator knows a user's password, he may impersonate a legal user to login the security-sensitive applications such as banking. In registration phase of Chen et al.'s protocol, the user transmits $\{ID, PW\}$ directly to the server. The server's administrator can gain the password for access the different applications.

D. Key Control Attack

Chen et al.'s protocol does not provide key control resilience. In Chen et al.'s protocol, the user can control the value of the session key.

The user extracts the data stored in smart card and select r, W and computes.

$C = B/h(ID)^{PW} \bmod p$,

$D = WC^{-1} \bmod p$, and

$M = h(ID||C||D||W||T_U)$, where T_U is U's timestamp.

The smart card submits the login request message $\{ID, D, M, T_U\}$to S. After the server verifies the request information, the user and the server agree on the session key $sk=h(W)$.

The weakness of Chen et al.'s protocol against the key control attack is due to the fact that the server has no contribution in generating the session key.

V. Our Enhanced Protocol

In this section, we present an enhanced two-factor authentication and key agreement protocol that can eliminate the weaknesses described in the previous section.

A. Registration Phase

The server S chooses large primes p and q where $p = 2q + 1$, its random secret key, and cryptographic hash function $h()$. Let G be the cyclic group of order q with a generator g. Then the protocol proceeds in the following steps:

1. The user U chooses his identity ID, the memorable password PW and a random value b, then U computes $h(b \oplus PW)$ and sends ID and $h(b \oplus PW)$ to S secretly.

2. If the requester is a new user, S set $N = 0$. If the user revokes his lost smart card and re-apply for a new smart card, S set $N = N+1$. Then S computes $R=h(ID||x||N) \oplus h(b \oplus PW)$, $Y=h(x||ID||N)$. The server stores $\{R, Y, p, g, h\}$ into a smart card. At last, S sends the smart card to U.

3. Upon receiving the smart card from the server, the user stores the secret random value b into the smart card. Finally, the parameters $\{b, R, Y, p, g, h\}$ are stored in the smart card.

The Figure 3 describes the registration phase of proposed protocol.

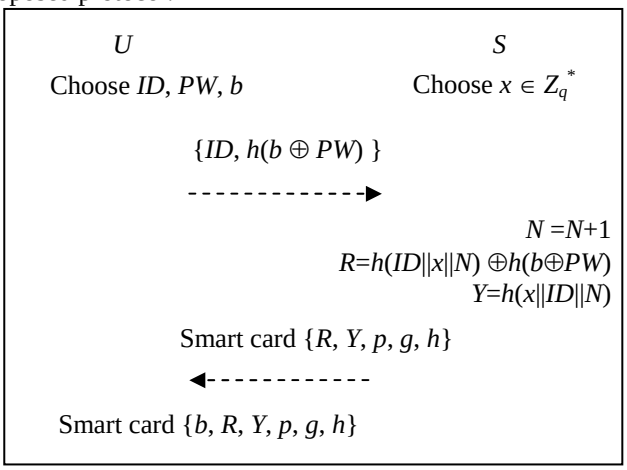

Figure 3. Registration phase of the proposed protocol

B. Authentication and Key Agreement Phase

When the user U wants to login the server, the user attaches his smart card to a reader and inputs his ID and PW. Then the remote server and the smart card will perform the following steps to achieve mutual authentication and agree on a shared session key.

1. U inserts his smart card into the reader, and then keys in his ID and PW. The smart card selects a random number r in Z_q^* and computes $C_1 = g^r$. The smart card submits the challenge message $\{ID, C_1\}$ to the server S.

2. After the message $\{ID, C_1\}$ is received, S chooses a random number w in Z_q^*, and computes $C_2 = g^w$, $D=(C_1)^w$ and $C_4 = h(D\| h(x\|ID\|N)\| C_1)$. S sends C_2 and C_3 to the smart card as the response message.

3. Upon receiving the message $\{C_2, C_3\}$, the smart card computes $D'=(C_2)^r$ and $C_3' = h(D' \|Y\| C_1)$. Then S compares C_3' with the received value of C_3. If they are equal, the smart card successfully authenticates S, otherwise the smart card terminates the protocol run. The smart card computes $k=h(D')$ as the session key.

4. The smart card sends a reply $C_4 = h(D' \| C_2\|R\oplus h(b\oplus PW))$ to the server. If C_4 is equal to $h(D \| C_2\| h(ID\|x\|N))$, the server authenticates the identity of the user. The server computes $k=h(D)$ as the session key.

