Comments on two password-based protocols

*Yalin Chen [1], Hung-Min Sun [2], Chun-Hui Huang [3], Jue-Sam Chou[4]
[1] Institute of information systems and applications, National Tsing Hua University
*: corresponding author
d949702@oz.nthu.edu.tw
[2] Institute of information systems and applications, National Tsing Hua University
hmsun@cs.nthu.edu.tw
[3] Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C
g6451519@mail.nhu.edu.tw
4 Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C
jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536

Abstract

Recently, M. Hölbl *et al.* and I. E. Liao *et al.* each proposed an user authentication protocol. Both claimed that their schemes can withstand password guessing attack. However, T. Xiang *et al.* pointed out I. E. Liao *et al.*'s protocol suffers three kinds of attacks, including password guessing attacks. We present an improvement protocol to get rid of password guessing attacks. In this paper, we first point out the security loopholes of M. Hölbl *et al.*'s protocol and review T. Xiang *et al*.'s cryptanalysis on I. E. Liao *et al*.'s protocol. Then, we present the improvements on M. Hölbl *et al.*'s protocol and I. E. Liao *et al.*'s protocol, respectively.

Keywords: *smart card, password authentication protocol, password change*

1. Introduction

Password-based authentication is widely adopted to login the remote server. It can provide authentication between the client and the server in an open network to ensure the legality of a user and the correctness of a server. Many schemes in this area were proposed, such as two-party password authenticated key exchange (2PAKE) protocols for the client-server architecture [1-15], 3PAKE protocols for the client-client-server architecture [16-23], or multi-server PAKE protocols for the client-servers architecture [24-25].

In 2006, M. Peyravian *et al.*[12] proposed *secure remote user access over insecure networks*. But in 2008, M. Hölbl *et al.*[10] pointed out M. Peyravian *et al.*'s protocol is vulnerable to password guessing attacks and proposed an improvement on them. However, we found M. Hölbl *et al.*'s protocol still suffers from password guessing attacks. In this paper, we will present the attack and improve M. Hölbl *et al.*'s protocol to make it really safe for practical applications. Also in 2006, I. E. Liao *et al.*[11] proposed *a password authentication scheme over insecure networks*. They proposed some requirements for evaluating a password-based authentication protocol and claimed that their protocol can achieve these requirements and are immune to various attacks. But in 2008, T. Xiang *et al.*[8] pointed out three kinds of attacks on I. E. Liao *et al.*'s protocol. However, they demonstrated the attacks without presenting a modification. Therefore, we will modify I. E. Liao *et al.*'s protocol to make them really secure. We will show both of

our two improvements are secure and efficient.

The remainder of this paper is organized as follows: In Section 2, we review M. Hölbl *et al.*'s and I. E. Liao *et al.*'s protocols, respectively. In Section 3, we analyze M. Hölbl *et al.*'s protocol and T. Xiang *et al.*'s three attacks on I. E. Liao *et al.*'s protocol. We present our two improvements for M. Hölbl *et al.*'s and I. E. Liao *et al.*'s protocols in Section 4. Then, We analyze the security and efficiency of our improvements in Section 5. Finally, a conclusion is given in Section 6.

2. Review of M. Hölbl *et al.*'s and I. E. Liao *et al.*'s protocols

In this section, we review M. Hölbl *et al.*'s protocol in Section 2.1 and I. E. Liao *al.*'s protocol in Section 2.2, respectively. The notations used are first described below.

C, S : a client and a server, respectively.
E : an adversary/attacker.
ID : the identity of C.
PW : the password of C.
p : a large prime number.
g : the primitive element in a Galois field GF(p) where GF(p) is the set of integers $\{0,1,\cdots,p-1\}$ with arithmetic operations defined on modulo p.
H : a collision-resistant one-way hash function.
(a,b) : string *a* is concatenated with string *b*
$\oplus$: an exclusive-or operation.
$\triangle T$: the tolerance time for transmission delay.
s : S's secret key.

2.1 Review of M. Hölbl *et al.*'s protocol

In this section, we review M. Hölbl *et al.*'s authentication protocol in Section 2.1.1 and password change protocol in Section 2.1.2.

2.1.1 User authentication protocol

We describe M. Hölbl *et al.*'s user authentication protocol as follows and also depict it in Figure 1.

In their scheme, a user, C, has to register at server S to become the legal client and S stores C's *IDPW-dig*(=H(*ID*, *PW*)) instead of *PW*. They perform the following steps.

1. C generates a random value r_C, chooses a large random integer *x*, and computes g^x mod p. Then, C masks g^x mod p by computing $m\text{-}g^x = g^x \oplus H(ID, IDPW\text{-}dig)$, where $IDPW\text{-}dig = H(ID, PW)$ and sends a message $\{ID, r_C, m\text{-}g^x\}$ to S.
2. After receiving the message, S retrieves g^x by computing $g^x = m\text{-}g^x \oplus H(ID, IDPW\text{-}dig)$. Then, S chooses a random value r_S, a large random integer *y* and computes g^y mod p. He calculates $(g^x)^y$ mod p, generates $ch1 = r_S \oplus H(g^{xy}, IDPW\text{-}dig, r_C)$, $ch2 = g^{xy} \oplus$

