PAPER

Improvements on Hsiang and Shih's Remote User Authentication Scheme using Smart Cards

Jung-Yoon Kim†, *Student Member* and Hyoung-Kee Choi††, *Member*

SUMMARY Hsiang and Shih discovered that Yoon *et al*.'s user authentication scheme was vulnerable to parallel session attack, impersonation attack, and offline password guessing attack. They proposed an improved scheme to prevent these attacks. Hsiang and Shih's scheme is still susceptible to offline password guessing attack and server impersonation attack. In this paper, we demonstrate how their scheme can be compromised and then propose an improved scheme based on the Rabin cryptosystem to overcome the weaknesses. Furthermore, we discuss the reason why we should use an asymmetric encryption algorithm to secure a password-based remote user authentication scheme using smart cards. We formally prove the security of our proposed scheme using the BAN logic.

key words: *Network-level security and protection, authentication, security, password*

1. Introduction

Password-based user authentication schemes are developed to achieve efficient user authentication. The schemes provide user convenience, because a user only has to remember a password to login to a server. Conversely, those who do not know the password are unable to login the server. In 1981, Lamport [1] proposed a password-based authentication scheme for a server to authenticate remote users over an insecure channel. In Lamport's scheme, the server maintains a verification table consisting of hashed users' passwords to authenticate the users. If an attacker can modify the verification table, the attacker can impersonate a legitimate user. Furthermore, if an attacker steals the verification table, the attacker can derive users' passwords from the table. In 2000, Hwang and Li [2] proposed a password-based user authentication scheme using smart cards to avert the attacks on Lamport's scheme. In Hwang and Li's scheme, a server is able to authenticate users without the verification table. Sun [3] improved Hwang and Li's scheme significantly, reducing communication and computation costs. Sun's scheme provides the advantages of Hwang and Li's scheme. However, neither Hwang and Li's scheme nor Sun's scheme provide mutual authentication.

Table 1 A quick description of the attacks that are referenced in the paper

Attack	Description
Reflection attack	An attacker intercepts a message sent by a legitimate user and replays it for impersonation
Insider attack	An insider intercepts a message sent by another user over a secure channel in the same system
Non-reparability	A system cannot be recovered within a reasonable time after being compromised
Parallel session attack	An attacker establishes a new session with the server by posing as another user
Offline password guessing attack	An attacker repeats a trial of a password candidate for all candidates until finding the right password
Impersonation attack	An attacker masquerades as a legitimate user or a server
Man-in-the-middle attack	An attacker controls or eavesdrops on communications between victims (a user and a server)

To alleviate the problem, in 2002, Chien *et al*. [4] proposed a new password-based authentication scheme using smart cards. They claimed that their scheme would provide mutual authentication. In 2004, however, Ku and Chen [5] found security flaws in Chien *et al*.'s scheme. It was subject to reflection attack [6], insider attack [7], and non-reparability [8]. Hsu [9] and Yoon *et al*. [10] in 2004 and Duan *et al*. [11] in 2006 stated that Ku and Chen's scheme was vulnerable to parallel session attack [9, 11]. Yoon *et al*. found that Ku and Chen's scheme was insecure in changing the user's password and proposed improvements to overcome the weaknesses. Table 1 shows a quick description of the possible attacks an authentication scheme commonly faces, including the above attacks.

In 2009, Hsiang and Shih [12] pointed out that Yoon *et al*.'s scheme was still susceptible to parallel session attack, impersonation attack, and offline password guessing attack. They proposed an improved scheme that enhanced the security of Yoon *et al*.'s scheme whilst inheriting the advantages of Yoon *et al*.'s scheme. They claimed their scheme to be secure against offline password guessing attack, even if an attacker steals a user's smart card and the attacker breaches secrets stored in the smart card.

Unlike their claim, however, we discover that Hsiang and Shih's scheme is unable to thwart offline password

Manuscript received September 14, 2011.
Manuscript revised September 14, 2011.

† The author is with the Department of Mobile Systems Engineering, Sungkyunkwan University, Suwon, 440-746 Korea.
†† The author is with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, 440-746 Korea. (Corresponding author)

Table 2 Notations used throughout this paper.

