IEICE TRANS. FUNDAMENTALS, VOL.E94–A, NO.6 JUNE 2011

PAPER

# Further Improved Remote User Authentication Scheme

**Jung-Yoon KIM**[†a)], ***Student Member***, **Hyoung-Kee CHOI**[††], ***Member***, ***and*** **John A. COPELAND**[†††], ***Nonmember***

**SUMMARY** Kim and Chung previously proposed a password-based user authentication scheme to improve Yoon and Yoo's scheme. However, Kim and Chung's scheme is still vulnerable to an offline password guessing attack, an unlimited online password guessing attack, and server impersonation. We illustrate how their scheme can be compromised and then propose an improved scheme to overcome the weaknesses. Our improvement is based on the Rabin cryptosystem. We verify the correctness of our proposed scheme using the BAN logic.

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

## 1. Introduction

User authentication is an essential security requirement for protecting systems and networks. Servers only allow legitimate users to access the systems via user authentication. Hence, the user authentication affects the system's security strength and performance.

Password-based user authentication schemes have been developed to achieve higher efficiency for user authentication. These schemes provide user convenience because a user only has to remember a password in order to log into a server. Lamport [1] proposed a password-based user authentication scheme that allows a server to authenticate remote users over an insecure channel. In Lamport's scheme, the server maintains a verification table that consists of hashing values of users' passwords in order to authenticate users. If an attacker can alter the verification table, then the attacker has the ability to impersonate a legitimate user. To thwart this attack, Hwang and Li [2] proposed a password-based user authentication scheme using smart cards, but did not provide mutual authentication. Chien et al. [3] proposed a new authentication scheme using smart cards in order to solve this problem. Later, Hsu [4] showed that the scheme proposed by Chien et al. is still vulnerable to a parallel session attack. Lee et al. [5], [6] proposed an improved scheme that eliminated the security flaw of Chien et al.'s scheme. However, Yoon and Yoo [7] found that Lee et al.'s scheme is susceptible to some malicious attacks such as a masquerading server attack. They also enhanced the security of Lee et al.'s scheme.

Recently, Kim and Chung [8] discovered that Yoon and Yoo's scheme has security flaws under the following types of attacks: an offline password guessing attack [9]–[12], a masquerading server attack, a masquerading user attack, and a stolen verifier attack [13]. Then, Kim and Chung proposed an improved scheme to remove the weaknesses that were found. Their scheme also claimed to provide that security against an offline password guessing attack, even if an attacker steals a user's smart card and extracts secrets stored in the smart card.

However, when attempting to verify their claim, we found that Kim and Chung's scheme is unable to avert the offline password guessing attack. Their scheme also fails to defend an unlimited online password guessing attack and server impersonation. In this paper, we describe how the offline password guessing attack, the unlimited online password guessing attack, and the server impersonation can be executed, and then propose an improvement for the elimination of security flaws.

The rest of this paper is organized as follows. In Sect. 2, we review Kim and Chung's scheme. Section 3 shows cryptanalysis of Kim and Chung's scheme. We propose an improved scheme in Sect. 4. The security analysis of our improved scheme is presented in Sect. 5. In Sect. 6, we discuss several issues important for practical implementation of our scheme. We give a conclusion in Sect. 7.

## 2. Review of Kim and Chung's Scheme

The notations used throughout this paper are summarized as shown in Table 1.

**Table 1** Notations used throughout this paper.

| | | | | |
|---|---|---|---|---|
| $U$ | user | $ID$ | identity of $U$ | 128 bits |
| $S$ | remote server | $PW$ | password of $U$ | 128 bits |
| $\rightarrow$ | sending in common channel | $K_{SU}$ | session key between $U$ and $S$ | 128 bits |
| $=>$ | sending in secure channel | $T_1$, $T_2$ | timestamps | 32 bits |
| $\Delta T$ | timestamp threshold | $x$ | permanent secret key of $S$ | 128 bits |
| $\oplus$ | XOR operation | $p$, $q$ | private keys of $S$ | 512 bits |
| $m$ | $p * q$ | h(·) | one-way hash function | 256 bits |

Manuscript received September 15, 2010.
Manuscript revised December 22, 2010.

