Comment on four two-party authentication protocols

Yalin Chen¹, Jue-Sam Chou²*, Chun-Hui Huang³

¹ Institute of information systems and applications, National Tsing Hua University
d949702@oz.nthu.edu.tw
² Department of Information Management, Nanhua University, Taiwan, R.O.C
*; corresponding author
jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536
³ Department of Information Management, Nanhua University, Taiwan, R.O.C
g6451519@mail.nhu.edu.tw

Abstract

In this paper, we analyze the protocols of Bindu et al., Goriparthi et al., Wang et al. and Hölbl et al.. After analyses, we found that Bindu et al.’s protocol suffers from the insider attack if the smart card is lost, both Goriparthi et al.’s and Wang et al.’s protocols can’t withstand the DoS attack on the password change phase which makes the password invalid after the protocol run, and Hölbl et al.’s protocol is vulnerable to the insider attack since a malevolent legal user can deduce KGC’s secret key xₙ.

Keywords: password authentication protocol, insider attack, denial-of-service attack, smart card lost problem, mutual authentication, man-in-the-middle attack

1. Introduction

Authentication protocols provide two entities to ensure that the other party is the intended one whom he attempts to communicate with and can be examined by using three factors: type, efficiency and security. Generally speaking, authentication protocols can be divided into two types. One is password-based that can make a user be authentic to a remote entity by using his human-rememberable password. And the other is public key cryptography-based that makes a user can be authenticated by using his private key instead of password. In a password-based protocol, a user registers at the remote server to become a legal user for accessing the server’s resource and the server maintains a password table for authenticating valid users. However, for avoiding the stolen-verifier attack, the server usually issues a smart card to the registered user to get rid of storing a password table. Thereafter, the user can take use of his password and the smart card to logon the server. In a public key-based system, users have to register at KGC (Key Generation Center) for obtaining their public keys and corresponding private keys. Then, they can be authenticated by the intended entity using their private keys. For improving the efficiency of key
management in an authentication protocol, an identity-based cryptosystem is usually used in which KGC issues a private key to a registering user and uses the user’s identity as his public key. As to other efficiency considerations, such as computational and communicational overhead, researchers generally reduce the computational load of a protocol by using simple techniques, such as secure one-way hash functions and symmetric key encryptions, as much as possible. Of course, asymmetric key encryptions which are less efficient in computations (i.e., RSA, ECC, ElGamal, and bilinear pairings) are used as well. As for giving thought to the communicational overhead, researchers usually do their best to reduce the number of passes (for instance, to only two), since it is the dominant factor in considering the efficiency in a protocol. The most important feature of an authentication protocol is its security since it provides two entities to authenticate each other through an insecure network. Attackers may eavesdrop, modify or intercept the messages to and from in a communication channel to collect and deduce some meaningful information to defraud the other party. Hence, the transmitted messages must be dealt with some techniques to prevent from various attacks, such as password guessing attack, impersonate attack, insider attack, man-in-the-middle attack, and so on.

Recently from 2002 to 2010, many studies [1-41] were proposed to secure authentication protocols. In 2008, Bindu et al. proposed an improved protocol [14] on Chien et al.’s scheme [3]. Their protocol is a smart card based password authentication protocol and operates with symmetric key encryption algorithm. They claimed that their protocol is secure, can achieve user anonymity, and prevent various attacks, such as replay attack, stolen-verifier attack, password guessing attack, insider attack, and man-in-the-middle attack. In 2009, Goriparthi et al. proposed a scheme [27] which is improved from Das et al.’s protocol [2] and can avoid the weakness existing in Chou et al.’s [5] (also modified from Das et al.’s). Goriparthi et al.’s protocol is also a smart card based password authentication protocol and bases on bilinear pairings. They claimed that their protocol is secure and can withstand replay attack and insider attack. In the same year, Wang et al. [31] proposed an improvement on Das et al.’s protocol [2]. Their scheme is a smart card based password authentication protocol as well and operates with secure one-way hash function. They claimed that their protocol is secure and can achieve mutual authentication. Also in 2009, Hölbl et al. [40] improved two identity-based authentication protocols, Hsieh et al. [1] and Tseng et al. [8]. Their protocols are neither password-based nor smart card based protocols. They are identity-based public key cryptosystem and operate with ElGamal signature scheme. Hölbl et al. claimed the protocols are not only efficient but also secure. Although all of the above schemes mentioned claimed that they are secure; however, in this paper, we found some threats existing in them, correspondingly. We will show
them in turn.

The remainder of this paper is organized as follows: In Section 2, we review and
attack on the protocol of Bindu et al. [14]. Then, we review and attack on the
protocols of Goriparthi et al. [27], Wang et al. [31], and Höbl et al. [40] in Section 3
through 5, respectively. Finally, a conclusion is given in Section 6.

