Improved Efficient Remote User Authentication Schemes

Xiaojian Tian, Robert W. Zhu, and Duncan S. Wong
(Corresponding author: Xiaojian Tian)

Department of Computer Science, City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong, China
(Received Oct. 7, 2005; revised and accepted Feb. 17, 2006)

Abstract

Recently, Yoon et al. proposed a new smart card based remote user authentication scheme. We show that this scheme is subject to forgery attacks if the information stored in the smart card is stolen. This violates the "two-factor security" objective of the smart card based remote user authentication schemes. We propose an amendment to this problem. We further propose two new schemes which are more efficient and secure than Yoon et al.'s scheme[1].

Keywords: Authentication, cryptography, security, and smart card

1 Introduction

A remote user authentication scheme allows a server to check the authenticity of a remote user through an insecure channel. In 1981, Lamport [9] proposed a password based remote user authentication scheme. As Hwang and Li [7] pointed out in 2000, this scheme suffers the risk of stolen password table and the high cost of maintaining and protecting the password table. Accordingly, Hwang and Li [7] proposed a smart card based remote user authentication scheme which eliminates the risk and cost in Lamport's scheme. However, their scheme was shown to have weaknesses and was improved in various ways [3, 4, 10, 12].

A typical smart card based remote user authentication scheme comprises three phases. In the registration phase, a user submits his identity and password to the server through a secure channel. The server uses the user's identity and password along with its long-term secret to generate some values and store them in a smart card which is then delivered to the user. In the login phase, a user attaches his smart card to a card reader and keys in his password. The smart card then uses the password and the values in the card to construct a login request and then sends it to the server. In the authentication phase, the server uses its long-term secret to check the validity of the login request. If mutual authentication is required, the server also uses its long-term secret to construct a message and sends it back to the user. The user then uses his password and the values in the smart card to check the validity of the message.

We consider the capabilities of an attacker that he may uses to thwart the security of the smart card based remote user authentication scheme. First, we assume that the attacker has total control over the communication channel between the users and the server in the login and authentication phase. That is, he may insert, delete, or modify any messages in the channel. Second, he may either steal a user's smart card and extract the values stored in the smart card, or steal a user's password. Obviously, if both the user's smart card and his password were stolen, then there is no way to prevent the attacker from masquerading as the user. So the best we can do is to guarantee the security of the scheme when either the user's smart card or his password is stolen, but not both. This security property is called *two-factor security*. We emphasize that, as Kocher et al. [8] and Messerges et al. [11] pointed out, all existing smart cards are vulnerable in that the secret keys stored in the smart card could be extracted by monitoring its power consumption. After an attacker obtains the secret values stored in a smart card, he may make another card that is digitally identical to the original card. If this happens, we must make sure that the attacker's best strategy is to launch an offline password guessing attack to guess the user's password. To thwart this attack, we must also require that the entropy of the user's password must be large enough so that it's impossible for the attacker to exhaust the user's password space within reasonable time and computation resource constraints.

Recently, Yoon et al. [13] proposed a new smart card based remote user authentication scheme which enhances Hwang and Li's scheme [7]. Yoon et al.'s scheme has several merits. It provides mutual authentication and session key generation. The user can choose and change his

[1]The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 9040904 (RGC Ref. No. CityU 1161/04E)).

password freely and securely without the help of the remote system. However, we find that their scheme does not provide two-factor security. Once an attacker gets the values in the smart card, he is able to forge any valid login request without knowing the user's password. After observing a user's valid login request, he is also able to forge the server's reply message. Thus, the objective of mutual authentication is totally broken. This is a serious problem in practice. We will give a modification to Yoon et al.'s scheme to eliminate this problem.

Yoon et al.'s scheme is based on generalized ElGamal signature scheme and uses expensive exponential operations which could be time-consuming for a small resource-constrained device such as a smart card. We propose two new smart card based remote user authentication schemes which only use cryptographic hash functions. They are more efficient and secure than Yoon et al.'s scheme while preserving all of its merits. One of the two schemes is based on timestamp, the other one uses a nonce based challenge-response mechanism.

This paper is organized as follows. We review Yoon et al.'s scheme in Section 2 and make an analysis and amendment in Section 3. In Section 4, we propose two new smart card based remote user authentication schemes and make security analysis. A comparison of our schemes and Yoon et al.'s scheme is made in Section 5.

2 Review of Yoon et al.'s Scheme

This section reviews a smart card based remote user authentication scheme proposed by Yoon et al. [13]. In their scheme, there are a server S and a set of users U_i, $1 \leq i \leq n$. Their scheme is divided into four phases: registration phase, login phase, authentication phase and password change phase.