U	user	ID	identity of U
S	remote server	PW	password of U
→	sending in common channel	n	number of times that U re-registers at S
=>	sending in secure channel	b	random number selected by U
z	S's public key for Rabin cryptosystem	x	permanent secret key of S
p, q	S's private keys for Rabin cryptosystem	$h(\cdot)$	one-way hash function

guessing attack and server impersonation attack. This research is motivated by our desires to report the security flaws of Hsiang and Shih's scheme, to discuss why these flaws are serious, to make it more secure with minimal modification, and to discuss a method for the improvement. In this paper, we describe how offline password guessing attack and server impersonation attack can be executed, and then propose an improved scheme to thwart these security flaws. Our proposed scheme is based on the Rabin cryptosystem [22].

The remainder of this paper is organized as follows. In Section 2, we review Hsiang and Shih's scheme. Section 3 shows security flaws in Hsiang and Shih's scheme. Section 4 proposes an improved scheme. Security analysis of the improved scheme is presented in Section 5. We discuss the reason why we should use an asymmetric encryption algorithm to secure a password-based authentication scheme using a smart card and compare the performance of Hsiang and Shih's scheme and our protocol in Section 6. We conclude in Section 7.

2. Review of Hsiang and Shih's Scheme

The notation describing protocols used throughout this paper is listed in Table 2. A secure channel is a way that enables a sender to transfer messages to a receiver without security threats such as tampering and eavesdropping. If two entities have a shared key, they are able to establish a secure channel using the key. An offline communication is generally regarded as another secure channel. A common channel is an insecure communication path that allows an attacker to eavesdrop and forge a message. The Internet is a representative common channel. Hsiang and Shih's scheme assumes that a secure channel can be established only in the registration phase.

Hsiang and Shih's scheme has four phases: registration, login, verification, and password change.

2.1 Registration Phase

This phase is invoked whenever U initially registers or re-registers to S. Let us denote n as the number of registrations. The following steps are involved in this phase.

1) U selects a random number b and computes $h(b \oplus PW)$.
2) $U \Rightarrow S$: ID, $h(PW)$, and $h(b \oplus PW)$.
3) If it is U's initial registration, S creates an entry for U in the account database and stores $n = 0$ in this entry. Otherwise, S sets $n = n + 1$ in the existing entry for U.
4) S performs the following computations:
   $P = h(EID \oplus x)$, where $EID = (ID \| n)$ and x is S's permanent secret key generated using a pseudo random number generator.
   $R = P \oplus h(b \oplus PW)$.
   $V = h(P \oplus h(PW))$.
5) $S \Rightarrow U$: a smart card containing V, R, and $h(\bullet)$.

U enters b into his smart card. Note that U's smart card contains V, R, b, and $h(\bullet)$, and U does not need to remember b after finishing the phase.

2.2 Login Phase

When U wants to login S, the following operations will perform:

1) U inserts his smart card in the smart card reader, and then enters ID and PW.
2) U's smart card computes the following:
   $C_1 = R \oplus h(b \oplus PW)$.
   $C_2 = h(C_1 \oplus T_U)$, where T_U is denoted as U's current timestamp.
3) $U \rightarrow S$: $C = \{ID, T_U, C_2\}$.

2.3 Verification Phase

After the message C is received, S and the smart card execute the following operations:

1) If either ID or T_U is invalid or $T_S - T_U \leq 0$, where T_S is denoted as S's current timestamp, S rejects U's login request. Otherwise, S computes $h(h(EID \oplus x) \oplus T_U)$. If the computed result equals the received C_2, S accepts U's login request and computes $C_3 = h(h(EID \oplus x) \oplus h(T_S))$. Otherwise, S rejects U's login request.
2) $S \rightarrow U$: T_S and C_3.
3) If either T_S is invalid or $T_S = T_U$, U terminates this session. Otherwise, U computes $h(C_1 \oplus h(T_S))$ and then compares the result with the received C_3. If they are equal, U successfully authenticates S.

2.4 Password Change Phase

This phase is invoked whenever U wants to change his password PW with a new one, PW_{new}.

1) U inserts his smart card in the smart card reader, enters ID and PW, and requests password change.
2) U's smart card computes $P' = R \oplus h(b \oplus PW)$ and $V' = h(P' \oplus h(PW))$.

3) U's smart card compares V' and the stored V in the card. The request is rejected if V' and V are not the same.
4) U selects a new password PW_{new}, the smart card computes $R_{new} = P' \oplus h(b \oplus PW_{new})$ yielding $h(EID \oplus x) \oplus h(b \oplus PW_{new})$, and then replaces R with R_{new}.
5) U's smart card computes $V_{new} = h(P' \oplus h(PW_{new}))$ yielding $h(h(EID \oplus x) \oplus h(PW_{new}))$, and then replaces V with V_{new}.