2. Review and attack on the improvement of Bindu et al.

In this section, we first review Bindu et al.’s scheme [14] in Section 2.1 then show
the insider attack launched by an insider who is supposed to have obtained another
legal user’s smart card in Section 2.2.

2.1 Review of Bindu et al.’s scheme

There are three phases in Bindu et al.’s scheme: the registration phase, the login
phase and the authentication phase.

In the registration phase, the server S issues to legal user i a smart card which
contains $m_i$ and $I_i$, where $m_i=H(ID_i\oplus s)\oplus H(s)\oplus H(PW_i)$, $I_i=H(ID_i\oplus s)\oplus s$, and $s$ is S’s
secret key.

When i wants to login to S, he starts the login phase by computing $r_i=g^x$ ($x$ is a
random number chosen by i), $M=m_i\oplus H(PW_i)$, $U=M\oplus r_i$, $R=I_i\oplus r_i=H(ID_i\oplus s)\oplus s\oplus r_i$,
and $E_R[r_i,ID_i,T]$ (T is a timestamp, and $E_R[r_i,ID_i,T]$ is a ciphertext encrypted by the
secret $R$). He then sends \{U, T, E_R[r_i,ID_i,T]\} to S.

In the authentication phase, after receiving \{U, T, E_R[r_i,ID_i,T]\} at timestamp $T_s$, S
computes $R=U\oplus H(s)\oplus s =M\oplus r_i\oplus H(s)\oplus s =m_i\oplus H(PW_i)\oplus r_i\oplus H(s)\oplus s = H(ID_i\oplus s)$
$\oplus H(s)\oplus H(PW_i)\oplus H(PW_i)\oplus r_i\oplus H(s)\oplus s = H(ID_i\oplus s)\oplus r_i\oplus s$, decrypts $E_R[r_i,ID_i,T]$, checks to see if $T_s-T$ is less than $\triangle T$, and compares R with $H(ID_i\oplus s)\oplus s\oplus r_i$ to see if they are equal. If they are, he sends \{ $T_s$, $E_R[r_s,r_t+1,T_s]$ \} to i, where $r_s=g^y$ and $y$ is a
random number chosen by S. After that, i verifies the validity of the time interval,
decrypts $E_R[r_s,r_t+1,T_s]$, and checks to see if $r_t+1$ is correct or not. If it is, S is
authentic. Then, i sends \{ $E_{K_{res}}[r_s+1]$ \} to S, where $K_{res}=r_s^x=g^{xy}$. S decrypts the message
and checks to see if the value of $r_s+1$ is correct or not. If it is, i is authentic.

2.2 Attack on Bindu et al.’s scheme

If C lost his smart card and the card is got by an insider E, E can impersonate C to
log into S. We show the attack in the following.

For that C’s smart card stores $m_e=H(ID_e\oplus s)\oplus H(s)\oplus H(PW_e)$ and $I_e=H(ID_e\oplus s)\oplus s$,
and E’s smart card stores $m_r=H(ID_r\oplus s)\oplus H(s)\oplus H(PW_r)$ and $I_r=H(ID_r\oplus s)\oplus s$,
suppose E gets C’s smart card but doesn’t have the knowledge of PW_e, E can choose a
random number \( x \) and computes \( r_c=g^x \), \( V= m_c \oplus I_c \oplus H(PW_c)=H(s) \oplus S \), \( M=I_c \oplus V=H(ID_c \oplus S) \oplus S=H(ID_c \oplus S) \oplus H(s) \) which equals \( m_c \oplus H(PW_c) \), \( U=M \oplus r_c \), and \( R=I_c \oplus r_c \). Then, E masquerades as C by sending \( \{U, T, E_R[r_c, ID_c, T]\} \) to S. After receiving the message, S computes \( R=U \oplus H(s) \oplus S \) and compares \( R \) with \( H(ID_c \oplus S) \oplus r_c \). If they are equal, S sends C the message \( \{T_s, E_R[r_c, r_s+1, T_s]\} \). E intercepts the message, decrypts \( E_R[r_c, r_s+1, T_s] \), and uses \( r_s \) to compute \( K_{us}=r_s^{xy} \). E then can send a correct message \( \{E_{ks}[r_s+1]\} \) to S, to let S authenticate him as C. In other words, insider E can successfully launch an insider attack if the user’s smart card is lost.