The Figure 4 describes the authentication and key agreement phase of the proposed protocol.

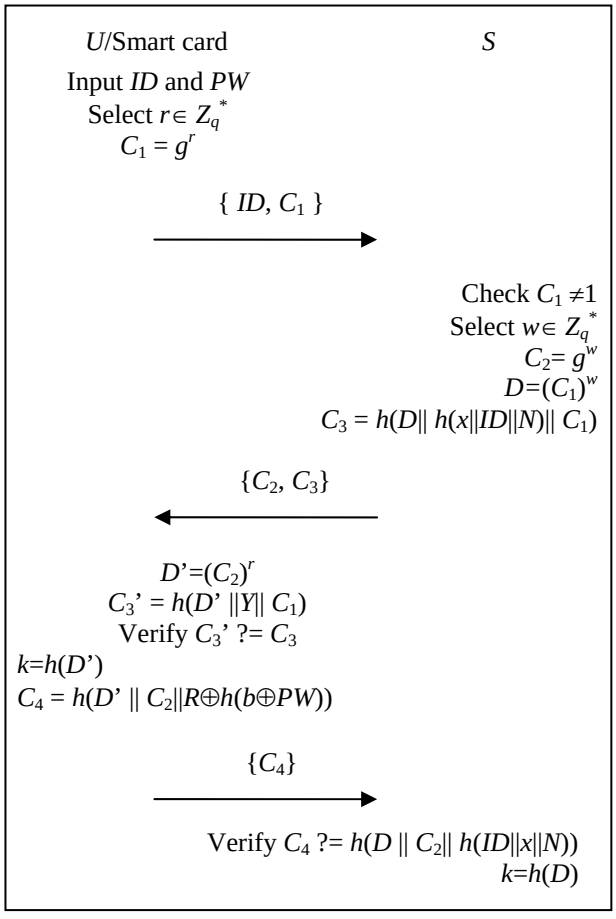

Figure 4. Authentication and key agreement phase of the proposed protocol

C. *Password change Phase*

In this phase, the user U freely changes his password PW with a new one PW_{new}. U inserts his smart card into the reader, and then keys in ID, PW and PW_{new}.

1-4. Step1 to 4 are same as these in Sec 5.2.

5. After mutaul authentication, S send $C_5 = h(D\| C_4)$ to U. Upon receiving the message C_5, the smart card computes $C_5' = h(D' \| C_4)$ and then compares it with the received value of C_5. If they are equal, the smart card compute $R_{new} = R\oplus h(b\oplus PW)\oplus h(b\oplus PW_{new})$ and then replaces R with R_{new}.

VI. Security Analysis

In this section, we discuss the security attributes of our proposed protocol.

(1) Mutual authentication

In the authentication and key agreement phrase, the smart card computes $C_3' = h(D' \|Y\| C_1)$ and then compares it with the received value of C_3. If they are equal, the smart card successfully authenticates S. The smart card sends a reply $C_4 = h(D' \| C_2\|R\oplus h(b\oplus PW))$ to the server. If C_4 is equal to $h(D \| C_2\| h(ID\|x\|N))$, the server authenticates the identity of the client. Now, we show the scheme can resist the following attacks:

Case1: User impersonation attack. If an adversary intends to impersonate a legal user, he should responds to message $\{C_2, C_3\}$ with a fresh message C_4 in order to pass the server's authentication. C_4 is protected in transmission messages by keyed hash function. To compute the item C_4, an adversary must know $R\oplus h(b\oplus PW)$. The adversary can not know R and b stored in the smart card and the password PW. This means that the adversary has no way to obtain $R\oplus h(b\oplus PW)$ and is therefore unable to compute C_4.

Case2: Server impersonation attack. If an adversary intends to impersonate the server to fool the requesting user, he should forge message $\{C_2, C_3\}$ to respond the challenge message. However, C_3 is protected by a key $h(x\|ID\|N)$. Without knowing the server's secret key x, the adversary can not compute a valid C_3.

Case3: Replay attack. In each session, the smart card and the server generate different random value r, and w. The adversary must send a fresh message $\{C_2, C_3\}$ to pass authentication by the user or send fresh message $\{C_4\}$ to pass authentication by the server. This challenge–response mechanism can avoid replay attack.