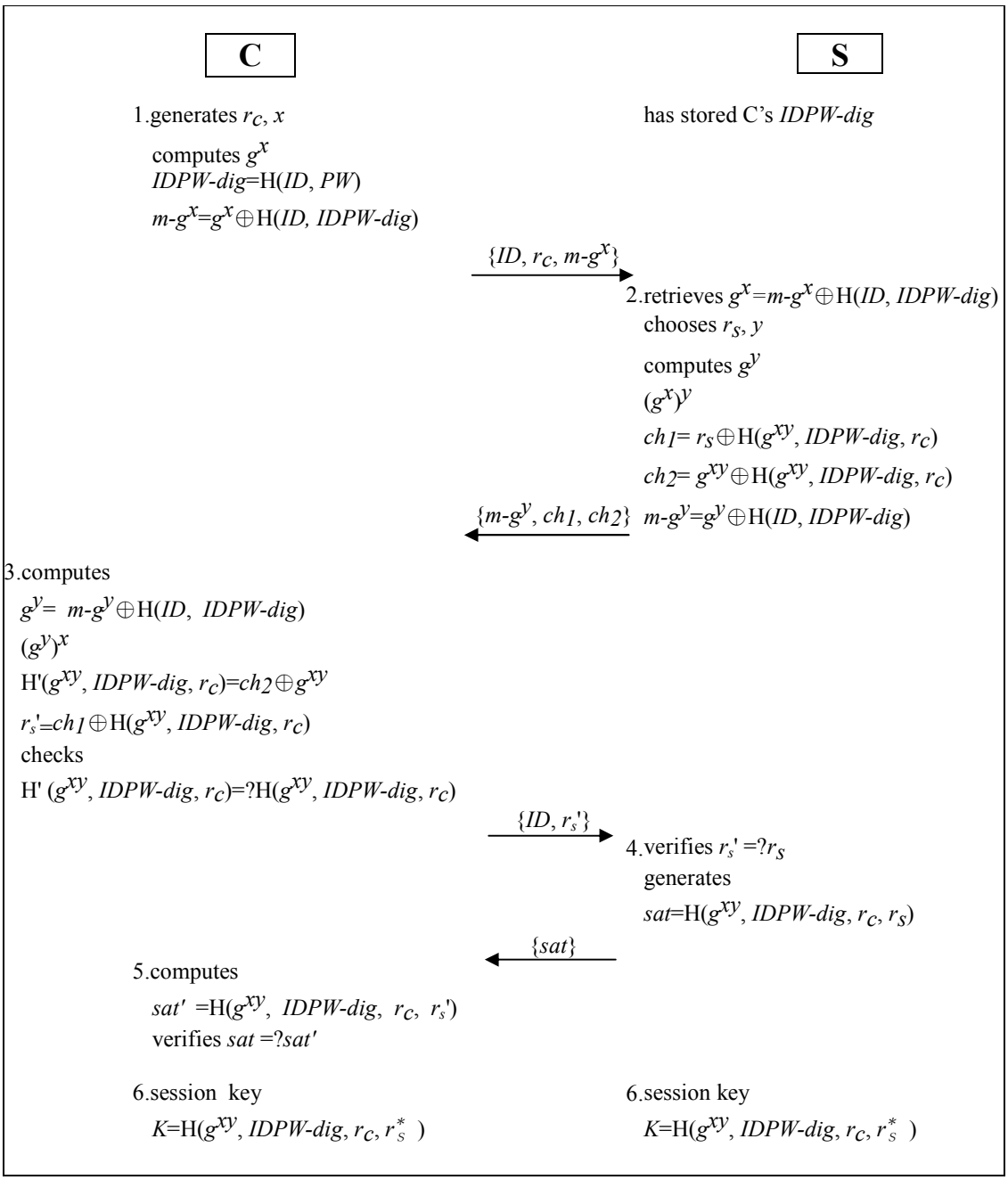

Fig. 1. M. Hölbl *et al.*'s user authentication protocol

H(g^{xy}, $IDPW\text{-}dig$, r_C) and masks g^y as $m\text{-}g^y$ by computing $m\text{-}g^y=g^y \oplus$H(ID, $IDPW\text{-}dig$). Then, S sends {$m\text{-}g^y$, $ch1$, $ch2$} to C.

3. On receipt of the message, C derives $g^y= m\text{-}g^y \oplus$H(ID, $IDPW\text{-}dig$). Then, C computes $(g^y)^x$ mod p and derives H'(g^{xy}, $IDPW\text{-}dig$, r_C) by computing $ch2 \oplus g^{xy}$. C checks to

see if the derived H' (g^{xy}, *IDPW-dig*, r_C) is equal to the computed H(g^{xy}, *IDPW-dig*, r_C). If it is, C then retrieves r_s' by computing $ch1 \oplus$H(g^{xy}, *IDPW-dig*, r_C). Otherwise, S is not genuine and C terminates the protocol. Then, C sends {*ID*, r_s'} to S.

4. After receiving {*ID*, r_s'}, S verifies if the received r_s' is the same as his own generated r_S. If they are the same, C is authentic. Next, S generates a authentication token *sat*=H(g^{xy}, *IDPW-dig*, r_C, r_S) and sends {*sat*} to C.
5. After receiving {*sat*}, C computes *sat'*=H(g^{xy}, *IDPW-dig*, r_C, r_s') and verifies if the received *sat* is equal to *sat'*. If the verification succeeds, S is authentic.
6. After successful authentication, they can generate the session key as *K*=H(g^{xy}, *IDPW-dig*, r_C, r_S^*) where r_S^* is r_S plus some fixed value in order for *K* to be different from *sat*.

2.1.2 Password change protocol

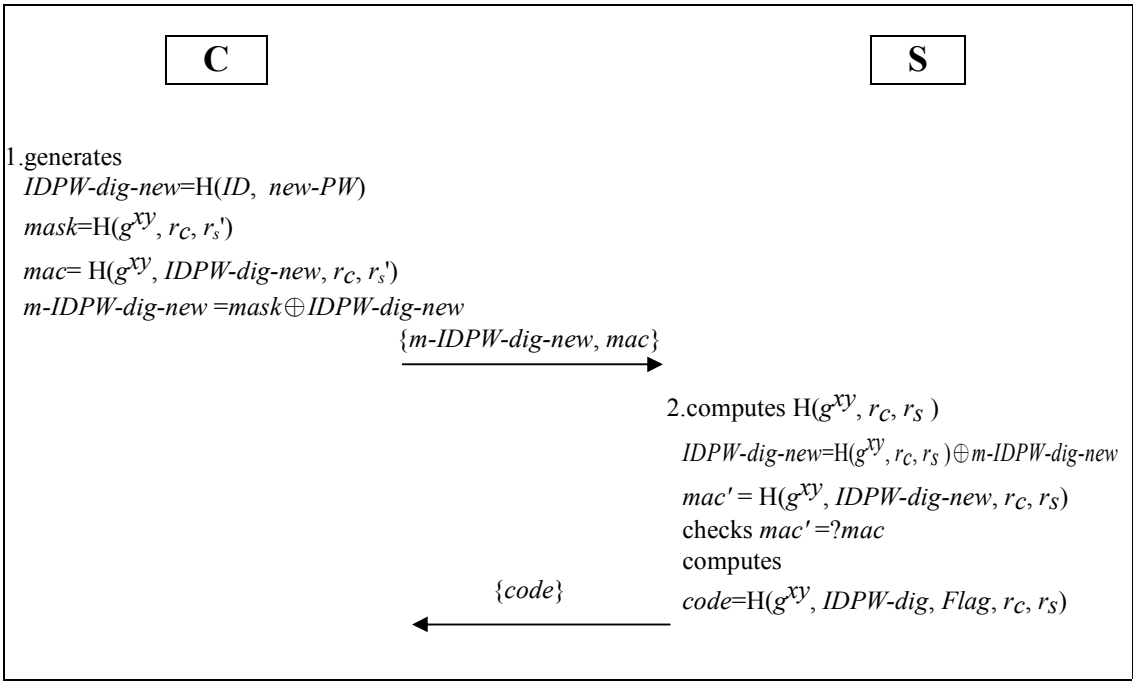

Fig. 2. Password update protocol of M. Hölbl et al.'s password change protocol

In their protocol, when C wants to update his password *PW* as *new-PW*, he proceeds with the password update protocol as follows. It is also shown in Figure 2.

1. After authenticating the server, C generates *mask*=H(g^{xy}, r_C, r_s'), *mac*= H(g^{xy}, *IDPW-dig-new*, r_C, r_s') and *m-IDPW-dig-new=mask* $\oplus$ *IDPW-dig-new*, where *IDPW-dig-new*=H(*ID*, *new-PW*). Then, C sends {*m-IDPW-dig-new*, *mac*} to S.
2. After receiving the message, S verifies the validity of the received *mac*. He retrieves *IDPW-dig-new* by computing H(g^{xy}, r_C, r_S)$\oplus$*m-IDPW-dig-new*. Next, S computes

mac' = H(g^{xy}, *IDPW-dig-new*, r_C, r_S) and compares mac' with the received *mac*. If it is valid, S accepts the password change and replaces *IDPW-dig* with *IDPW-dig-new*. Otherwise, he rejects the password change. He then sends a message *code*=H(g^{xy}, *IDPW-dig*, *Flag*, r_C, r_S) to C, where *Flag* is set to either '*accept*' or '*reject*' depending upon whether the password change is accepted or rejected.