[†]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)
[†††]The author is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA.
a) E-mail: steal83@ece.skku.ac.kr

DOI: 10.1587/transfun.E94.A.1426

Kim and Chung's scheme consists of the following four phases: registration, login, verification, and password change.

### 2.1 Registration Phase

In this phase, the user $U$ initially registers with the server $S$ as follows.

1) $U$ => $S\{ID, PW\}$: $U$ chooses his $ID$ and $PW$, and sends them over a secure communication channel to $S$.
2) Upon receiving $ID$ and $PW$, $S$ derives $K_1 = \mathrm{h}(ID \oplus x) \oplus N$ and $K_2 = \mathrm{h}(ID \oplus x \oplus N) \oplus \mathrm{h}(PW \oplus \mathrm{h}(PW))$, where $N$ is a random number unique to the user $U$. Then $S$ computes a quantity $R = K_1 \oplus \mathrm{h}(PW)$.
3) $S$ stores the secure information $K_1$, $K_2$, $R$, and $\mathrm{h}(\cdot)$ into $U$'s smart card.
4) $S$ finishes the registration procedure by delivering the completed smart card to $U$.

### 2.2 Login Phase

In this phase, the user $U$ sends a login request message to the server $S$ whenever $U$ needs to access resources upon $S$.

1) $U$ inserts his smart card into a smart card reader and then inputs his $ID$ and $PW$.
2) Using $PW$, the smart card computes $C_1 = R \oplus \mathrm{h}(PW)$. If $C_1$ is not equal to the stored $K_1$, then the smart card rejects $U$'s login request. Otherwise, it computes $C_1' = K_2 \oplus \mathrm{h}(PW \oplus \mathrm{h}(PW))$ and then $C_2 = \mathrm{h}(C_1' \oplus T_1)$, where $T_1$ is the current timestamp.
3) $U \rightarrow S$ $\{ID, T_1, C_1, C_2\}$.

### 2.3 Verification Phase

In this phase, the server $S$ verifies the authenticity of the login message requested by the user $U$.

1) Upon receiving the message $\{ID, T_1, C_1, C_2\}$, $S$ checks the validity of $ID$ and the freshness of $T_1$. The freshness of $T_1$ is checked by performing $T' - T_1 \leq \Delta T$, where $T'$ is the time that $S$ receives the above message and $\Delta T$ is a valid time interval. If $ID$ is not valid or $T_1$ is not fresh, $S$ aborts the current session. Otherwise, $S$ computes $N' = C_1 \oplus \mathrm{h}(ID \oplus x)$ and checks that $\mathrm{h}(\mathrm{h}(ID \oplus x \oplus N') \oplus T_1)$ is equal to the received $C_2$. If not, $S$ terminates the current session. Otherwise, $S$ successfully authenticates $U$ and computes $C_3 = \mathrm{h}(\mathrm{h}(ID \oplus x \oplus N') \oplus C_2 \oplus T_2)$, where $T_2$ is the current timestamp.
2) $S \rightarrow U$ $\{T_2, C_3\}$.
3) Upon receiving the message $\{T_2, C_3\}$, $U$ checks the freshness of $T_2$ in the same way as described above. If $T_2$ is not fresh, $U$ terminates the current session. Otherwise, $U$ checks again to see if $\mathrm{h}(C_1' \oplus C_2 \oplus T_2)$ is equal to the received $C_3$. If it is not equal, $U$ terminates the current session. Otherwise, $U$ successfully authenticates $S$.

### 2.4 Password Change Phase

In this phase, the user $U$ is able to change his password at any time.