3. Security Flaws in Hsiang and Shih's Scheme

In this section, we describe security flaws in Hsiang and Shih's scheme, depicting how offline password guessing attack and server impersonation attack can be executed.

3.1 Offline Password Guessing Attack

Offline password guessing attack [13, 14, 15, 16] means that an attacker tries to find a user's password in an offline manner. The attacker can freely guess a password and then verify if it is correct without limitation in the number of guesses. In general, the attacker can easily obtain a user's password via offline password guessing attack within a reasonable time boundary, because users tend to choose simple and weak passwords for their convenience [15, 16, 17, 18, 19]. In this case, even if the password is converted into an unpredictable random number by using a one-way hash function, the attacker can easily find the correct password by comparing a hashed password candidate with the hashed correct password because the password candidate consists of a limited allowable character set. Furthermore, since the users tend to use the same password in several servers for convenience [13, 16, 19], the attacker can login the servers as a legitimate user after purloining the user's password. For these reasons, all password-based user authentication schemes should be able to prevent offline password guessing attack. Hsiang and Shih contended their approach could withstand offline password guessing attack even if an attacker successfully accessed the secret values stored in a smart card. Despite their claim, we found their approach remains vulnerable to offline password guessing attack. We present two scenarios of offline password guessing attack for their scheme. In these scenarios, the attacker can breach the secrets V, R, $h(\cdot)$, and b stored in U's smart card in various ways [20, 21] after the attacker has stolen the smart card, and can intercept packets between a user and the server.

In the first scenario, the attacker performs the following operations.

1) The attacker selects a password candidate PW'.
2) The attacker computes $P' = R \oplus h(b \oplus PW')$.
3) The attacker computes $V' = h(P' \oplus h(PW'))$.
4) The attacker repeats the above steps from 1) to 3) until the computed result V' equals the breached secret V.
5) If they are equal, $PW' = PW$.

The attacker is able to guess PW using three XORs, three hash functions, and one comparison for each password candidate in an offline manner.

In the second scenario, the attacker performs the following operations using the intercepted messages C_2 and T_U, and the breached secrets R, $h(\cdot)$, and b:

1) The attacker selects a password candidate PW'.
2) The attacker computes $C_2' = h(R \oplus h(b \oplus PW') \oplus T_U)$.
3) The attacker repeats the above steps until the computed result C_2' equals the intercepted message C_2.
4) If they are equal, $PW' = PW$.

The attacker is able to guess PW using three XORs, two hash functions, and one comparison for each password candidate in an offline manner.

3.2 Server Impersonation Attack

In general, an attacker can masquerade as a user if the attacker obtains the user's password, because password-based user authentication is based on the knowledge of the password. However, an attacker should not be able to impersonate the server even after obtaining a user's password. Hsiang and Shih's scheme allows an attacker to masquerade as the server if the attacker obtains a user's password.

After obtaining the user U's password through offline password guessing attack described in Section 3.1 and extracting secrets R and b stored in the user U's smart card, an attacker A can masquerade as the server S by performing the following operations.

1) When a user U sends the message $C = \{ID, T_U, C_2\}$ to the server S in the login phase of Hsiang and Shih's scheme, the attacker A intercepts the message C and computes the following:
   $C_3 = h(R \oplus h(b \oplus PW) \oplus h(T_A))$, where T_A is denoted as the attacker A's current timestamp.
2) $A \rightarrow U : T_A$ and C_3.

After receiving the message T_A and C_3, the user U executes the following operations:

1) If either T_A is invalid or $T_A = T_U$, U terminates this session. Otherwise, U computes $h(C_1 \oplus h(T_A))$ and then compares the result with the received C_3. If they are equal, U successfully authenticates A. Note that $C_1 = R \oplus h(b \oplus PW) = P$.

The received C_3 should be equal to $h(C_1 \oplus h(T_A))$. Hence, the attacker A successfully impersonates S.

It is ideal to impersonate the server S without the user U's password in the attacker A's viewpoint. In Hsiang and Shih's scheme, although this is impossible, the attacker A is able to obtain other benefits by impersonating S with U's password; A can violate the user U's privacy and provide forged services to the user U.

The privacy violation scenario is as follows.

1) A user connects to an attacker who masquerades as a server and requests a service to the attacker, because the victim believes that the attacker is the genuine server.