2.2 Review of I. E. Liao *et al.*'s protocol

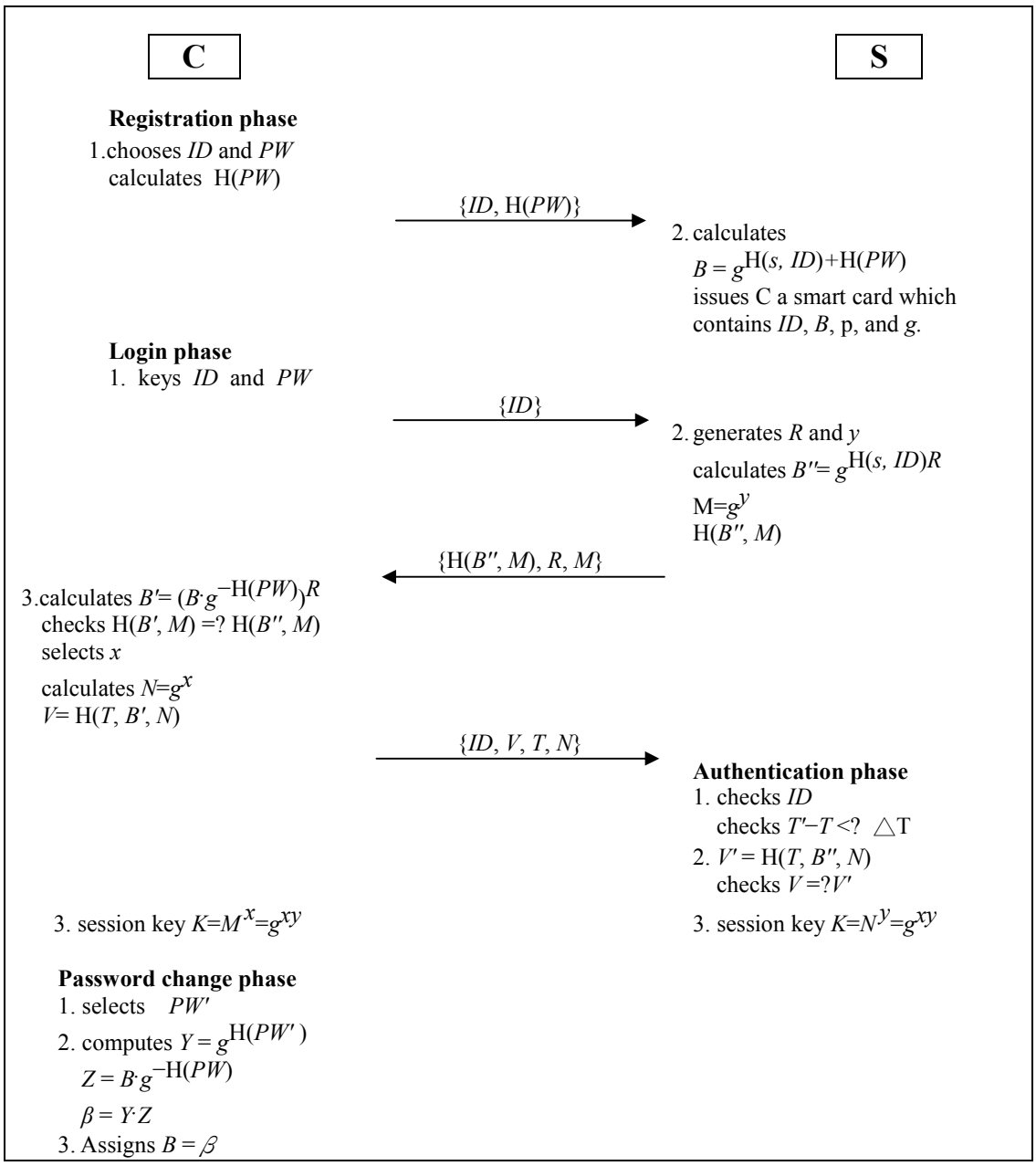

Fig. 3. Review of I. E. Liao et al.'s protocol

In this section, we briefly review I. E. Liao *et al.*'s scheme. The scheme consists of four phases, registration phase, login phase, authentication phase and password change phase. This four-phase protocol is described as follows and also illustrated in Figure 3.

2.2.1 Registration phase

In this phase, C performs the following steps to register S for obtaining a smart card.

1. C freely chooses his *ID* and *PW*, and calculates H(*PW*). C then sends {*ID*, H(*PW*)} to S through a secure channel.
2. S calculates $B = g^{\mathrm{H}(s,\ ID)+\mathrm{H}(PW)} \bmod p$. S then issues to C a smart card which contains *ID*, *B*, p, and *g* through a secure channel.

2.2.2 Login phase

When C wants to login to S, he inserts his smart card and cooperates with S to perform the following steps.

1. C keys his *ID* and *PW* to smart card and sends {*ID*} to S.
2. S generates random numbers *R* and *y* and calculates $B'' = g^{\mathrm{H}(s,\ ID)R} \bmod p$ and $M=g^{y}$. He then computes H(*B*″, *M*) and sends {H(*B*″, *M*), *R*, *M*} to C.
3. C calculates $B' = (B \cdot g^{-\mathrm{H}(PW)})^{R} \bmod p$ and checks to see if H(*B*′, *M*) is equal to H(*B*″, *M*). If so, S is authentic. C then selects a random number *x*, calculates $N=g^{x} \bmod p$ and computes *V*= H(*T*, *B*′, *N*), where *T* is the timestamp of this login. He then sends {*ID*, *V*, *T*, *N*} to S.

2.2.3 Authentication phase

In this phase, S executes the following steps to determine whether C is allowed to login or not.

1. S generates the timestamp *T*′, checks *ID* and compares if $T'-T$ is less than $\triangle$T. If ID is invalid or $T'-T > \triangle$T, the login request is rejected.
2. S computes *V*′ = H(*T*, *B*″, *N*), and then checks if *V* is equal to *V*′. If it is, C is authentic. Otherwise, S stops the protocol.
3. After successful authentication, S computes the session key as $K=N^{y}=g^{xy}$ and C also has the session key as $K=M^{x}=g^{xy}$.

2.2.4 Password change phase

This phase will be invoked if C wants to change his password from *PW* to *PW*′.

1. C selects a new password *PW*′.
2. C computes $Y = g^{\mathrm{H}(PW')} \bmod p$, $Z = B \cdot g^{-\mathrm{H}(PW)} \bmod p$, and $\beta = Y \cdot Z$, where *PW* is the

old password and B is the variable stored in the smart card.
3. C assigns $B = \beta$ in the smart card.

3. Security issues of M. Hölbl *et al.*'s and I. E. Liao *et al.*'s protocols

In this section, we will show the security loopholes of M. Hölbl *et al.*'s protocol in Section 3.1 and review T. Xing*et al.*'s cryptanalysis on I. E. Liao *et al.*'s protocol in Section 3.2.