From above analysis, we conclude the proposed protocol satisfies mutual authentication.

(2) Key secrecy

In our protocol, the session is obtained from ephemeral values r and w generated by the user and the server. The adversary cannot compute the shared secret $k=h(g^{rw})$ from the eavesdropped messages C_1, and C_2 since Diffie-Hellman problem is assumed intractable.

(3) Known-key security

For each new login request, the protocol generates different random values r and w to compute session key k. This means that the generated session keys are independent among the different protocol runs. The

compromise of one session key could not compromise another session key.

(4) Perfect forward secrecy

In our protocol, the two ephemeral values *r* and *w* are randomly generated and independent among each protocol execution. Therefore, compromise of the user's password *PW*, secret value *b* and the server's long-term key *x* cannot reveal any previous session keys. Thus, the proposed protocol has perfect forward secrecy.

(5) Privileged administrator resilience

In our protocol, a new user can freely choose his easy-to-remember password *PW* for registering to the server. The user blind his password with a secret value *b* and sends $h(b \oplus PW)$ to the server, without knowing *b* the server has no feasible method to obtain or guess *PW*. Thus, the proposed protocol can resist the attacks launched by the privileged administrator.

In our protocol, no password or verification table is stored in the server. Thus, the cost of maintaining the security-sensitive information table is reduced. If the server maintains some verification table to store verification information, it will become an attractive target for the adversaries. The proposed scheme contains no verification table in server side. In this case, stolen-verifier attacks by the privileged administrator or outside adversaries are considered impossible.

(6) Key control resilience

As shown in authentication and key agreement phase, after finishing mutual authentication, the user and the remote server use the Diffie-Hellman key exchange to generate a new session key $k=h(g^{rw})$. They can use *k* to protect the transmitting messages in the communication. Since key materials *r* and *w* are random generated and independent by the user and the server, neither the user nor the server can predict the value of the session key.

(7) Freely chosen password

In our protocol, the user can choose his password in the registration phase and change his password in the password change phase.

(8) Two-factor security

Suppose an adversary obtains a user's password. If he intends to impersonate the user to login the server, he should compute the challenge $C_4 = h(D' \| C_2 \| R \oplus h(b \oplus PW))$. However, without *R* and *b* stored in smart card, the adversary cannot construct C_2.

Suppose an adversary gets a legal user's smart card. In the following, we discuss two of the most common attacks.

Case1:Off-line guessing attack. If a smart card is lost, the adversary gets $R=h(ID\|x\|N) \oplus h(b \oplus PW)$, $Y=h(x\|ID\|N)$ and *b*. When the adversary tries to deduce a password from *R* and *b*, he cannot verify without knowing *x*.

Case2: Impersonation attack. We consider the scenario that an adversary has stolen a smart card from a legal user, but does not know the password. If an adversary impersonates the user to login the server, he can not construct the valid message $C_4 = h(D' \| C_2 \| R \oplus h(b \oplus PW))$, since he doesn't know the password.

From above analysis, we conclude the proposed protocol is securing even that the adversary steals the smart card or gets the user's password.

(9) Availability

In password change phase, the user changes his password must be verified by the server. If the adversary obtains the user's smart card temporarily, he could not change the password to a new one without the user's password. Thus, there is no password synchronization problem between the user and the smart card. Since the server does not store the password verification information, there is no password synchronization problem between the user and the server. Therefore, the protocol is not prone to the denial of service attack.

TABLE I.
PERFORMANCE COMPARISON.

Related works	Cost of Authentication and key Agreement phase
Chen et al.[14]	$3T_e + 3T_m + 8T_h$
Xie[15]	$6T_e + 8T_h$
Xu et al. [9]	$6T_e + 9T_h$
Ours	$4T_e + 8T_h$

VII. PERFORMANCE COMPARISONS

For convenience, let T_e , T_m and T_h be the time of once modular experimental operation, once multiplication/division operation and once one-way hash function operation, respectively. We ignore exclusive OR operations and string concatenation operations. We mainly consider performance of the authentication and key agreement phase. To evaluate the time complexity of the proposed protocol, we compare our scheme with Chen et al. [14], Xie [15], and Xu et al. [9] in Table 1.

Although Chen et al.'s protocol [14] is more efficient, but does not provide perfect forward secrecy. To achieve perfect forward secrecy, Xie [5], Xu et al. [9] and ours use the Diffie-Hellman key exchange to generate session keys. Compared with this two protocols, our protocol is more efficient.