1) $U$ inserts his smart card into a smart card reader and then types in his $ID$ and $PW$.
2) The smart card computes $K_1' = R \oplus \mathrm{h}(PW)$ and compares $K_1'$ with the stored $K_1$. If they are not equal, then the smart card rejects the password change request. Otherwise, $U$ chooses a new password $PW'$.
3) The smart card then computes $R' = K_1' \oplus \mathrm{h}(PW')$ and $K_2' = K_2 \oplus \mathrm{h}(PW \oplus \mathrm{h}(PW)) \oplus \mathrm{h}(PW' \oplus \mathrm{h}(PW'))$ and replaces $R$ and $K_2$ with the newly updated $R'$ and $K_2'$, respectively.

## 3. Cryptanalysis of Kim and Chung's Scheme

In this section, we describe the security flaws of Kim and Chung's scheme and depict how an offline password guessing attack, an unlimited online password guessing attack, and server impersonation can be executed.

### 3.1 Offline Password Guessing Attack

An offline password guessing attack [9]–[12] means that an attacker attempts to identify a user's password in an offline manner. Specifically, the attacker is allowed to freely guess and verify a password and has an unlimited number of guesses. In general, because users tend to choose simple and weak passwords for their personal convenience [11], [12], [14]–[16], the attacker can easily obtain a user's password via the offline password guessing attack within a reasonable time. Due to the fact that users tend to use the same password in several servers for convenience [9], [12], [16], the attacker can log into the servers as a legitimate user after purloining the user's password. For these reasons, all the password-based user authentication schemes should be designed to prevent the allowance of an offline password guessing attack. Kim and Chung affirmed that their scheme is able to thwart the offline password guessing attack even if an attacker can extract secret values stored in a user's smart card. Unlike their assertion, however, their scheme is still vulnerable to this situation. The scenario of an offline password guessing attack is presented for their scheme. Note that $K_1$, $K_2$, $R$, and $\mathrm{h}(\cdot)$ stored in $U$'s smart card can be extracted in various ways [17], [18] after an attacker has stolen the smart card.

The attacker performs the following operations after obtaining a legitimate user's smart card somehow.

1) The attacker selects a password candidate $PW'$.
2) The attacker computes $R \oplus \mathrm{h}(PW')$.

3) The attacker compares the computed result with $K_1$. If they are equal, $PW'$ is the correct password. Otherwise, the attacker repeats the above steps until the correct password is found.

The attacker is able to derive $PW$ in an offline manner using one XOR, one hash function, and one comparison for each password candidate.

### 3.2 Unlimited Online Password Guessing Attack

Here is an online password guessing attack scenario without the limitation of the number of guesses using a security flaw of the password change phase. We call this attack an unlimited online password guessing attack.

An online password guessing attack [9]–[11] means that an attacker tries to use guessed passwords iteratively to pass the verification of the smart card in an online manner. Most of the password-based authentication schemes are insecure against an online password guessing attack. Hence, in these schemes, the possible number of password guesses is limited to mitigate the online password guessing attack; it is difficult for an attacker to find the correct password within the limitation.

In these schemes, if an online server limits the number of password guesses, an online password guessing attack can hardly succeed because of this limitation. In Kim and Chung's scheme, a smart card may limit the number of authentication trials without the help of the server for itself in order to prevent an online password guessing attack. However, if an attacker tampers with the smart card to remove the limitation of the number of authentication trials using some methods [19]–[21], the smart card cannot defend against the online password guessing attack any more in Kim and Chung's scheme. The attack scenario is as follows.

After an attacker obtains a legitimate user's smart card somehow and removes the limitation of the number of password guesses by tampering with the smart card, the attacker performs the following operations to obtain the user's password without the limitation of the number of guesses.

1) The attacker selects a password candidate $PW'$.
2) The attacker performs the password change phase with $ID$ and $PW'$. Note that the attacker can easily obtain $ID$, because $ID$ is sent in plaintext in the login phase.
3) If the smart card requests the attacker to enter a new password, $PW'$ is the correct password. Otherwise, the attacker repeats the above steps until the correct password is found. Note that these three steps can be automatically performed by tampering with the smart card.

The attacker is able to derive $PW$ using one XOR, one hash function, and one comparison for each password candidate.