3.1 Off-line password guessing attack on M. Hölbl *et al.*'s protocol

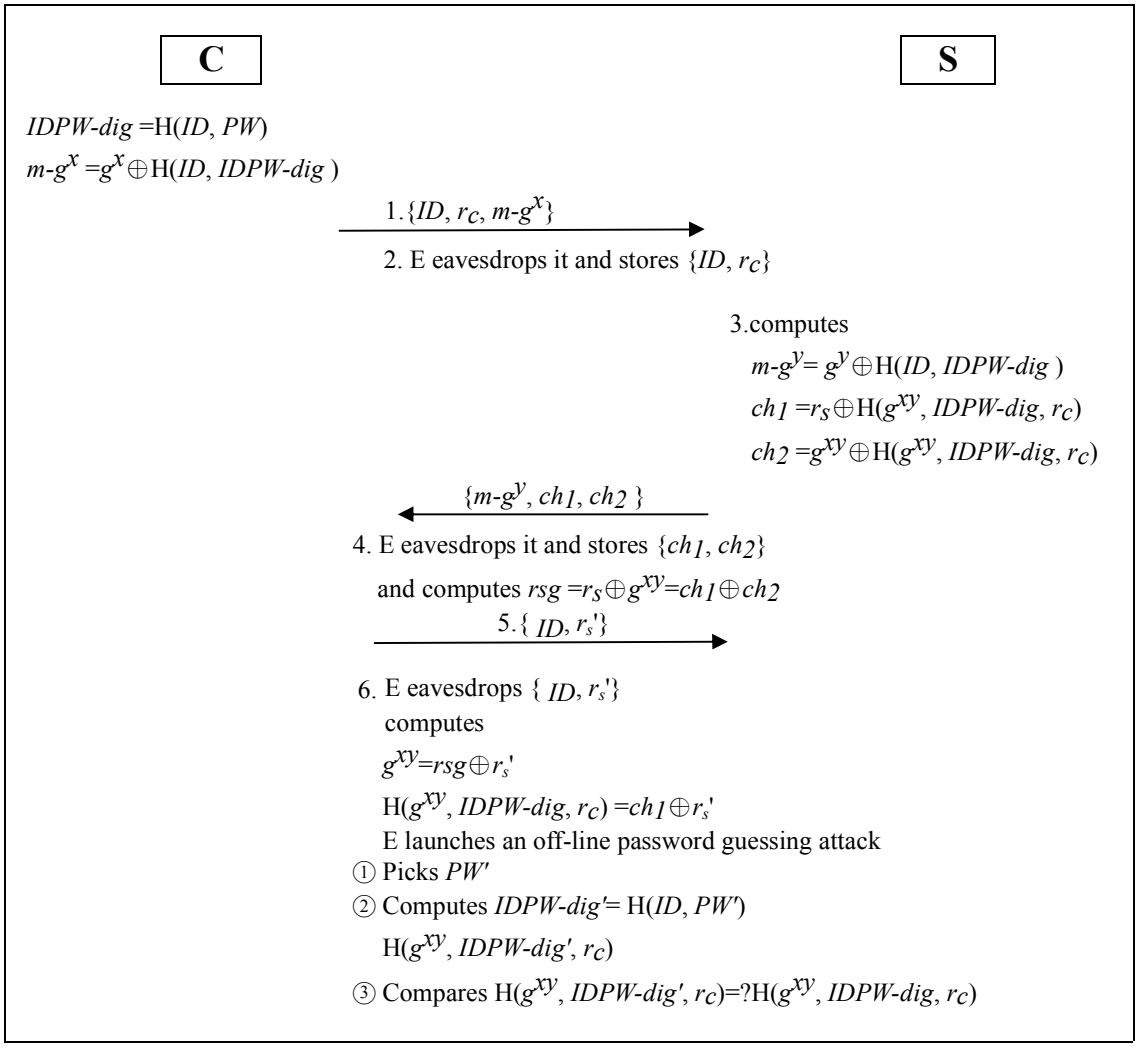

Fig. 4. Off-line password guessing attack on M. Hölbl et al.'s protocol

In 2006, Peyravian and Jeffries proposed *secure remote user access over insecure networks* [12]. They claimed their protocol is secure. But M. Hölbl *et al.* pointed out their protocol is insecure and proposed their improved protocol [10]. They also claimed that their improved scheme of the Peyravian-Jeffries's user authentication protocol and

password change protocol is secure against off-line password guessing attacks. However, we found that their improvement is still unable to get rid of such the attack. We describe this as follows and also depict it in Figure 4.

1. C starts the protocol run and sends a message $\{ID, r_C, m\text{-}g^x\}$ to S, where $m\text{-}g^x = g^x \oplus$ H(ID, $IDPW\text{-}dig$) and $IDPW\text{-}dig$ =H(ID, PW).
2. An adversary E eavesdrops it and stores the message $\{ID, r_C\}$.
3. After receiving the message $\{ID, r_C, m\text{-}g^x\}$, S computes $m\text{-}g^y = g^y \oplus$H($ID$, $IDPW\text{-}dig$), $ch_1 = r_S \oplus$H(g^{xy}, $IDPW\text{-}dig$, r_C) and $ch_2 = g^{xy} \oplus$H(g^{xy}, $IDPW\text{-}dig$, r_C). S sends the message $\{m\text{-}g^y, ch_1, ch_2\}$ to C.
4. Then, E eavesdrops and stores the message $\{ch_1, ch_2\}$. E calculates $ch_1 \oplus ch_2$ and gets $rsg = r_S \oplus g^{xy}$.
5. On receipt of the message, C verifies it. After authenticating S to be valid, he sends $\{ ID, r_s'\}$ to S, where $r_s' = r_S$ if all messages have not been modified by another attacker.
6. Next, E eavesdrops it and computes $rsg \oplus r_s'$. E can therefore retrieve g^{xy} and extract H(g^{xy}, $IDPW\text{-}dig$, r_C) from $ch_1 \oplus r_s'$.

Finally, E can launch an off-line password guessing attack and find the password PW by iterating upon all possible choices of PW':

① Picks a candidate password PW'.

② Computes $IDPW\text{-}dig'$= H(ID, PW') and H(g^{xy}, $IDPW\text{-}dig'$, r_C) .

③ Compares H(g^{xy}, $IDPW\text{-}dig'$, r_C) with H(g^{xy}, $IDPW\text{-}dig$, r_C) .

3.2 T. Xiang *et al.*'s cryptanalysis on I. E. Liao *et al.*'s protocol

In 2006, I. E. Liao *et al.* proposed *a password authentication scheme over insecure networks* [11]. They also claimed that their scheme can be extended to support Diffie-Hellman key agreement protocol. However, in 2008, T. Xiang *et al.*[8] pointed out that I. E. Liao *et al.*'s scheme is potentially vulnerable to three kinds of attacks, off-line password guessing attack, replay attack and Denial-of-Service attack. We will review these three kinds of attacks in turn.

3.2.1 Off-line password guessing attack

An attacker E gets C's smart card and reads all data $\{ID, B, p, g\}$. E impersonates C and sends the login request to S. S then sends E the message $\{$H(B'', M), R, $M\}$. Now, E can guess the password PW' to compare if the value of H(B'', M) is equal to the value of H($(B \cdot g^{-\mathrm{H}(PW')})^R$ mod p, M). E repeats the way by using all possible choices of PW'. We show this attack in Figure 5.

3.2.2 Impersonating the server by replay attack

E can replay the intercepted message {H(*B''*, *M*), *R*, *M*} from S to C and C will believe

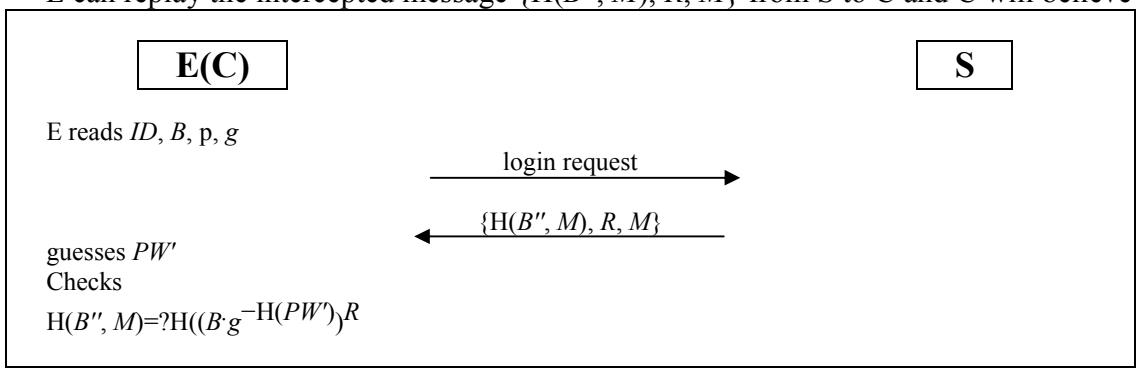

Fig. 5. Off-line password guessing attack on I. E. Liao et al.'s protocol

it is from the legal server S. However, E can't get the large random number y from M ($=g^y \bmod p$) because it is a discrete logarithm problem. And E can't know the session key $K=g^{xy}$ from $N=g^x$ and $M=g^y$ because it is the Decision Diffie-Hellman problem.

3.2.3 Denial-of-Service attack on password change

Assuming that a user frequently uses the smart card, the card is often inserted to the card reader on line for a long time. An active attacker can eavesdrop, modify, remove and insert messages into the channel. Suppose that an attacker E temporarily gets access to the client C's smart card and performs the following operations. He randomly selects two different passwords: *PW** as the old password and *PW'* as the new password. Then E sends a password change request to the smart card. The smart card will compute $Y = g^{H(PW')} \bmod p$, $Z = B \cdot g^{-H(PW^*)} \bmod p$, and $\beta = Y \cdot Z = g^{H(PW')} \cdot B \cdot g^{-H(PW^*)} = g^{H(PW')} \cdot g^{H(s, ID)+H(PW)} \cdot g^{-H(PW^*)} = g^{H(s, ID)+H(PW)-H(PW^*)+H(PW')}$, then replaces B with β. From then on, C can't authenticate S and vice versa with his password or his new password changed by himself.