VIII. CONCLUSIONS

Chen et al.'s proposed a password authentication protocol based smart cards. They claimed their protocol is secure against certain known attacks. In this paper, we have shown that their protocol suffers from the off-line password guessing attack, privileged administrator attack, key control attack and lacks of forward security. We also proposed an efficient protocol to eliminate the weaknesses. In our protocol, we removed the timestamp and avoided clock synchronization between the user and the server.

REFERENCES

[1] LI W M, WEN Q Y, SU Q, ZHANG H, JIN Z P. ,"Password-Authenticated Multiple Key Exchange Protocol for Mobile Applications," in *China Communications*, vol. 9,No. 1,pp. 64-72,2012.
[2] LAMPORT L. ,"Password Authentication with Insecure Communication," *in Commun Acm*,vol. 24,No. 11,pp. 770-772, 1981.
[3] KOCHER P, JAFFE J, JUN B. ,"Differential power analysis," in *Advances in Cryptology-CRYPTO'1999, Lecture Notes in Computer Science, Springer Berlin/ Heidelberg*, 1999,vol. 1666, pp. 388–397.
[4] MESSERGES T S, DABBISH E A. SLOAN R H. , " in Examining smart card security under the threat of power analysis attacks," in *IEEE Transaction on Computers*, vol. 51,No. 5,pp.541–552,2002.
[5] WANG R-C, JUANG W-S, LEI C-L," A simple and efficient key exchange scheme against the smart card loss problem," *in Proceedings of the 2007 IFIP International Conference on Embedded and Ubiquitous Computing, Lecture Notes in Computer Science, Springer Berlin/ Heidelberg*, vol. 4809,pp. 728–744,2007.
[6] JABLON D. , "Extended password key exchange protocols immune to dictionary attack," *in Proceedings of the WETICE Workshop on Enterprise Security*, pp. 248-255,1997.
[7] AHN L, BLUM M, HOPPER N, LANGFORD J. , "CAPTCHA: Using Hard AI Problems for Security," in Conference on EUROCRYPT'03, Lecture Notes in Computer Science, 2003,vol. 2656, pp. 294-311.
[8] HWABG M S, LI L H. ,"A new remote user authentication scheme using smart cards," *in IEEE Transactions on Consumer Electronics*, vol. 46,No. 1,pp. 28–30,2000.
[9] XU J, ZHU W T, FENG D G. ,"An improved smart card based password authentication scheme with provable security," *in Computer Standards & Interfaces*, vol. 31,No. 4,pp. 723–728, 2009.l
[10] LEE S W, KIM H S, YOO K Y.," Improvement of Chien et al.'s remote user authentication scheme using smart cards," *Computer Standards & Interfaces*, vol. 27,No. 2,pp. 181–183, 2005.
[11] LEE N Y, CHIU Y C. "Improved remote authentication scheme with smart card", *Computer Standards & Interfaces*, vol. 27,No. 2,pp.177–180, 2005.
[12] SONG R., "Advanced smart card based password authentication protocol", *Computer Standards & Interfaces*, vol. 32, No. 5,pp. 321–325,2010.
[13] SOOD S K, SARJE A K, SINGH K. , "An improvement of Xu et al.'s authentication scheme using smart cards," in *the Conference on Proceedings of The Third Annual ACM Bangalore, Bangalore, Karnataka, India*,2010,pp. 1–5.
[14] CHEN B-L, KUO W-C, WUU L-C. , "Robust smart-card-based remote user password authentication scheme", *International Journal of Communication Systems*, Article first published online: 21 MAY 2012.
[15] Qi Xie, "Improvement of a security enhanced one-time two-factor authentication and key agreements cheme," in *Scientia Iranica*, 2012. from http://dx.doi.org/10.1016/j.scient.2012.02.029.

Fenghua Liu was born in Henan, China, in December 1976. She is currently working towards her Ph.D. degree at the School of Information and Electrical Engineering, CUMT,China. She received the B.S. degree in Computer Software and M.S. degree in Computer Applied Technology in 1998 and 2005, respectively, both from the China University of Mining and Technology.

She currently works for XuZhou Institute of Technology, China, where she is Senior Lecturer in Computer. Her research interests include Computer Security, Data Mining and Networks.