Cryptanalysis of an Exquisite Mutual Authentication Scheme with Key Agreement Using Smart Card

He Debiao*, Chen Jianhua, Hu Jin
School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China

Abstract*:* The weakness of an exquisite authentication scheme based on smart cards and passwords proposed by Liao et al. [C. H. Liao, H. C. Chen, and C. T. Wang, An Exquisite Mutual Authentication Scheme with Key Agreement Using Smart Card, *Informatica*, Vol. 33, No. 2, 2009, 125-132.] is analyzed. Five kinds of weakness are presented in different scenarios. The analyses show that Liao et al.'s scheme is insecure for practical application.
Key words: Authentication; Security; Cryptanalysis; Smart card; Attacks.

1. Introduction

Remote authentication is a method to authenticate remote users over insecure communication channel. Password-based authentication schemes have been widely deployed to verify the legitimacy of remote users. Recently, many password authentication schemes using smart cards have been proposed by some researchers. In these schemes, the smart-card-oriented remote login authentication scheme is used to authenticate a legitimate user. The smart card contains a microprocessor, which can perform arithmetic operations quickly, an I/O port, a RAM, and a ROM in which some messages are stored. Therefore, there is no need to store a password table or verification table in the server.

Recently, Liao et al.[1] proposed an authentication protocol using Diffie-Hellman scheme [2] to enhance the security level and efficiency but to reduce the computation load for a smart card. In their method, the smart card is responsible for simple computations and the server is responsible for complicated ones. Their scheme also uses the one-way hash function and the exclusive-or operation to maintain security and convenience. To prevent the replay attacks and the synchronization problem, they adopted the nonces in their scheme instead of using time-stamp. Furthermore, they introduced the design of transformed identity [3] in our scheme to avoid the duplication of identities. They claimed their scheme can withstand various attacks. In this paper, however, some security loopholes of their scheme will be pointed out and the corresponding attacks will be described.

The rest of the paper is organized as follows: Section 2 briefly reviews Liao et al.'s scheme. Section 3 elaborates the weakness of Liao et al.'s scheme. At last, section 4 concludes this paper.

*Corresponding author.
E-mail: hedebiao@163.com, *Tel*:+0086015307184927, *Fax*: +008602787817667

2. Review of Liao et al.'s Scheme

In order to facilitate future references, frequently used notations are listed below with their descriptions.

- p : a prime number;
- q : a prime number and $q \mid p-1$;
- G : a subgroup with order q in the group Z_p^*;
- g : a generator of the group G;
- U : a user;
- ID_U : U 's identifier;
- S : a remote server;
- PW_U : U 's password;
- x : S 's long secret key with length k;
- $h:\{0,1\}^* \rightarrow \{0,1\}^k$: a target collision resistant hash function;
- $\oplus$: bitwise XOR operation.
- ||: concatenation operation

Liao et al.'s scheme consists of four phases: registration phase, login-and-authentication phase, key agreement phase and password update phase. We describe them as follows.

2.1. Registration Phase

In this phase, everyone who wants to register at the server should obtain a smart card. The user U begins his registration at the server S as follows.

1). U freely chooses his password PW_U and identifier ID_U and sends it to S through a secure channel.

2). Upon receiving PW_U and ID_U, S computes the transformed identity $TID_U = TS_U \,||\, ID_U$, $A_U = h(TID_U \oplus x)$, $B_U = (g^{A_U} \bmod p) \oplus PW_U$, where TS_U is of the registration time. Then S stores TS_U, B_U and $h(\cdot)$ into a smart card. At last, S issue the smart card to U.

2.2. Login-and-authentication phase

In this phase, the user U sends a login request message to the server S whenever U wants to access some resources upon S. Then the server S verifies the authenticity of the login message requested by the user U.

1) U inserts his smart card into a smart card reader and then inputs his ID_U and PW_U. U's smart card generates a nonce n_U and computes $TID_U = TS_U \| ID_U$, $NTID_U = TID_U \oplus n_U$ and $C_U = h(B_U \oplus PW_U) \oplus n_U$. Then U's smart card sends $M_1 = \{ID_U, NTID_U, C_U\}$ to the server S.