2) The attacker can find which service this victim requests by reading this request message. In addition, the attacker can perform man-in-the-middle attack by connecting to the genuine server using the message $C = \{ID, T_U, C_2\}$ sent from the victim in the login phase (that is, the attacker forwards the message C to the server S). Then, the attacker A intercepts the server S's response (that is, T_S and C_3) and impersonates the server S as described earlier in this section. As a result, the messages exchanged between the victim and the genuine server are disclosed to the attacker; the victim's privacy is broken.

The scenario for providing forged services is as follows.

1) A user connects to an attacker masquerading as a server and requests a service to the attacker.

2) The attacker provides a forged service, such as a service including forged information, to the victim. The victim may accept this forged service, because the victim believes that this service is provided by the genuine server.

4. Our Proposed Scheme

We propose an improved scheme to alleviate the security flaws in Hsiang and Shih's scheme. The proposed scheme inherits the advantages of Hsiang and Shih's scheme, including: (1) no verification table, (2) no restrictions in choosing their passwords, and (3) mutual authentication between U and S. At the same time, the proposed scheme can overcome the flaws and vulnerabilities discovered in Hsiang and Shih's scheme. Additionally, our proposed scheme enables a user and the server to establish a session key between them. Our scheme is based on the Rabin cryptosystem [22]. There are also four phases in our scheme – registration, login, verification, and password change. Our proposed scheme assumes that a secure channel can be established only in the registration phase. Each phase works as follows.

4.1 Registration Phase

The registration phase is invoked when a user U registers to a remote server S.

1) U selects a random number b and computes $h(b \oplus PW)$.

2) $U \Rightarrow S : ID$ and $h(b \oplus PW)$.

3) If it is U's initial registration, S creates an entry for U in the account database and stores ID and $n = 0$ in this entry. Otherwise, S sets $n = n + 1$ in the existing entry for U.

4) S computes the following equations:
$P = h(EID \oplus x)$, where $EID = (ID \| n)$.
$R = P \oplus h(b \oplus PW)$.

5) $S \Rightarrow U$: a smart card containing R, z, and $h(\cdot)$, where $z = p * q$ is the public key of S for the Rabin cryptosystem. p and q are the private keys of S corresponding to z.

6) U enters b in his smart card. U's smart card contains R, z, b, and $h(\cdot)$.

4.2 Login Phase

U performs the following operations to login S:

1) U inserts his smart card in the smart card reader, and then enters ID and PW.

2) U's smart card computes the following equations:
$C_1 = R \oplus h(b \oplus PW)$.
$C_2 = (ID \| C_1 \| M_U \| T_U)^2 \bmod z$, where M_U is a random number.

3) $U \rightarrow S : C_2$.

4.3 Verification Phase

S and U mutually authenticate each other in this phase. They perform the following operations after the message C_2 is received:

1) S decrypts C_2 with its private keys p and q using the Rabin cryptosystem [22]. If ID or T_U is invalid, S rejects this login request. Otherwise, S computes $h(EID \oplus x)$ to compare with the decrypted result C_1. Note that it can be claimed that ID and T_U are valid if they are in the ID space and the timestamp space, respectively. If the comparison is positive, S accepts U's login request and computes $k_{SU} = h(M_U \| M_S)$ and $C_3 = h(T_S \| k_{SU})$, where T_S and M_S are denoted as S's current timestamp and a random number, respectively. Otherwise, the authentication fails.

2) $S \rightarrow U : T_S$, M_S, and C_3.

3) If $T_S > T_U$, U computes $k_{SU} = h(M_U \| M_S)$ and $h(T_S \| k_{SU})$ to compare to C_3. If they are equal, U successfully authenticates S. Otherwise, the smart card terminates this session.

S and U use $k_{SU} = h(M_U \| M_S)$ as a session key between them for providing confidentiality and integrity. The session key k_{SU} is temporarily stored in the volatile memory of the smart card until removing the smart card from the smart card reader.

4.4 Password Change Phase

U can change his password with a new one, PW_{new}, in this phase.

1) U inserts his smart card in the smart card reader and enters ID, PW, and PW_{new}, to request a password change.

2) U's smart card and S invoke the login and verification phases sequentially.
3) U's smart card computes $D_1 = h(b \oplus PW_{new})$ and encrypts D_1 and a timestamp T_U with k_{SU}. E_1 is denoted as the encrypted result.
4) $U \rightarrow S : E_1$.
5) S obtains D_1 and T_U by decrypting E_1 with k_{SU} and checks if T_U is a valid timestamp. If it is invalid, S rejects the password change request. Otherwise, S computes $R_{new} = h(EID \oplus x) \oplus D_1$ and encrypts R_{new} and a timestamp T_S with k_{SU}. E_2 is denoted as the encrypted result.
6) $S \rightarrow U : E_2$.
7) U's smart card decrypts E_2 with k_{SU} and checks if T_S is valid. If so, this smart card replaces R with R_{new}. Otherwise, this smart card terminates the password change phase.