If a password can be changed without the help of the server, an attacker can find the password via an online password guessing attack after stealing the smart card and removing the limitation of the number of guesses. Then, the attacker can replace the password with a new one. Hence, the number of password guesses should be limited by involving the server in the password change phase. If the number of authentication trials with an incorrect password is limited by the server, a password guessing attack can be averted. Hence, the password change phase should be performed through the server.

Note that it is possible for an attacker to steal a smart card due to its small size, light weight, and portability. A lost or stolen smart card can be compromised/tampered with in some ways [17]–[21]. However, it is impractical to steal, compromise, and tamper with a server, because in general a server is physically protected from theft, damage, and unauthorized access; a service provider may spend enough money to protect its server from theft, compromise, and tampering, because the server compromise and tampering are critical. Hence, we assume that the server is hardly compromised/tampered with.

### 3.3 Server Impersonation

In general, an attacker can impersonate 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 masquerade as the server even if a user's password is revealed. Kim and Chung's scheme allows an attacker to impersonate the server if the attacker obtains a user's password.

1) An attacker $A$ obtains a user's password using an offline password guessing attack and $K_2$ using the method described in Sect. 3.1.
2) The attacker $A$ computes $C_1' = K_2 \oplus \mathrm{h}(PW \oplus \mathrm{h}(PW))$.
3) When a user sends $ID$, $T_1$, $C_1$, and $C_2$ to the server in the login phase, $A$ intercepts the messages and sends $T_2$ and $C_3 = \mathrm{h}(C_1' \oplus C_2 \oplus T_2)$ to the user, where $T_2$ is the current timestamp.

Upon receiving $\{T_2, C_3\}$, the user then authenticates the attacker $A$ as the server $S$. Consequently, the attacker can successfully impersonate the server.

It is ideal to impersonate a server without a user's password in the attacker's viewpoint. In Kim and Chung's scheme, although this is impossible, an attacker can obtain other benefits by impersonating a server with a user's password; an attacker can violate a user's privacy and provide forged services to a user by impersonating a server with the victim's password.

The privacy violation scenario is as follows.

1) A user connects to an attacker masquerading 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 the received message. In addition,

the attacker can perform a man-i*n*-the-middle attack [22] by connecting to the genuine server with the victim's *ID* and password. As a result, all the exchanged messages 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. Proposed Scheme

We propose an improved scheme to overcome the demonstrated vulnerabilities of Kim and Chung's scheme based on the Rabin cryptosystem [23]. Furthermore, our scheme enables the smart card and the server to establish a session key between them. The session key can then be used for encrypting and authenticating messages after mutual authentication. The proposed scheme consists of the following four phases — registration, login, verification, and password change. We assume that all the random numbers used in our scheme are fresh.

### 4.1 Registration Phase

In this phase, the user $U$ initially registers with the server $S$ as follows.

1) $U$ => $S$ {*ID*, h(*PW*)}: $U$ chooses his *ID* and *PW*, and then sends *ID* and h(*PW*) over a secure communication channel to $S$.
2) Upon receiving *ID* and h(*PW*), $S$ derives $K_1 = ID \oplus x \oplus N$ and $K_2 = \mathrm{h}((K_1 \oplus p \oplus q)^x \bmod m)$, where $N$ is the number of times $U$ changes the password and $S$ initially sets $N$ to zero. Then $S$ computes a quantity $R = K_2 \oplus \mathrm{h}(PW)$.
3) $S$ stores the secure information $R$, $m$, $N$, and h(·) into $U$'s smart card.
4) $S$ finishes the registration procedure by delivering the completed smart card to $U$.

### 4.2 Login Phase

In this phase, the user $U$ sends a login request message to the server $S$ whenever $U$ needs to access resources upon $S$.

1) $U$ inserts his smart card into a smart card reader and then inputs his *ID* and *PW*.
2) The smart card chooses a random number $KM_U$, where the length of $KM_U$ is 128 bits.
3) Using *PW*, the smart card computes $C_1 = R \oplus \mathrm{h}(PW)$ and then $C_2 = (C_1 \| KM_U \| N \| T_1)^2 \bmod m$, where $T_1$ is the current timestamp.
4) $U \rightarrow S$ {*ID*, $T_1$, $C_2$}.