2) Upon receiving the message M_1, S checks the validity of ID_U. If ID_U is not valid, S aborts the current session. Otherwise, S computes $TID_U = TS_U \| ID_U$, $n'_U = TID_U \oplus NTID_U$, $A_U = h(TID_U \oplus x)$, $h(g^{A_U} \bmod p)$ and $n''_U = h(g^{A_U} \bmod p) \oplus C_U$. S checks if n''_U equals n'_U. If n''_U does not equal n'_U, S stops the session. Otherwise, S generates a nonce n_S and computes $D_U = C_U \oplus n_U \oplus n_S$ and $NTID_S = TID_U \oplus n_S$. Then S sends $M_2 = \{D_U, NTID_S\}$ to the smart card.

3) Upon receiving the message M_2, U's smart card computes $n'_S = TID_U \oplus NTID_S$ and $n''_S = C_U \oplus n_U \oplus D_U$. If n''_S does not equal n'_S, U's smart card stops the session. Otherwise, U's smart card computes $E_U = (C_U \oplus n_U) \| (n_S + 1)$ and sends the message $M_3 = \{E_U\}$ to S.

4) Upon receiving the message M_3, S extracts $n_S + 1$ and finds n_S in there. At this time the server ensures that the authentication user does have the nonce n_S, and the user is authenticated.

2.3. Key agreement phase

After receiving the nonce, n_S, sent from the server, the user creates a session key $SK_U = h((B_U \oplus PW_U) \| n_S \| n_U)$. Once the server ensures that u has the nonce, ns, it generates a session key $SK_S = h((g^{A_U} \bmod p) \| n_S \| n_U)$. It easy to verify that SK_S equals SK_U.

2.4. Password change phase

1) U inserts his smart card into the smart-card reader of a terminal, enters the old password PW_U, and requests to change password. Next, U enters the new password PW_U^*.

2) U 's smart card computes $B_U^* = B_U \oplus h(PW_U) \oplus h(PW_U^*)$, and then replaces B_U with B_U^*.

3. Cryptanalysis of Liao et al.'s Scheme

To evaluate the security of Liao et al.'s scheme[1], we assume that an attacker may have the following capabilities.

(1) The attacker has total control over the communication channel between the user and the server. That is, the attacker may intercept, insert, delete, or modify any message in the channel. He also can store the message transmitted between the user the client into the database.

(2) The attacker may either (i) obtain a user's password, or (ii) extract the secret information of the smart card, but cannot achieve both (i) and (ii).

For Capability (2) (ii), it is important to note that breaching smart cards has been shown to be relatively quick and easy, allowing the secrets stored in a smart card to be revealed by monitoring the power consumption [4] or by analyzing the leaked information [5]. Although some smart card manufacturers have taken into account the risk of these attacks and provided countermeasures to defer the reverse engineering attempt, these smart cards are more costly. In most cases, due to the limited resources (e.g., cost, display sizes, computing capability) of mobile devices, many applications do not deploy this costly feature. Therefore, a better approach is to take into account smart card security breach when designing smart card based authentication schemes.

Obviously, it is trivial to see that if the attacker has both Capabilities (2) (i) and (2) (ii), there is no way to prevent the attacker from masquerading as the user. In this letter, we focus on the security of Liao et al.'s scheme for the case that the attacker has Capabilities (1) and (2) (ii).

3.1. Impersonation attack

In the Step 4) of the login and authentication phase, $E_U = (C_U \oplus n_U) \| (n_S + 1)$ is transmitted to the server directly, then the attacker A can extract $n_S + 1$ and finds n_S in easily. The attacker A can carry out the impersonation attack using the property. The detail of the attack is described as follows.

Phase I:

1) A intercept the message $M_1 = \{ID_U, NTID_U, C_U\}$, $M_2 = \{D_U, NTID_S\}$ and $M_3 = \{E_U\}$ transmitted between the user and the server.

2) A extracts $n_S + 1$ and finds n_S in $E_U = (C_U \oplus n_U) \| (n_S + 1)$.

3) A computers $TID_U = NTID_S \oplus n_S$ and $n_U = TID_U \oplus NTID_U$.