5. Security Analysis

We describe the purpose and correctness of the modifications that we introduce to Hwang and Shih's scheme in Section 5.1. We contend that our scheme would be secure against offline password guessing attack, user impersonation attack, and server impersonation attack in Sections 5.2, 5.3, and 5.4, respectively. We also provide the formal proof of the security of the proposed scheme in Section 5.5.

5.1 Correctness

We modify the registration phase of Hsiang and Shih's scheme by eliminating V, because V may cause offline password guessing attack as described in Section 3.1. z is used for encryption of authentication parameters and session key generation between S and U. This session key is used for mutual authentication and secure communication. The other values have the same purpose with those of Hsiang and Shih's scheme.

In the login phase, C_2 is modified to conceal authentication parameters and send the session key between U and S securely. The authentication parameters ID, C_1, and T_U and a random number M_U are encrypted using the Rabin cryptosystem. The reason why the Rabin cryptosystem should be used is discussed in Section 6. C_1, ID, and T_U are used for authentication of U. Note that the encryption algorithm of the Rabin cryptosystem is not injective; that is, there are four possible results that can be obtained by decrypting one ciphertext. In general, a meaningful result is chosen as the true result (the true plaintext). In the proposed scheme, the server S can determine the true result among four results decrypted from C_2 using ID and T_U, because the true result includes the valid ID and T_U. If there is a valid pair of ID and T_U among the four decrypted results, S can find the true C_1 and M_U from the decrypted C_2 which includes the valid ID and T_U.

In the verification phase, user authentication should be successful if PW is correct. An attacker cannot masquerade as a legitimate user, because an attacker is unable to forge the valid authentication value C_1, without knowing the correct password. C_3 is used for server authentication. An attacker cannot impersonate the server because of the unknown session key k_{SU}. Only the server that knows k_{SU} can generate C_3 using k_{SU}. Hence, our proposed scheme provides mutual authentication.

In Hsiang and Shih's scheme, V is used for changing a password. In the proposed scheme, the session key k_{SU} is used for securing the password change phase. E_1 is the encrypted value of $h(b \oplus PW_{new})$ and T_U. E_2 is the encrypted one of $R_{new} = h(EID \oplus x) \oplus h(b \oplus PW_{new})$ and T_S. T_U and T_S are used for authentication of E_1 and E_2, respectively. Because of the unknown secret k_{SU}, an attacker cannot obtain $h(b \oplus PW_{new})$ and R_{new} or alter E_1 and E_2 used for password change. As a result, a user can freely change a password because the password change phase is well protected.

5.2 Offline Password Guessing Attack

Hsiang and Shih's scheme is vulnerable to offline password guessing attack because all the values used for authentication, except a password, are revealed to an attacker; they are saved in the smart card or sent from the smart card to the server in plaintext. The attacker is able to find the correct password by verifying whether some information generated through a password candidate is valid or not. In Hsiang and Shih's scheme, the attacker is able to find the correct password by comparing the generated value $V' = h(R \oplus h(b \oplus PW') \oplus h(PW'))$ with $V = h(R \oplus h(b \oplus PW) \oplus h(PW))$ stored in the smart card, or by comparing $C_2' = h(R \oplus h(b \oplus PW') \oplus T_U)$ generated by the attacker with $C_2 = h(R \oplus h(b \oplus PW) \oplus T_U)$ sent from the smart card.

Our improved scheme is secure against offline password guessing attack, because C_1, which is a value used for authentication, and a session key k_{SU} are not revealed; it is neither saved in the smart card nor sent in plaintext from the smart card. An attacker is unable to verify whether some information generated through a password candidate is valid or not, even if the attacker obtains all of the values stored in the smart card and sent from the smart card. For the verification, the attacker should solve the integer factorization problem; it is hard for the attacker to solve the problem in polynomial time. Consequently, the attacker cannot find the correct password via offline password guessing attack in our proposed scheme.

5.3 User Impersonation Attack

The message C_1 in the login phase of the proposed scheme is encrypted with the server's public key. Hence, an attacker cannot falsify the valid C_1 without knowing the server's private keys. Note that since C_2 includes the current timestamp, an attacker cannot impersonate the user by replaying the valid C_2 after eavesdropping on it.