More clarity, we demonstrate why \( R=U \oplus H(s) \oplus S \) is equal to \( H(ID_c \oplus S) \oplus S \oplus r_c \) by the following equations.

\[
\begin{align*}
R &= U \oplus H(s) \oplus S \\
&= M \oplus r_c \oplus H(s) \oplus S \quad \text{..........................................................} \quad \therefore \quad U = M \oplus r_c \\
&= I_c \oplus V \oplus r_c \oplus H(s) \oplus S \quad \text{..........................................................} \quad \therefore \quad M = I_c \oplus V \\
&= H(ID_c \oplus S) \oplus S \oplus V \oplus r_c \oplus H(s) \oplus S \quad \text{..........................................................} \quad \therefore \quad I_c = H(ID_c \oplus S) \oplus S \\
&= H(ID_c \oplus S) \oplus S \oplus r_c \oplus H(s) \oplus S \quad \text{..........................................................} \quad \therefore \quad V = H(s) \oplus S \\
&= H(ID_c \oplus S) \oplus S \oplus r_c
\end{align*}
\]

3. Review and attack on the protocol of Goriparthi et al.

In this section, we first review Goriparthi et al.’s scheme [27] in Section 3.1 then we demonstrate that it is vulnerable to the DoS attack on the password change phase which can make the password invalid after their protocol run in Section 3.2.

3.1 Review of Goriparthi et al.’s scheme

In the password change phase of Goriparthi et al.’s protocol, when client C wants to change his password \( PW \), he keys his \( ID \) and \( PW \) to his smart card. The smart card verifies \( ID \) (without verifying his password \( PW \)) to see if it is correct. If it is, the smart card will subsequently receive a new password \( PW^* \) submitted by C and compute \( Reg_{ID} = RegID - h(PW)+h(PW^*)= s h(ID) + h(PW^*) \), where \( RegID= s h(ID) + h(PW) \) is stored in C’s smart card, \( h(\cdot) \) is a map-to-point hash function \( h: \{0,1\}^* \rightarrow G_1 \), and \( G_1 \) is a group on an elliptical curve. Finally, the smart card will replace \( RegID \) with \( Reg_{ID} \).

3.2 Attack on Goriparthi et al.’s scheme

In the protocol, assume that there is an attacker temporarily gets access to C’s smart card. He randomly selects two passwords \( PW' \) and \( PW'' \) as the old and the new ones, respectively. The smart card will then compute \( Reg_{ID} = RegID - h(PW') + h(PW'') = s h(ID) + h(PW') + h(PW'') \) and replace \( RegID \) with \( Reg'_{ID} \). This
would make C’s password $PW$ invalid.

4. Review and attack on the protocol of Wang et al.

In this section, we first review Wang et al.’s scheme [31] in Section 4.1, then demonstrate that it is vulnerable to the DoS attack on the password change phase which can make the password invalid after the protocol run in Section 4.2.

4.1 Review of Wang et al.’s protocol

In Wang et al.’s protocol, C inserts his smart card, keys $PW$, and requests to change the password $PW$ to a new one $PW^*$. Then, the smart card computes $N_i^* = N_i \oplus H(PW) \oplus H(PW^*)$ and replaces $N_i$ with $N_i^*$, where $N_i = H(PW_i) \oplus H(x)$ is stored in C’s smart card, $PW_i$ is chosen by the user when he registers at the remote server $S$, and $x$ is $S$’s secret key.

4.2 Attack on Wang et al.’s protocol

Obviously, this protocol also exits the same security loophole as does in [27]. Since if an attacker temporarily gets access to C’s smart card and reads the value of $N_i$, he can use two random values $PW''$ and $PW'''$ to compute $N_i' = N_i \oplus H(PW'') \oplus H(PW''')$ and replace $N_i$ with $N_i'$. From then on, client C can never pass the authentication and the attack succeeds.

5. Review and attack on the protocol of Hölbl et al.

Hölbl et al. [40] proposed two improvements of two-party key agreement protocols. In the following, we first briefly review then present our attack on both of their protocols, respectively.

5.1 Review of Hölbl et al.’s first protocol

Hölbl et al.’s first protocol consists of three phases: the system setup phase, the private key extraction phase, and the key agreement phase.

In the system setup phase, KGC chooses a random number $x_s$ and keeps it secret. He computes $y_s = g^{x_s}$ and publishes it.

In the private key extraction phase, with each user having his identity $ID$, KGC selects a random number $k_i$, and calculates i’s private key $v_i = I_i k_i + x_s u_i \pmod{p-1}$ and public key $u_i = g^{k_i} \pmod{p}$, where $I_i = H(ID_i)$.

In the key agreement phase, user A chooses a random number $r_a$, computes $t_a = g^{r_a}$, and then sends $\{ u_a, t_a, ID_a \}$ to user B. After receiving $\{ u_a, t_a, ID_a \}$, B chooses a
random number $r_b$, calculates $t_b=g^{r_b}$, and then sends $\{u_b, t_b, ID_b\}$ back to A. Finally, A and B can compute their common session key, $K=(u_b I_b y_I b t_b)(v_a+r_a)=g^{(v_b+r_b)(v_a+r_a)}$ and $K=(u_a I_a y_I a t_a)(v_b+r_b)=g^{(v_b+r_b)(v_a+r_a)}$, respectively, where $I_a=H(ID_a)$ and $I_b=H(ID_b)$.