Registration phase
This phase is invoked when a user U_i registers to S . It comprises the following steps:

1) U_i submits his identity ID_i and password PW_i to S through a secure channel.

2) S computes $VPW_i = g^{x_s} \mod p$, where x_s is S's longterm secret, p is a large prime number of bit size 1024-2048, q is a prime divisor of $p-1$ of bit size 160, and g is an element of order q in the finite field $GF(p)$.

3) S computes $R_i = h(ID_i, x_s)$ and $X_i = R_i \oplus h(ID_i, PW_i)$, where $\oplus$ denotes the bitwise exclusive OR operation, $h(\cdot)$ denotes a one-way hash function. The bit size of the output of $h(\cdot)$ is $|q|$, which denotes bit size of q.

4) S personalizes the smart card with the following information: $\{ID_i, VPW_i, R_i, X_i, h(\cdot), p, q, g\}$ and sends the smart card to the user in a secure way.

Login phase
This phase is invoked when U_i logins to S . U_i attaches his smart card to the card reader and keys in his password PW_i^* . The smart card then performs the following operations:

1) Generate a random number $r \in_R Z_q^*$.

2) Compute $k = (VPW_i)^r \mod p$.

3) Compute $t = h(k, T)$, where T is the current date and time of the input device.

4) Compute $V_i = X_i \oplus h(ID_i, PW_i^*)$.

5) Compute $s = r - V_i t \mod q$.

6) Send to S the login request $C_1 = \{ID_i, t, s, T\}$.

Authentication phase
Upon receiving the login request $C_1 = \{ID_i, t, s, T\}$, the server S and the user's smart card perform the following steps for mutual authentication between the user U_i and the server S.

1) The server checks the format of ID_i . If the format is incorrect, the login request is rejected.

2) The server verifies the freshness of T . If $T' - T \geq \triangle T$, where T' is the server's current time and $\triangle T$ is the expected valid time interval for a transmission delay, the server rejects the login request.

3) The server computes $V_i' = h(ID_i, x_s)$.

4) The server computes $k' = g^{(s+V_i' t)x_s} \mod p$.

5) The server compares t and $h(k', T)$. If they are equal, the server accepts the login request and proceeds to the next step, otherwise it rejects the login request.

6) The server acquires the current time T'' and computes $C_2 = h(k', V_i', T'')$. The server sends back the message $\{C_2, T''\}$

7) Upon receiving the message $\{C_2, T''\}$, the user U_i's smart card verifies the validity of the time interval between T'' and its current time, then computes $C_2' = h(k, V_i, T'')$ and compares C_2 and C_2'. If they are equal, then the user accepts the authenticity of the server, otherwise the user interrupts the connection.

8) After mutual authentication is completed, the user and the server use $k = k' = g^{x_s r} \mod p$ as the session key.

Password change phase
This phase is invoked when a user U_i wants to change his password from PW_i to PW_i'. In this phase, the user attaches his smart card to the card reader and keys in his password PW_i^* , then the smart card performs the following operations:

1) Compute $R_i' = X_i \oplus h(ID_i, PW_i^*)$.

2) Compare R_i' with R_i. If they are equal, then the smart card concludes that $PW_i^* = PW_i, R_i = R_i'$ and lets the user select a new password PW_i', otherwise it rejects the password change request.

3) Compute $X_i' = R_i \oplus h(ID_i, PW_i')$.

4) Store X_i' in smart card in place of X_i.

3 Forgery Attacks on Yoon et al.'s Scheme and an Amendment

In this section, we show that in Yoon et al.'s scheme, if an attacker steals a user's smart card and extracts the values stored in it through some means [8, 11] without being noticed, then the attacker can either masquerade as the user to forge a valid login request, or masquerade as the server to forge a valid reply message. Notice that the attacker does not need to know the user's password in any of our attacks. This also shows that their scheme does not achieve two-factor security. We then propose an amendment to Yoon et al.'s scheme to solve this problem.

Masquerade as a user
We note that, in step 4 of the login phase of Yoon et al.'s scheme, V_i should be equal to R_i in the smart card if $PW_i^* = PW_i$. This means that an attacker needs not to know PW_i to calculate V_i if he had known R_i from the smart card. Now the attacker can easily go through the steps in the login phase to forge a valid login request.