4) A computers $h(g^{A_U} \bmod p) = h(B_U \oplus PW_U) = C_U \oplus n_U$.

Phase II:

1) A generates a nonce $\overline{n_U}$ and computes $\overline{NTID_U} = TID_U \oplus \overline{n_U}$ and $\overline{C_U} = h(B_U \oplus PW_U) \oplus \overline{n_U}$. Then A sends $M_1 = \{ID_U, \overline{NTID_U}, \overline{C_U}\}$ to S.

2) It easy to say $M_1 = \{ID_U, \overline{NTID_U}, \overline{C_U}\}$ can pass the authentication of S. Then S generates a nonce n_S and computes $D_U = \overline{C_U} \oplus \overline{n_U} \oplus n_S = h(B_U \oplus PW_U) \oplus n_S$ and $NTID_S = TID_U \oplus n_S$. Then S sends $M_2 = \{D_U, NTID_S\}$ to the smart card.

3) Upon receiving the message M_2, A computes $n_S = h(B_U \oplus PW_U) \oplus D_U$ $\overline{E_U} = (\overline{C_U} \oplus \overline{n_U}) \| (n_S + 1) = h(B_U \oplus PW_U) \| (n_S + 1)$ and sends the message $M_3 = \{\overline{E_U}\}$ to S.

4) It easy to say $M_3 = \{\overline{E_U}\}$ can pass the authentication of the server, and A impersonate U successfully.

3.2. Password guessing attack

In password authentication schemes that the user is allowed to choose his password, the user

tends to choose a password that can be easily remembered for his convenience. However, these easy-to-remember passwords are potentially vulnerable to password guessing attack, in which an adversary can try to guess the user's password and then verify his guess. In general, the password guessing attack can be classified into online password guessing attack and offline password guessing attack. The adversary tries to use guessed passwords iteratively to pass the verification of the server in an online manner in online password guessing attack. While in off-line password guessing attack, the adversary intercepts some password-related messages exchanged between the user and the server, and then iteratively guesses the user's password and verifies whether his guess is correct or not in an offline manner. Online password guessing attacks can be easily thwarted by limiting the number of continuous login attempts within a short period. In an offline password guessing attack, since there is no need for the server to participate in the verification, the server cannot easily notice the attack. While in password authentication scheme using smart cards, two points should be noticed to resist this kind of attack. One is that the password should not be transmitted between the client and the server during the authentication; otherwise it has the risk of being intercepted and recovered. The other is that the sensitive data stored in smart cards should be well protected so that the password would not be leaked even if smart cards are lost and all the data inside are disclosed. Although Liao et al. claim that their scheme is secure even when the client's smart card is lost [1], an off-line password guessing attack method will be given here as a counter example.

Suppose the user's smart card is lost, an attacker A can read all the data, including TS_U and B_U from the smart card via physically access to the storage medium [4, 5]. Then A can carry out the password guessing attack as follows.

1) A guess the identity ID'_U and the password PW'_U.

2) A gets a record between the user and the server in the database according the identity ID'_i. The record includes $M_1 = \{ID_U, NTID_U, C_U\}$, $M_2 = \{D_U, NTID_S\}$ and $M_2 = \{E_U\}$ obviously.

3) A computes $TID'_U = TS_U \| ID'_U$, $n'_U = TID'_U \oplus NTID_U$ and $n''_U = h(B_U \oplus PW'_U) \oplus C_U$.

4) A checks if n'_U equals n''_U. If n'_U equals n''_U, then A finds the correct passwords. Otherwise, A repeats steps 1, 2 and 3 until the correct password is found.

From the description, we can see that the search space for the guessing attack is $|ID| \times |PW|$, where ID and PW are the set of possible passwords and possible identity separately. $|\cdot|$ represents the cardinality of a set. Note that generally $|ID|$ and $|PW|$ are not very big, and unlike a space for cryptographic key. The attacker also can carry out the above attack

using $M_2 = \{D_U, NTID_S\}$.