4. Our improved protocols

In this section, we present two improved protocols on M. Hölbl *et al.*'s protocol and I. E. Liao *et al.*'s protocol in Section 4.1 and Section 4.2, respectively.

4.1 Improvement on M. Hölbl *et al.*'s protocol

In this section, we describe the improvement on M. Hölbl *et al.*'s authentication protocol as shown in figure 6.

The improved authentication protocol performs the following steps.

1. C generates a random nonce r_C, chooses a large integer nonce x, computes $g^x \bmod p$ and masks it by computing $m\text{-}g^x = g^x \oplus H(ID, IDPW\text{-}dig)$. Then C sends message {*ID*,

r_C, $m\text{-}g^x$} to S.

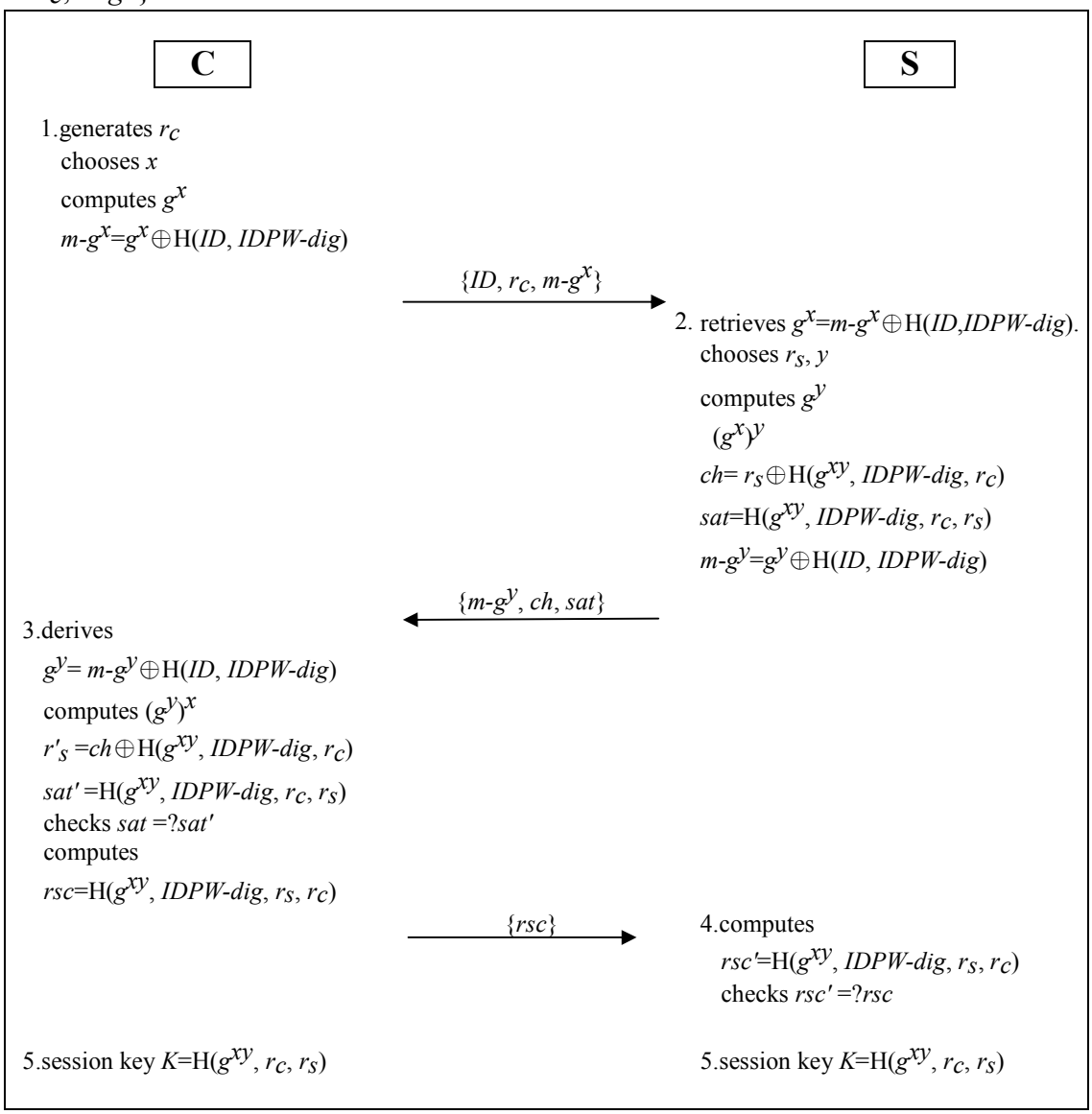

Fig. 6. Improvement on M. Hölbl et al.'s authentication protocol

2. After receiving the message, S retrieves g^x by computing $g^x=m\text{-}g^x \oplus H(ID,IDPW\text{-}dig)$. Then, he chooses a random nonce r_S, a large integer nonce y and computes g^y mod p. S calculates $(g^x)^y$ mod p, generates $ch= r_S \oplus H(g^{xy}, IDPW\text{-}dig, r_C)$, $sat=H(g^{xy}, IDPW\text{-}dig, r_C, r_S)$ and masks g^y as $m\text{-}g^y$ by computing $m\text{-}g^y=g^y \oplus H(ID, IDPW\text{-}dig)$. Then, S sends $\{m\text{-}g^y, ch, sat\}$ to C.
3. On receipt of the message from S, C derives $g^y= m\text{-}g^y \oplus H(ID, IDPW\text{-}dig)$, computes $(g^y)^x$ mod p and retrieves r_s' by computing $ch \oplus H(g^{xy}, IDPW\text{-}dig, r_C)$. C also

computes sat'=H(g^{xy}, *IDPW-dig*, r_C, r_S) and checks if *sat* is equal to the computed sat'. If it is, S is authentic. C then computes *rsc*=H(g^{xy}, *IDPW-dig*, r_S, r_C) and sends {*rsc*} to S.

4. After receiving {*rsc*}, S computes rsc'=H(g^{xy}, *IDPW-dig*, r_S, r_C) and verifies if rsc' is the same as the received *rsc*. If they are the same, C is authentic.
5. After successful mutual authentication, C and S have the same session key K=H(g^{xy}, r_C, r_S).

4.2 Improvement on I. E. Liao *et al.*'s protocol

For improving I. E. Liao *et al.*'s protocol, our proposed scheme consists of three phases, registration phase, login phase, and authentication phase. We describe them as follows.

4.2.1 Registration phase

Our registration phase is the same as original scheme described in Section 2.2.1.

4.2.2 Login and authentication phases

Generally, we login to a server just for doing something. It needs only authentication and seldom for password change. Therefore, for efficiency consideration, our login and authentication phases are divided into two scenarios: (A) for authentication only as shown in Figure 7. (B) for authentication and password change as shown in Figure 8.

(A) For authentication only

(1) Login phase

If C wants to communicate with S without changing his password, C will run the following steps.

1. C inserts his smart card and keys {*ID*, *PW*}.
2. C generates the timestamp T and a random nonce x. He then computes N=g^x mod p, B'=($B \cdot g^{-\mathrm{H}(PW)}$ mod p)$\cdot N$ and V=H(T, B').
3. C sends {*ID*, *V*, *T*, *N*} to S.

(2) Authentication phase

When receiving the message {*ID*, *V*, *T*, *N*}, S runs the following steps to verify the legitimacy of C and negotiates the session key.