Masquerade as the server
Suppose an attacker intercepts a valid login request $C_1 = \{ID_i, t, s, T\}$ from a user U_i. Since $V_i' = V_i = R_i$, from step 5 of the login phase, the attacker can compute $r = s + V_i t \mod q = s + R_i t \mod q$ and $k' = k = (VPW_i)^r \mod p$. The attacker then gets the current time T'' and computes $C_2 = h(k', V_i', T'')$. The message $\{C_2, T''\}$ is obviously a valid reply message. The objective of the mutual authentication is now defeated and the session key k is exposed to the attacker.

An amendment
We note that in Yoon et al.'s scheme, R_i is stored in the smart card in order to check the validity of the user's password in the password change phase. However, to serve for that purpose, it is unnecessary to store R_i directly. We propose to store $h(R_i)$ instead. The step 2 of the password change phase should accordingly be modified to "Compare $h(R_i')$ with the stored value of $h(R_i)$ in smart card". Due to the one-way property of $h(\cdot)$, an attacker cannot reverse $h(R_i)$ to get R_i . Our fix forces the attacker who has extracted the values stored in the smart card to guess the password in order to obtain the value of R_i, which requires the attacker to launch offline dictionary attack against the password. That is, besides the values in the smart card, the attacker also needs to know the user's password for launching any of the attacks. Therefore, two-factor security is ensured.

4 Two New Remote User Authentication Schemes

Yoon et al.'s scheme is based on generalized ElGamal signature scheme and uses expensive exponential operations which could be time-consuming for a small resource-constrained device such as a smart card. In the following we propose two new smart card based remote user authentication schemes which use only cryptographic hash functions. They are more efficient and secure than Yoon et al.'s scheme while preserving all of its merits.

4.1 The First Scheme

The first scheme uses the timestamp mechanism, so it needs the users and the server to share a standard time, such as the Greenwich Mean Time. The scheme also has four phases: registration phase, login phase, authentication phase and password change phase.

Registration phase

1) U_i submits his identity ID_i and password PW_i to S through a secure channel. We require that the entropy of U_i 's password must be large enough to thwart the offline password guessing attack.

2) The server chooses four distinct cryptographic one-way hash functions $h(\cdot), h_1(\cdot), h_2(\cdot)$, and $h_3(\cdot)$.

3) The server computes $R_i = h(ID_i, x_s), H_i = h(R_i)$, and $X_i = R_i \oplus h(ID_i, PW_i)$, where $\oplus$ denotes the bitwise exclusive OR operation.

4) The server personalizes the smart card with $\{ID_i, H_i, X_i, h(\cdot), h_1(\cdot), h_2(\cdot), h_3(\cdot)\}$ and sends it to the user in a secure way.

Login phase
In this phase, U_i attaches his smart card to the card reader and keys in his password PW_i^*. Then the smart card performs the following operations:

1) Compute $R_i' = X_i \oplus h(ID_i, PW_i^*)$ and $H_i' = h(R_i')$.

2) Compare H_i' with H_i. If they are equal, then the smart card concludes that $PW_i^* = PW_i, R_i = R_i'$ and proceeds to the next step, otherwise it denies access from the user.

3) Acquire the current time T and compute $C_1 = h_1(S, ID_i, R_i, T)$.

4) Send to S the login request $\{ID_i, T, C_1\}$.

Authentication phase
Upon receiving the login request $\{ID_i, T, C_1\}$, the server S and the user U_i perform the following steps for mutual authentication:

1) S checks the validity of ID_i .

2) S checks the freshness of T.

3) S computes $R_i = h(ID_i, x_s)$ and checks whether $C_1 = h_1(S, ID_i, R_i, T)$. If the check passes, S deems U_i authentic and proceeds to the next step, otherwise it rejects the request.

4) S acquires the current time T' and computes $C_2 = h_2(ID_i, S, R_i, T')$. S sends back to user $\{T', C_2\}$. S and U_i use different hash functions in order to prevent the parallel session attack [5].

5) Upon receiving the server's reply message $\{T', C_2\}$, the user first checks the freshness of T' , then checks whether $C_2 = h_2(ID_i, S, R_i, T')$. If the check passes, the user accepts the authenticity of the server, otherwise it interrupts the connection.

6) After mutual authentication is completed, the user and the server use $h_3(ID_i, S, R_i, T, T')$ as the session key.

Password change phase
This phase is invoked when a user U_i wants to change his password from PW_i to PW_i'. In this phase, the user attaches his smart card to the card reader and keys in his password PW_i^* , then the smart card performs the following operations:

1) Compute $R_i' = X_i \oplus h(ID_i, PW_i^*)$ and $H_i' = h(R_i')$.

2) Compare H_i' with H_i . If they are equal, then the smart card concludes that $PW_i^* = PW_i, R_i = R_i'$ and lets the user select a new password PW_i' , otherwise it rejects the password change request.