5.4 Server Impersonation Attack

In the proposed scheme, the legitimate user *U* authenticates the server *S* using the shared secret M_U. This secret is randomly generated by the legitimate user *U* and sent to the server *S* after encryption with *S*'s public key. M_U is known to only its generator *U* and the server *S*, so the attacker cannot impersonate the server. Without the knowledge of the server's private keys, the attacker cannot obtain the value used for server authentication, such as M_U.

5.5 Formal Proof

We formally prove the security of the proposed scheme based on the BAN logic [25]. We use the following notations by convention: *U* and *S* are two entities, k_{SU} is the fresh session key shared between *S* and *U*, and *z* is *S*'s public key; other notations follow those of the BAN logic [25]. We focus on the messages exchanged for mutual authentication and key agreement between a user *U* and the server *S* and verify whether they can ascertain that they share a fresh session key k_{SU} with each other.

The assumptions that we make before the verification are:

1) $U \mid\equiv \overset{z}{\mapsto} S$;
2) $S \mid\equiv \overset{z}{\mapsto} S$;
3) $U \mid\equiv \#(M_U)$;
4) $S \mid\equiv \#(M_S)$;
5) $U \mid\equiv U \overset{C_1}{\rightleftharpoons} S$;
6) $S \mid\equiv U \overset{C_1}{\rightleftharpoons} S$;
7) $U \mid\equiv S \mid\equiv U \overset{C_1}{\rightleftharpoons} S$;
8) $S \mid\equiv U \mid\equiv U \overset{C_1}{\rightleftharpoons} S$;
9) $U \mid\equiv S \mid\Rightarrow U \xleftrightarrow{k_{SU}} S$;
10) $S \mid\equiv U \xleftrightarrow{k_{SU}} S$;
11) $S \mid\equiv U \mid\sim (U \xleftrightarrow{k_{SU}} S)$.

Assumptions 1) and 2) state that *U* and *S* believe that *S* possesses a public key *z*. Assumptions 3) and 4) mean that *U* and *S* generate two fresh random numbers *MU* and *MS* and assure their freshness. Assumptions 5), 6), 7), and 8) mean that *U* and *S* have the shared secret C_1. Assumptions 9), 10), and 11) tell that *U* and *S* have the shared session key k_{SU}. Assumption 9) states *U* believes *S* has jurisdiction right over k_{SU}, because once *U* generates M_U and sends it to *S* with the shared secret C_1, k_{SU} is finally determined by the random number M_S generated by *S* from the viewpoint of *U*. Assumptions 10) and 11) hold because *S* computes the fresh session key k_{SU} with two fresh random numbers chosen by *U* and itself.

The verification is shown as follows:

Message 1 $U \rightarrow S : \{M_U, U \overset{C_1}{\rightleftharpoons} S\}_z$.

12) $S \triangleleft (M_U, U \overset{C_1}{\rightleftharpoons} S)$;
13) $S \mid\equiv \#(M_U, U \overset{C_1}{\rightleftharpoons} S)$;
14) $S \mid\equiv U \mid\sim (M_U, U \overset{C_1}{\rightleftharpoons} S)$;
15) $S \mid\equiv U \mid\equiv (M_U, U \overset{C_1}{\rightleftharpoons} S)$.

Message 2 $S \rightarrow U : \{U \xleftrightarrow{k_{SU}} S\}_{k_{SU}}$.

16) $U \triangleleft \{U \xleftrightarrow{k_{SU}} S\}_{k_{SU}}$;
17) $U \mid\equiv U \xleftrightarrow{k_{SU}} S$;
18) $U \mid\equiv \#(U \xleftrightarrow{k_{SU}} S)$;
19) $U \mid\equiv S \mid\sim (U \xleftrightarrow{k_{SU}} S)$;
20) $U \mid\equiv S \mid\equiv (U \xleftrightarrow{k_{SU}} S)$;
21) $S \mid\equiv U \xleftrightarrow{k_{SU}} S$;
22) $S \mid\equiv \#(U \xleftrightarrow{k_{SU}} S)$.

In the login phase (Message 1), a user calculates the shared secret C_1 using *R*, *b*, and *PW* and then securely sends C_1 and a fresh random number M_U to *S*. The Message 1 is fresh for each authentication attempt because of the random number M_U. Because of the shared secret C_1, *S* can authenticate *U*. In the verification phase (Message 2), *S* generates a fresh random number M_S and calculates the session key k_{SU} shared between *S* and *U* using M_U and M_S. Then, *S* proves that it can generate k_{SU} by sending C_3 which is generated using k_{SU}. Note that only *S* can generate $k_{SU} = h(M_U \| M_S)$, because only the entity that has the corresponding private keys of *S*'s public key *z* can find M_U from the Message 1.