1. S generates the timestamp T', checks *ID* and compares if $T'-T$ is less than $\triangle$T. If ID is invalid or $T'-T > \triangle$T, the login request is rejected.
2. S generates a random nonce y and calculates M=g^y mod p, B''=($g^{\mathrm{H}(s,\ ID)}$ mod p)$\cdot N$ mod p and H(T, B''). S checks to see if H(T, B'') is equal to V. If so, C is authentic. S

computes U=H(M, B'') and sends {M, U} to C.

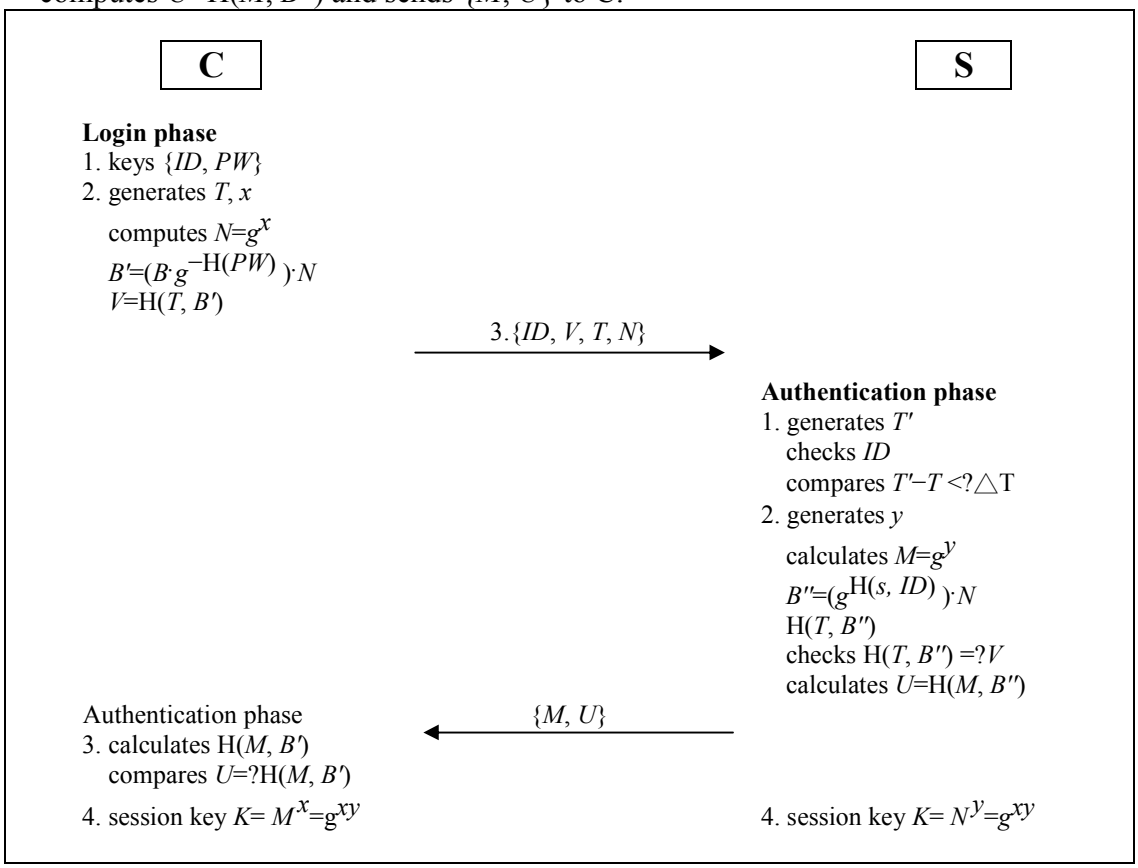

Fig. 7. The improvement for authentication only

3. After receiving the message, C calculates H(M, B'), and compares if U is equal to H(M, B'). If it is, S is authentic.
4. After successful mutual authentication, S has the session key $K = N^y = g^{xy}$ and C also has the same session key $K = M^x = g^{xy}$.

(B) For authentication and password change

(1) Login phase

Although in the password change protocol of I. E. Liao *et al.*, C can change his password without communicating with S. However, T. Xiang *et al.* found it still suffers from Denial-of-Service as described in Section 3.2.3. In the following, we propose an improvement to resist such an attack.

Assume that C wants to change his password PW to PW'.

1. C inserts his smart card and keys {ID, PW}.
2. C generates the timestamp T, a random nonce x and computes $N=g^x$ mod p, $B'=(B\cdot g^{-H(PW)} \bmod p)\cdot N$ and V=H(T, B'). He chooses new password PW' and

calculates $Y= g^{H(PW')} \bmod p$, $Z = B \cdot g^{-H(PW)} \bmod p$, $VP = H(Y, T, B')$ and $Y \oplus Z$.

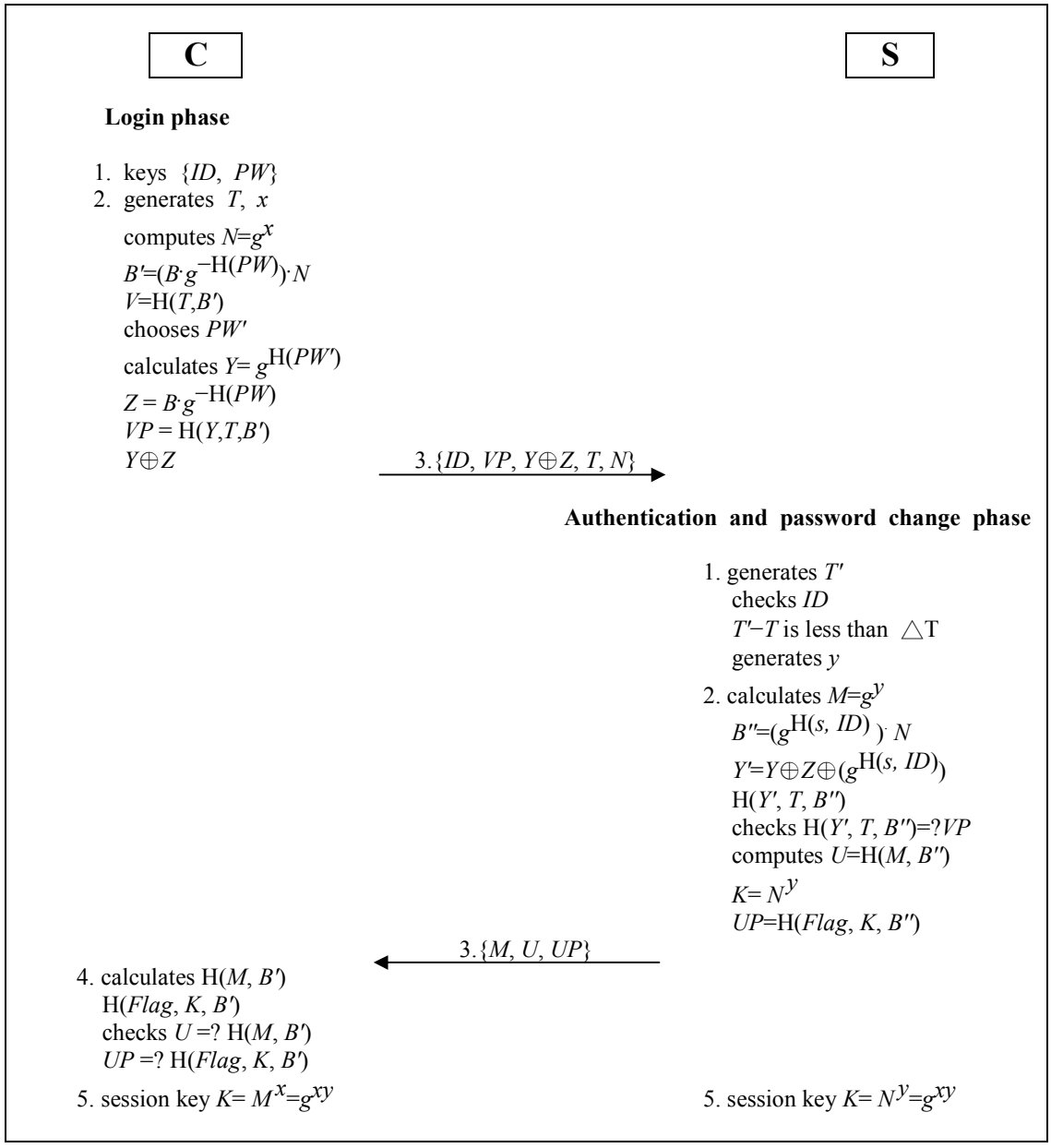

Fig. 8. The improvement for authentication and password change

3. C sends $\{ID, VP, Y \oplus Z, T, N\}$ to S.

(2) Authentication and password change phase

When receiving the message $\{ID, VP, Y \oplus Z, T, N\}$ from C, S executes the following steps to identify C, accepts the login request for password change if C is legal and constructs the session key.