3) Compute $X_i' = R_i \oplus h(ID_i, PW_i')$

4) Store X_i' in smart card in place of X_i.

4.2 The Second Scheme

The second scheme uses a nonce based challenge-response mechanism, so it avoids the time synchronization problem in the first scheme. This scheme also has four phases: registration phase, login phase, authentication phase and password change phase.

The registration phase and password change phase of the second scheme are the same as that of the first scheme and are omitted. We only elaborate the login phase and authentication phase below.

Login phase
In this phase, U_i attaches his smart card to the card reader and keys in his password PW_i^* . Then the smart card performs the following operations:

1) Compute $R_i' = X_i \oplus h(ID_i, PW_i^*)$ and $H_i' = h(R_i')$.

2) Compare H_i' with H_i. If they are equal, then the smart card concludes that $PW_i^* = PW_i, R_i = R_i'$ and proceeds to the next step, otherwise it denies access from the user.

3) Send to S the login request $\{ID_i, N_i\}$, where N_i is the nonce selected by U_i .

Authentication phase
Upon receiving the login request $\{ID_i, N_i\}$, the server S and the user U_i perform the following steps for mutual authentication:

1) S checks the validity of ID_i .

2) S chooses a nonce N_s , computes $R_i = h(ID_i, x_s)$,$C_1 = h_1(S, ID_i, R_i, N_i, N_s)$ and sends to $U_i : \{C_1, N_s\}$.

3) Upon receiving the message $\{C_1, N_s\}$, U_i checks whether $C_1 = h_1(S, ID_i, R_i, N_i, N_s)$. He deems S authentic if the check passes, otherwise he interrupts the connection.

4) U_i computes $C_2 = h_2(ID_i, S, R_i, N_s, N_i)$ and sends it to S .

5) Upon receiving C_2 , S checks whether $C_2 = h_2(ID_i, S, R_i, N_s, N_i)$. It deems U_i authentic if the check passes, otherwise it interrupts the connection.

6) After mutual authentication is completed, the user and the server use $h_3(ID_i, S, R_i, N_i, N_s)$ as the session key.

4.3 Security Analysis

In the following, we assume that all hash functions used in our schemes behave like random oracles [1].

In our two schemes, we note that an attacker must have the value R_i to masquerade as a user U_i to forge a valid login request to the server. Since the attacker has total control over the security channel in the login and authentication phase, he may try to deduce R_i from the communications between the user and the server that he observes. But since he can only observes $h_1(\cdots, R_i, \cdots)$ and $h_2(\cdots, R_i, \cdots)$ where "$\cdots$" denotes some other parameters, due to the randomness of $h_1(\cdot)$ and $h_2(\cdot)$, he cannot get R_i in this way. Alternatively, he may try to steal the user's smart card or his password. Obviously, if he steals both of them, then he must succeed in masquerading as the user. So we only consider the situation that he only obtains either the user's smart card or his password, but not both. First, if he obtains the user's password but doesn't get his smart card, then he cannot get R_i because R_i can only be deduced from $R_i = X_i \oplus h(ID_i, PW_i)$ which requires both the user's password PW_i and the secret value X_i stored in the user's smart card. On the other hand, if the attacker obtains the user's smart card and extracts the secret values

Table 1: Comparison of Yoon et al.'s Scheme and Our Two Schemes

	Yoon et al's Scheme	Our First (Second) Scheme
Computation of registration phase	1 exponential +2 hashing	3 hashing
Computation of login phase	1 exponential +2 hashing	3(2) hashing
Computation of authentication phase	2 exponential +4 hashing	6 (7) hashing
Computation of password change	2 hashing	3 hashing
Two-factor security	No	Yes

$\{ID_i, H_i, X_i, h(\cdot), h_1(\cdot), h_2(\cdot), h_3(\cdot)\}$ stored in the smart card, he still cannot get R_i directly since the smart card only stores the hash value of R_i , but not R_i . The attacker's best strategy is then to launch an offline password guessing attack, i.e., he may repeatedly chooses a password candidate PW_i' , calculates $R_i' = X_i \oplus h(ID_i, PW_i')$, and compares $h(R_i')$ with H_i , until he finds a PW_i' such that $h(R_i')$ equals H_i . That PW_i' is then equals to the user's password. With the user's password and the smart card in hand, the attacker can successfully masquerade as the user now. But as we stated before, the entropy of the user's password PW_i is large enough so it's impossible for an attacker to exhaust the user's password space within reasonable time and computation resource constraints. That is, the attacker cannot get the user's password in this way. In conclusion, our two schemes indeed provide *two-factor security*.