6. Discussion

We discuss the reason why we should use an asymmetric encryption scheme to prevent an offline password guessing attack in this section.

Let an authentication parameter be a value used for authentication and calculated by using a password. For instance, C_2 and *V* are authentication parameters in Hsiang and Shih's scheme. If a user would choose a weak password and an attacker could obtain an authentication parameter and all the input values used for calculating the authentication parameter except a password, the attacker should be able to derive the password via offline password guessing attack. This is because the attacker calculates an authentication parameter using a password candidate and can then check if this candidate is the same as the correct password by comparing the calculated authentication parameter with the obtained one as described in Section 5.2. Even if a client and the server use symmetric encryption such as Advanced Encryption Standard (AES) [23] to conceal the authentication parameter and input values, the attacker can decrypt them after breaching the symmetric key stored in the smart card. Hence, schemes that use only symmetric operations such as hash, XOR, and symmetric encryption should be insecure against offline password guessing attack because an authentication parameter and its input values cannot be concealed by using only these operations.

Asymmetric encryption is a viable option to conceal the authentication parameter and input values. If a client encrypts them with the server's public key, an attacker cannot obtain the authentication parameter and input values because the attacker is unable to decrypt them even if the attacker breaches all the secret values stored in the smart card. The corresponding private key for decryption is not stored in the smart card but stored only in the server. It is almost impossible for an attacker to find the private key because most of the asymmetric encryption schemes are based on the intractability of the integer factorization problem or the discrete logarithm problem.

We adopt the asymmetric encryption to improve Hsiang and Shih's scheme. In our scheme, an authentication parameter C_1 is encrypted using the server's public key z. To find C_1, the attacker should first obtain the server's private keys p and q and decrypt C_2 using these private keys. The attacker cannot obtain the private keys, because they are stored in the server securely. After guessing C_1 using R, b, and a password candidate PW' and generating C_2' using the guessed C_1, the attacker can try to compare C_2 with C_2' to find the correct C_1. If they are same, the password candidate PW' is equal to the correct PW. However, to generate C_2', the attacker should first obtain the correct M_U. The attacker cannot find or guess M_U, because this value is a large random number generated using a pseudo random number generator and is not stored in the smart card and not sent in plaintext. To find M_U, the attacker should acquire the server's private keys by breaking the Rabin cryptosystem. As a result, the security of our proposed scheme is based on the integer factorization problem.

Hsiang and Shih's scheme needs nine hashes and eight XORs in the login and verification phases whereas our scheme requires five hashes, three XORs, one modular multiplication, and one decryption using the Rabin cryptosystem. The Rabin cryptosystem can be adopted into our proposed scheme because a user should perform only one modular multiplication and some inexpensive operations, such as a hash function, per mutual authentication. In general, a smart card performs up to 3000 modular multiplications per second [24]. Although it is necessary for the server to perform expensive operations in our scheme, it may cause an insignificant decrease in overall system performance because it is assumed that a server is able to maintain sufficient performance. Hence, our proposed scheme is practical in terms of both security and performance.

7. Conclusion

Yoon *et al.* proposed a password-based authentication scheme to provide efficient and secure user authentication. However, this scheme has some security flaws such as parallel session attack, impersonation attack, and offline password guessing attack. Hsiang and Shih proposed an improved scheme to eliminate these security flaws of Yoon *et al.*'s scheme. We found Hsiang and Shih's scheme is still vulnerable to offline password guessing attack and server impersonation attack. We demonstrated the attack scenarios and then proposed an improved scheme. Our scheme is secure against the offline password guessing attack, user impersonation attack, and server impersonation attack even if an attacker can breach the secret values stored in the smart card with minimal additional overheads.

References

[1] L. Lamport, "Password authentication with insecure communication," Communications of the ACM, vol. 24, no. 11, 1981, pp. 770-772.

[2] 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, 2000, pp. 28-30.

[3] H.M. Sun, "An efficient remote user authentication scheme using smart cards," IEEE Transactions on Con-sumer Electronics, vol. 46, no. 4, 2000, pp. 958-961.

[4] H.Y. Chien, J.K. Jan, and Y.M. Tseng, "An efficient and practical solution to remote authentication smart card," Computers & Security, vol. 21, no. 4, 2002, pp. 372-375.