1. After receiving the message, S generates the timestamp T', checks ID and compares if

$T'-T$ is less than $\triangle$T. If *ID* is valid and $T'-T < \triangle$T, the login request can be continued.

2. S generates a random nonce y, calculates $M=g^y \bmod p$, $B''=(g^{H(s, ID)} \bmod p) \cdot N \bmod p$, $Y'=Y \oplus Z \oplus (g^{H(s, ID)} \bmod p)$ and $H(Y', T, B'')$. He checks if $H(Y', T, B'')$ is equal to *VP*. If it isn't, S refuses the request and terminates the protocol. Otherwise, C is authentic and S accepts the password change request. S then computes $U=H(M, B'')$, $K=N^y \bmod p$, and $UP=H(Flag, K, B'')$ where *Flag* is set to '*accept*' after the password change request is accepted.
3. S sends $\{M, U, UP\}$ to C.
4. After receiving the message, C calculates $H(M, B')$ and $H(Flag, K, B')$. C compares if U is equal to $H(M, B')$. If it is, S is authentic. C then compares *UP* with the value of $H(Flag, K, B')$. If they are equal, C can confirm that his password change request is accepted.
5. After successful mutual authentication, S computes the session key $K= N^y= g^{xy}$ and C also can compute the same session key $K= M^x= g^{xy}$.

5. Security and efficiency analysis for both of the improved protocols on (a) M. Hölbl *et al.*'s protocol and (b) I. E. Liao *et al.*'s protocol, respectively.

In this section, we first show that both of our improvements can withstand various attacks in Section 5.1. Then, we examine the efficiency of both schemes in Section 5.2. For abbreviation, we make the analysis behind notation (a) to denote that it is for M. Hölbl *et al.*'s protocol and the analysis behind notation (b) to stand for it is for I. E. Liao *et al.*'s protocol. Moreover, notations (1) and (2) following notation (b) stand for they are for authentication only and for authentication and password change, respectively.

5.1 Security analysis

Our improved protocols not only can provide mutual authentication and perfect forward secrecy but also can resist the following attacks: off-line password guessing attack, insider attack, replay attack, on-line password guessing attack, Denial-of-Service attack on the password change phase and user impersonation attack if an attacker obtains the smart card. We show them in turn.

5.1.1 Mutual authentication

(a) For authenticating S, C has to verify the validity of the evidence $sat=H(g^{xy}, IDPW\text{-}dig, r_C, r_S)$. Conversely, for authenticating C, S must check the validity of $rsc=H(g^{xy}, IDPW\text{-}dig, r_S, r_C)$. For only S and C can know o deduce the common secret data, g^{xy}, *IDPW-dig*, and r_S, no one else can forge the valid evidences. In other words, after the validities of *sat* and *rsc* are verified by C and S respectively, the mutual authentication in our protocol is achieved.

(b) (1)In order to authenticate C in the phases for authentication only, S has to verify validity of the evidence V=H(T, B'). On the other hand, C must check the validity of U=H(M, B'') to authenticate S. These evidences are computed with the common secret message B'/B''. Because only C and S know the common secret message B'/B'', no one else can forge the evidences. When the validity of V and U is verified by S and C respectively, the mutual authentication between them is achieved.

(2) In order to authenticate C in the phases for authentication and password change, S has to verify validity of the evidence VP = H(Y, T, B'). On the other hand, C must check the validity of U=H(M, B'') and UP=H($Flag$, K, B'') to authenticate S. These evidences are computed with the common secret message B'/B''. Because only C and S know the common secret message B'/B'', no one else can forge the evidences. When the validity of VP, U and UP is verified by S and C respectively, the mutual authentication between them is achieved.

5.1.2 Perfect forward secrecy

In both of the improved protocols, a compromised password can't be used to construct previous session keys for we use the Diffie-Hellman key agreement protocols which are based on large random nonces. Naturally, it provides perfect forward secrecy.

5.1.3 Preventing the off-line password guessing attack

(a) If an adversary E has eavesdropped on the transmitted messages {ID, r_C, m-g^x}, { m-g^y, ch, sat} and {rsc } between C and S, he can't know the values of g^x, g^y and r_S from those intercepted messages to perform the off-line password guessing attack for m-g^x=$g^x \oplus$H(ID, $IDPW$-dig), m-g^y=$g^y \oplus$H(ID, $IDPW$-dig), ch= $r_S \oplus$ H(g^{xy}, $IDPW$-dig, r_C), sat=H(g^{xy},$IDPW$-dig, r_C, r_S) and rsc=H(g^{xy},$IDPW$-dig, r_S, r_C). Without the knowledge of $IDPW$-dig, E can not figure out g^x and g^y. Even he can figure out g^x and g^y, he can by no means figure out g^{xy} without the knowledge of x or y. Therefore, E can't implement the off-line password guessing attack.

(b) Supposing that C's smart card is lost, E can read the value of B. But he still can't get the value of $g^{H(s,\ ID)}$ for s is the secret of S. Hence, E can't launch the off-line password guessing attack by guessing password PW as PW' and verifying whether $B \cdot g^{-\mathrm{H}(PW')}$ is equal to $g^{\mathrm{H}(s,\ ID)}$.

(1)In the phases of authentication only, assume that E intercepted the message {V, T, N}, where $N = g^x$, V = H(T, B'), B'=$(B \cdot g^{-\mathrm{H}(PW)}) \cdot N$. However, he doesn't know both the values of B stored in C's smart card, and C's password. Therefore, he can't compute the value of H(T, $(B \cdot g^{-\mathrm{H}(PW)}) \cdot N$) and compare the intercepted value V with the computed result. Thus, the off-line password guessing attack can't work.

(2)In the phases of authentication and password change, assume that E intercepted the message $\{VP, Y \oplus Z, T, N\}$, where $N = g^x$, $Z = B \cdot g^{-\mathrm{H}(PW)}$, $Y = g^{\mathrm{H}(PW')}$, $VP = \mathrm{H}(Y, T, B')$, $B' = (B \cdot g^{-\mathrm{H}(PW)})$. But he doesn't know the values of B stored in C's smart card, the value of PW which is kept secret by C and the value of PW' chosen by C. Hence, he can't compute the values of Y and $\mathrm{H}(Y, T, B \cdot g^{-\mathrm{H}(PW)})$, and compare the intercepted VP with this computed result. Therefore, the off-line password guessing attack can't work.

5.1.4 Preventing the insider attack

(a) If a legal client D wants to impersonate client C to login to S. without the knowledge of C's *IDPW-dig*, the g^x he computes would be different with the value S will deduce. Hence, the value *rsc* which D will produce in pass three would be different from the value *rsc'* computed by S. That is, he can't be authenticated by S. Therefore, the insider attack fails.