5.2 Attack on Hölbl et al.’s first protocol

Assume that an insider C calculates $I_c=H(ID_c)$ and $q=gcd(I_c, u_c)$, and computes $w=I_c q, z= u_c q$, and $j= v_c q$, where $v_c$ is C’s private key. Hence, $gcd(w, z)=1$. Then, he can use the extended Euclid’s algorithm to find $\alpha$ and $\beta$ both satisfying that $\alpha w+\beta z =1$. As a result, he can obtain both $x_s$ and $k_s$, since $v_c=1 j_c:q_c= (\alpha w+\beta z)j_c q_c=(\alpha I_c q+\beta u_c)j q=(\alpha I_c+\beta u_c)j =I_c (\alpha j)+ (\beta j) u_c$ and $v_c=I_c k_s +x_s u_c$, where $x_s$ is KGC’s secret key and $k_s$ is a random number selected by KGC satisfying $u_c=g^{k_s}$. More clearly, the value $x_s$ he obtains is equal to $\beta/j$.

After obtaining $x_s$, C can deduce any user’s private key in the same manner. As an example, in the following, we demonstrate how C can deduces i’s private key, $k_i$. C calculates $I_i=H(ID_i)$ and $q_i=gcd(I_i, u_i)$, computes $w_i=I_i q_i$ and $z_i=u_i q_i$, and then uses the extended Euclid’s algorithm to compute $\gamma$ and $\varepsilon$ satisfying that $\gamma w_i+\varepsilon z_i=1$. Finally, since $v_i=1 j_i:q_i= (\gamma I_i q_i+\varepsilon u_i)j_i q_i=(\gamma I_i+\varepsilon u_i)j_i =I_i (\gamma j_i)+ (\varepsilon j_i) u_i$ and $v_i=I_i k_i +x_j u_i$, he can calculate $j_i=x_i j_i$ and thus obtains i’s private key by computing $v_i=j_i q_i$. With the knowledge of i’s private key, insider C can impersonate user i to communicate with any other legal user. That is, to a minimum, an insider attack exists.

5.3 Review of Hölbl et al.’s second protocol

Hölbl et al.’s second protocol consists of three phases: the system setup phase, the private key extraction phase, and the key agreement phase.

The system setup phase of this protocol is the same as the one in the first protocol.

In the private key extraction phase, with each user having his identity ID, KGC selects a random number $k_i$, and calculates i’s private key $v_i=k_i+x_i H(ID_i, u_i)$ and public key $u_i=g^{k_i}$.

In the key agreement phase, user A chooses a random number $r_a$, computes $t_a=g^{r_a}$, and then sends $\{u_a, t_a, ID_a\}$ to user B. After receiving $\{u_a, t_a, ID_a\}$, B chooses a random number $r_b$, calculates $t_b=g^{r_b}$, and then sends $\{u_b, t_b, ID_b\}$ to A. Finally, A and B can compute their common session key, $K=(u_b y_s H(ID_b, u_b) t_b)(v_a+r_a)=g^{(v_b+r_b)(v_a+r_a)}$ and $K=(u_a y_s H(ID_a, u_a) t_a)(v_b+r_b)=g^{(v_a+r_a)(v_b+r_b)}$, respectively.

5.4 Attack on Hölbl et al.’s second protocol
Likewise, we can launch the same attack, as do in the first one, on this scheme. Since gcd(1, H(IDc, u))=1, an insider C can use the extended Euclid’s algorithm to find α and β both satisfying that α1+βH(IDc, u)=1. And since v i=κ+iH(IDc, u) and 1=(κ/v i)1+(x i/v i)H(IDc, u), he can obtain both x i and κ i by letting x i=βv i and κ i=αv i, where v i is C’s private key, x i is KGC’s secret key and κ i is a random number selected by KGC satisfying u i=gκ i. Consequently, similar to the result as shown in the attack of the first protocol, insider C can impersonate user i to communicate with any other legal user. That is, to the minimum, there exists an insider attack in their second scheme. Therefore, the protocol fails.

6. Conclusion

We have analyzed the protocols of Bindu et al. [14], Goriparthi et al. [27], Wang et al. [31], and Hölbl et al. [40]. After analyses, we found that Bindu et al.’s suffers from the insider attack if the smart card is lost, Goriparthi et al.’s and Wang et al.’s can’t withstand the DoS attack on the password change phase which can make the password invalid after the protocol run, and Hölbl et al.’s are vulnerable to the insider attack since a malevolent legal user can deduce KGC’s secret key x s.

References