[5] W.C. Ku and S.M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 50, no. 1, 2004, pp. 204-207.

[6] C. Mitchell, "Limitations of challenge-response entity authentication," Electronic Letters, vol. 25, no. 17, 1989, pp. 1195-1196.

[7] W.C. Ku, C.M. Chen, and H.L. Lee, "Cryptanalysis of a variant of Peyravian–Zunic's password authentica-tion scheme," IEICE Transactions on Communication, vol. E86-B, no. 5, 2003, pp. 1682-1684.

[8] T. Hwang and W.C. Ku, "Reparable key distribution protocols for internet environments," IEEE Transactions on Communications, vol. 43, no. 5, 1995, pp. 1947-1950.

[9] C.L. Hsu, "Security of Chien et al.'s remote user authentication scheme using smart cards," Computer Stan-dards and Interfaces, vol. 26, no. 3, 2004, pp. 167-169.

[10] E.J. Yoon, E.K. Ryu, and K.Y. Yoo, "Further improvement of an efficient password based remote user au-thentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 50, no. 2, 2004, pp. 612-614.

[11] X. Duan, J.W. Liu, and Q. Zhang, "Security improvement on Chien et al.'s remote user authentication scheme using smart cards," in: Proceedings of IEEE International Conference on Computational Intelligence and Security (CIS'06), 2006, pp. 1133-1135.

[12] H.C. Hsiang and W.K. Shih, "Weaknesses and improvements of the Yoon-Ryu-Yoo remote user authentica-tion scheme using smart cards," Computer Communications, vol. 32, no. 4, 2009, pp. 649-652.

[13] W.C. Ku, "Weaknesses and drawbacks of a password authentication scheme using neural networks for mul-tiserver architecture," IEEE Transactions on Neural Networks, vol. 16, no. 4, 2005, pp. 1002-1005.

[14] C.C. Yang and R.C. Wang, "Cryptanalysis of improvement of password authenticated key exchange based on RSA for

imbalanced wireless networks," IEICE Transactions on Communication, vol. E88-B, no. 11, 2005, pp. 4370-4372.

[15] T. Cao and D. Lin, "Cryptanalysis of two password authenticated key exchange protocols based on RSA," IEEE Communications Letters, vol. 10, no. 8, 2006, pp. 623-625.

[16] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in: Proceedings of 9th ACM Con-ference on Computer and Communications Security (CCS'02), 2002, pp. 161-170.

[17] B.T. Hsieh, H.M. Sun, and T. Hwang, "On the security of some password authentication protocols," Informatica, vol. 14, no. 2, 2003, pp. 195-204.

[18] C.L. Lin and T. Hwang, "A password authentication scheme with secure password updating," Computers & Security, vol. 22, no. 1, 2003, pp. 68-72.

[19] L. Gong, M.A. Lomas, R.M. Needham, and J.H. Saltzer, "Protecting poorly chosen secrets from guessing attacks," IEEE Journal on Selected Areas in Communications, vol. 11, no. 5, 1993, pp. 648-656.

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

[21] 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, 2002, pp. 541-552.

[22] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Technical Re-port: TR-212, 1979.

[23] C. Kaufman, R. Perlman, and M. Speciner, Network Security: Private Communication in a Public World (2nd ed.). Upper Saddle River: Prentice Hall PTR, 2002, ch. 2-3.

[24] C. Zouridaki, B. L. Mark, K. Gaj, and R. K. Thomas, "Distributed CA-based PKI for mobile ad hoc networks using elliptic curve cryptography," Lecture Notes in Computer Science, vol. 3093, 2004, pp. 232-245.

[25] M. Burrows, M. Abadi, R. Needham, A logic of authentication, in: Proceedings of Royal Soc. London A, 1989, vol. 426, pp. 233-271.

Jung-Yoon Kim is a Ph.D. student of Sungkyunkwan University in Korea. He received his B.S. degree (2006) and M.S. degree (2008) from Sungkyunkwan University in Korea. He held internship at AhnLab (2004) where he worked for quality assurance of network security systems. He has research interests in network security, especially authentication and key management.

Hyoung-Kee Choi received a Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology in 2001. He is an associate professor and a director of the Education Center for Mobile Communications in Sungkyunkwan University, Korea. He joined Lancope in 2001 and remained until 2004, where he guided and contributed to research in Internet security. His research interests span network security and Internet traffic modeling. He serves as an Associate Editor for ACM Transactions on Internet Technology.