3.3. Insider's attack

In real environments, it is likely that the user uses the same password to access several servers for his convenience. If a privileged insider of S has learned the user's password, he may try to impersonate U to access other servers. In the user registration phase, U's password PW_U will be revealed to S because it is transmitted directly to S. Then, the privileged insider of S may try to access the servers outside this system. If the targeted outside server adopts the normal password authentication scheme, it is possible that the privileged insider of S can successfully impersonate U to login it by using PW_U. Although it is also possible that all the privileged insiders of S are trusted and U does not use the same password to access several servers, the implementers and the users of the scheme should be aware of such a potential weakness. Clearly, Liao et al.'s scheme is vulnerable to an insider attack.

3.4. Denial-of-service attack on password changing

In password authentication, DoS attack can cause permanent error on authentication by introducing unexpected data during the procedures of authentication. The most vulnerable procedure is the password changing phase since it usually refreshes the data on storage. If an attacker can modify the password, or tamper the message containing password with valid data format, the updated password or its related verification data will then be different from what the client expects. The client can never pass the subsequent authentication thereby. In Liao et al.'s scheme, the password changing phase is performed on the client terminal with smart cards, i.e., the client can change his password without communicating with the server [1]. This enhances the security of password changing as no sensitive message need to be transmitted over the insecure network. Meanwhile, it relieves the overhead of server. However, due to the drawbacks of design, it is still possible to load a DoS attack on password changing in their scheme.

Suppose an attacker temporarily gets access to the user U's smart card, he then inserts the card in a terminal device and performs the following operations. He randomly selects two different passwords PW' and PW'' as the old and the new password, respectively. Then he sends a changing password request to the smart card. As described in the previous section, the smart card will then compute $B_U^* = B_U \oplus h(PW') \oplus h(PW'')$, then it replaces B_U with B_U^*. From then on, U can never pass the password authentication by the server. This is because in the login phase, U cannot be verified by the server in the third step of authentication phase.

3.5. Session-Key Problem

Forward secrecy requires that, if long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities can be unaffected. Obviously, if the attacker A steals the user's smart card, extracts the values stored including TS_U and B_U in the smart card and obtains correct identity ID_U password PW_U using the method described in section 3.2 or other method, he can compute any previous session key as follows.

1) A computes $g^{A_U} \bmod p = B_U \oplus PW_U$, $TID_U = TS_U \| ID_U$,

$n_U = TID_U \oplus NTID_U$ and $n_S = TID_U \oplus NTID_S$.

2) A computes $SK_S = h((g^{A_U} \bmod p) \| n_S \| n_U)$.

From the above description, we know Liao et al.'s scheme can't provide forward secrecy. In addition, once the master key and the table of TS_U of the server are obtained by A, he can compute TID_U and $g^{A_U} \bmod p$ directly, and does not need to carry out the password guessing attack. Then A can get the session more easily.

4. Conclusion

Smart card-based user authentication technology has been widely deployed in various kinds of applications, such as remote host login, withdrawals from automated cash dispensers, and physical entry to restricted areas. Recently, Liao et al. proposed a mutual authentication based on smart cards and passwords. However, after review of their scheme and analysis of its security, four kinds of attacks are proposed in different scenarios. We also point out the session key problem of Liao et al.'s protocol. The analyses show that Liao et al.'s scheme is insecure for practical application.

Reference

[1]. C. H. Liao, H. C. Chen, and C. T. Wang, An Exquisite Mutual Authentication Scheme with Key Agreement Using Smart Card, Informatica, Vol. 33, No. 2, May 2009, pp. 125-132.

[2]. W. Diffie, M. Hellman, (1976) New directions in cryptography, IEEE Trans. Inform. Theory, 22, pp.476– 492.

[3]. C.T. Wang, C.C. Chang, C.H. Lin, (2004) Using IC Cards to Remotely Login Passwords without Verification Tables, Proceedings of the 18th International Conference on Advanced Information Networking and Application(AINA), Fukoka, Japan, pp.321–326.

[4]. P. Kocher, J. Jaffe, B. Jun, Differential power analysis, Proc. Advances in Cryptology (CRYPTO'99), (1999) 388–397.

[5]. T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart card security under the threat of power analysis attacks, IEEE Transactions on Computers 51 (5) (2002) 541–552.