Our two schemes can also withstand replay attacks due to the freshness of the timestamp or the nonce.

An eavesdropper does not know the generated session key because he cannot compute $h_3(\cdots, Ri, \cdots)$ without knowing Ri . The freshness of the generated session key is also ensured due to the freshness of the timestamp or the nonce. We use a distinct hash function in the session key generation procedure in order to enhance the confidentiality of the generated session key.

In a parallel session attack, an attacker masquerade as a user through replaying the server's reply message. This is impossible in our schemes because the user's login request and the server's reply message use different hash functions.

In our second scheme, the inclusion of ID_i , S , N_i , and N_s in $h_1(\cdot)$ and $h_2(\cdot)$ is for defending against interleaving attacks [2].

In the two schemes, because the smart card verify H_i' with H_i in step 2 of the login phase and step 2 of the password change phase, if the smart card is stolen, unauthorized users cannot use the smart card or change the password of it.

By following the same set of security goals as the paper of Yoon et al. [13], we do not consider forward secrecy [6] in our paper.

5 Performance Comparison

We compare the performance of Yoon et al.'s scheme and our two schemes in Table I. We can see that our schemes only use hash functions which cost much less computational resources than exponential operations, so our schemes are more suitable to be used in a smart card based scenario. In practice, the smart card only needs to store the description of one cryptographic hash function, $h(\cdot)$. The other three functions will then be derived from the hash function, e.g. $h_1(\cdot) = h('11' \parallel h(\cdot))$, $h_2(\cdot) = h('22' \parallel h(\cdot))$, and $h_3(\cdot) = h('33' \parallel h(\cdot))$. So our schemes don't need too much storage space in the smart card.

Furthermore, our schemes can provide two-factor security while Yoon et al.'s scheme cannot. So our schemes are more efficient and secure than Yoon et al.'s scheme.

References

[1] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," *First ACM Conference on Computer and Communications Security*, pp. 62-73, 1993.

[2] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M. Yung, "Systematic design of two-party authentication protocols," in *Proceedings of Advances in Cryptology (CRYPTO'91)*, pp. 44-61, 1992.

[3] C. K. Chan and L. M. Cheng, "Cryptanalysis of a remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 992-993, Nov. 2000.

[4] C. C. Chang and K. F. Hwang, "Some forgery attacks on a remote user authentication scheme using smart cards," *Informatics*, vol. 14, no. 3, pp. 289-294, 2003.

[5] L. Gong, "A security risk of depending on synchronized clocks," *Operating Systems Review*, vol. 26, no. 1, pp. 49- 53, 1992.

[6] C. G. Günther , "An identity-based key-exchange protocol", *Advances in Cryptology (EUROCRYPT'89)*, pp. 29-37, 1990.

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

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

[9] L. Lamport, "Password authentication with insecure communication," *Communication of ACM*, vol. 24, pp. 770-772, 1981.

[10] K. C. Leung, L. M. Cheng, Anthony S. Fong, and C. K. Chan, "Cryptanalysis of a modified remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1243-1245, Nov. 2003.

[11] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541-552, May 2002.

[12] J. J. Shen, C. W. Lin, and M. S. Hwang, "A modified remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 414-416, May 2003.

[13] E. J. Yoon, E. K. Ryu, and K. Y. Yoo, "Efficient remote user authentication scheme based on generalized ElGamal signature scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 568-570, May 2004.

Xiaojian Tian received the BEng degree from the Department of Computer Science at Tianjin University, China in 1997, the MPhil degree from the same department in 2000 and the PhD degree from the Department of Computer Science at the Hong Kong University of Science and Technology in 2004. He is now a senior research associate in the Department of Computer Science at the City University of Hong Kong. His research interests include cryptography and computer security. Contact him at: xjtian@cityu.edu.hk.

Robert W. Zhu received his BEng degree from the Pilot Class in the School of Electronics and Electric Engineering at Shanghai Jiao Tong University, Shanghai, P. R. China in 2004. He is currently working for the MPhil degree in the Department of Computer Science at the City University of Hong Kong. His research interests include applied cryptography and computer security. Contact him at: zhuwei@cs.cityu.edu.hk.

Duncan S. Wong received the BEng degree from the University of Hong Kong in 1994, the MPhil degree from the Chinese University of Hong Kong in 1998, and the PhD degree from Northeastern University, Boston, MA, U.S.A. in 2002. He is an assistant professor in the Department of Computer Science at the City University of Hong Kong. Contact him at: duncan@cityu.edu.hk.