(b) (1)Similarly, if a legal client D wants to impersonate client C to login to S. Without the knowledge of C's password *pw* and *B*, he can not deduce *V* and be successfully authenticated by S.

(2)With the same reason, if a legal client D wants to impersonate client C to login to S. Without the knowledge of C's password and B, he can not deduce *VP* and be successfully authenticated by S.

5.1.5 Preventing the replay attack

(a) We use random nonces r_C, r_S, x, y to prevent replay attack. An attacker cannot be authenticated by resending previous messages transmitted by a legal client.

(b) Similarly, an adversary cannot be authenticated by resending previous messages transmitted by a legal client for we use random nonces x, y and the timestamp T to withstand this kind of attack.

5.1.6 Preventing the on-line password guessing attack

The two protocols we proposed are mutual authentication between C and S. We can set both the protocols to tolerant three times of wrong password logins. If this amount of wrong login times is achieved, the system would reject the logins. Under such a setting, both of our schemes can resist on-line password guessing attack.

5.1.7 Preventing Denial-of-Service attack on password change

For both of our improvements provide mutual authentications, the password change request can accepted only after successful mutual authentications. Consequently, these

two improvement schemes can resist against Denial-of-Service attack.

5.1.8 Preventing user's impersonation attack if an attacker obtains the smart card

(a) We don't examine this protocol, for the protocol using no smart cards.

(b) (1)In the phases of authentication only, if E has got C's smart card and knows B. He starts the authentication protocol for being authenticated by S. However, he doesn't know C's password pw. He can not deduce C's B' and henceforth V which will be verified by S. Therefore, he couldn't be authenticated by S successfully.

(2)Similarly, in the phases of authentication and password change, assume that E has got C's smart card and knows B. He starts the authentication protocol for being authenticated by S. However, he faces the same reason as stated in (a) that he doesn't know C's password pw He can not deduce C's B' and henceforth VP. That is, he can not authenticated by S.

5.2 Efficiency analysis

(a) M. Hölbl *et al.*'s protocol needs four passes to establish the secure communication channel. However, our improvement needs only three passes. Therefore, our scheme is more efficient than theirs.

(b) I. E. Liao *et al.*'s protocol needs three passes to establish the secure communication channel. However, our improvement needs only two passes either for authentication only or for authentication and password change. Consequently, our scheme outperforms theirs in efficiency.

6. Conclusion

We have analyzed the security of M. Hölbl *et al.*'s protocol and review the cryptanalysis on I. E. Liao *et al.*'s protocol. Although M. Hölbl *et al.* claimed their protocol can resist against various attacks, we have showed that their protocol is indeed insecure against the password guessing attack. In addition, we have proposed improved protocols for both schemes which not only can provide mutual authentication efficiently, but also can really withstand various attacks.

References

[1]T. H. Chen, W. B. Lee, "A new method for using hash functions to solve remote user authentication", *Computers & Electrical Engineering*, Vol. 34, No. 1, pp. 53-62, January 2008.

[2]G. Yang, D. S. Wong, H. Wang, X. Deng, "Two-factor mutual authentication based on smart cards and passwords", *Journal of Computer and System Sciences*, In Press, Corrected Proof, Available online 15 May 2008.

[3]J. S. Chou, Y. Chen, T. H. Chen, "A Novel Password Authentication Scheme Based

On ID-based", unpublished.
[4]H. S. Rhee, J. O. Kwon, D. H. Lee, "A remote user authentication scheme without using smart cards", *Computer Standards & Interfaces*, In Press, Corrected Proof, Available online 5 December 2007.
[5]T. Goriparthi, M. L. Das, A. Saxena, "An improved bilinear pairing based remote user authentication scheme", *Computer Standards & Interfaces*, In Press, Corrected Proof, Available online 5 December 2007.
[6]J. Y. Liu, A. M. Zhou, M. X. Gao, "A new mutual authentication scheme based on nonce and smart cards", *Computer Communications*, Vol. 31, No. 10, pp. 2205-2209, June 2008.
[7]W. S. Juang, W. K. Nien, "Efficient password authenticated key agreement using bilinear pairings", *Mathematical and Computer Modelling*, Vol. 47, No. 11-12, pp. 1238-1245, June 2008.
[8]T. Xiang, K. Wong, X. Liao, "Cryptanalysis of a password authentication scheme over insecure networks", *Computer and System Sciences*, Vol. 74, No. 5, pp. 657-661, August 2008.
[9]J. S. Chou, Y. Chen, X. W. Hou, "Cryptanalysis and improvement of Yang-Wang password authentication schemes", unpublished.
[10]M. Hölbl, T. Welzer, B. Brumen, "Improvement of the Peyravian-Jeffries's user authentication protocol and password change protocol", *Computer Communications*, Vol. 31, No. 10, pp. 1945-1951, June 2008.
[11]I. E. Liao, C. C. Lee, M. S. Hwang, "A password authentication scheme over insecure networks", *Journal of Computer and System Sciences*, Vol. 72, No. 4, pp. 727-740, June 2006.
[12]M. Peyravian, C. Jeffries, "Secure remote user access over insecure networks", *Computer Communications*, Vol. 29, No. 5, pp. 660-667, March 2006.
[13]W. S. Juang, S. T. Chen, H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards", *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 6, pp. 2551-2556, June2008.
[14]C. S. Bindu, P. C. S. Reddy, B. Satyanarayana, "Improved remote user authentication scheme preserving user anonymity", *International Journal of Computer Science and Network Security*, Vol. 8, No. 3, pp. 62-65, March 2008.
[15]Y. Lee, J. Nam, D. Won, "Vulnerabilities in a Remote Agent Authentication Scheme Using Smart Cards", *LNCS: AMSTA*, Vol. 4953, pp. 850-857, April 2008.
[16]M. L. Das, "On the Security of an efficient and complete remote user authentication scheme", *Cryptography and Security*, arXiv:0802.2112v1 [cs.CR], February 2008.
[17]H. Guo, Z. Li, Y. Mu, X. Zhang, "Cryptanalysis of simple three-party key exchange protocol", *Computers & Security*, Vol. 27, No. 1-2, pp. 16-21, March 2008.
[18]H. B. Chen, T. H. Chen, W. B. Lee, C. C. Chang, "Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks", *Computer Standards & Interfaces*, Vol. 30, No. 1-2, pp. 95-99, January 2008.
[19]E. J. Yoon, K. Y. Yoo, "Improving the novel three-party encrypted key exchange protocol", *Computer Standards & Interfaces*, Vol. 30, No. 5, pp. 309-314, July 2008.
[20]J. Nam, Y. Lee, S. Kim, D. Won, "Security weakness in a three-party pairing-based protocol for password authenticated key exchange", *Information Sciences*, Vol. 177,

No. 6, pp. 1364-1375, March 2007.
[21]H. R. Chung, W. C. Ku, “Three weaknesses in a simple three-party key exchange protocol”, *Information Sciences*, Vol. 178, No. 1-2, pp. 220-229, January 2008.
[22]R. C. Phan, W. C. Yau, B. M. Goi, “Cryptanalysis of simple three-party key exchange protocol (S-3PAKE)”,*Information Sciences*, Vol. 178, No. 13, pp. 2849-2856, July 2008.
[23]C. C. Chang, J. S. Lee, T. F. Cheng, “Security design for three-party encrypted key exchange protocol using smart cards”, *ACM ISBN: 978-1-59593-993-7*, pp. 329-333, 2008.
[24]J. L. Tsai, “Efficient multi-server authentication scheme based on one-way hash function without verification table”, *Computers & Security*, Vol. 27, No. 3-4, pp. 115-121, May-June 2008.
[25]Y. Liao, S. S. Wang, “A secure dynamic ID based remote user authentication scheme for multi-server environment”, *Computer Standards & Interfaces*, In Press, Corrected Proof, Available online 17 October 2007.