### 4.3 Verification Phase

In this phase, the server $S$ verifies the authenticity of the requested login message by the user $U$.

1) Upon receiving the message {*ID*, $T_1$, $C_2$}, $S$ checks the validity of *ID* and the freshness of $T_1$ by performing $T' - T_1 \leq \Delta T$, where $T'$ is the time that $S$ receives the above message and $\Delta T$ is a valid time interval.
2) $S$ decrypts $C_2$ using the private keys $p$ and $q$, based on the Rabin cryptosystem [23]. The decrypted result yields $C_1' \| KM_U' \| N' \| T_1'$. Then $S$ computes $D_1 = \mathrm{h}((ID \oplus x \oplus N' \oplus p \oplus q)^x \bmod m)$. If $C_1' = D_1$ and $T_1 = T_1'$ hold, $S$ successfully authenticates $U$, chooses a random number $KM_S$, and computes the session key $K_{SU}' = \mathrm{h}(KM_U' \| KM_S)$ and $D_2 = \mathrm{h}(K_{SU}' \oplus T_2)$, where $T_2$ is the current timestamp and the length of $KM_S$ is 128 bits.
3) $S \rightarrow U$ {$T_2$, $KM_S$, $D_2$}.
4) Upon receiving the message {$T_2$, $KM_S$, $D_2$}, $U$ checks the freshness of $T_2$ in the same manner stated above. $U$ computes the session key $K_{SU} = \mathrm{h}(KM_U \| KM_S)$ and checks again if $\mathrm{h}(K_{SU} \oplus T_2)$ is equal to the received $D_2$. If they are equal, $U$ successfully authenticates $S$.

### 4.4 Password Change Phase

In this phase, the user $U$ changes the password when desired as follows.

1) $U$ inserts his smart card into a smart card reader and then enters *ID*, *PW*, and *PW'*.
2) The smart card and the server perform a mutual authentication via the login and verification phases. They obtain a session key $K_{SU}$ during the mutual authentication.
3) If the mutual authentication is successful, the server calculates $D_3 = \mathrm{h}(K_{SU}) \oplus \mathrm{h}((ID \oplus x \oplus (N+1) \oplus p \oplus q)^x \bmod m)$.
4) $S \rightarrow U$ {$D_3$}.
5) The smart card computes $R' = \mathrm{h}(K_{SU}) \oplus D_3 \oplus \mathrm{h}(PW')$ and replaces $R$ with $R'$.
6) The smart card sets $N = N + 1$.

## 5. Security Analysis

In this section, we analyze our proposed scheme in terms of security. The proposed scheme is safe from an offline password guessing attack, an unlimited online password guessing attack, server and user impersonation attacks, and a replay attack, guarantees forward and backward secrecies of

a session key, and provides key freshness. We also provide the formal correctness verification of the proposed scheme.

### 5.1 Offline Password Guessing Attack

In Kim and Chung's scheme, an attacker can obtain every authentication parameter except the password because the parameters are sent in plaintext and saved in a smart card. After gathering the parameters, the attacker performs an offline password guessing attack using the parameters and a password candidate in order to find the correct password. This attack can be avoided by either creating strong passwords or by not allowing the authentication parameter to be disclosed by the attacker. However, because users tend to choose simple and weak passwords for their convenience [11], [12], [14]–[16], concealing an authentication parameter may be the only available option.

The authentication parameter is concealed via symmetric encryption with a shared key. However, the shared key can be disclosed by the attacker because the shared key is also saved in the smart card for future use. Hence, asymmetric cryptography, such as the Rivest-Shamir-Adleman (RSA) cryptosystem and the Rabin cryptosystem, should be used to conceal the authentication parameter. Our scheme adopts the Rabin cryptosystem in order to conceal an authentication parameter $C_1$. In our proposed scheme, the server's private decryption key is not stored in the smart card. Although the public encryption key is saved in the smart card, the attacker is unable to derive the private decryption key from the public key, even if an attacker extracts the public key from the smart card. This is because it is based on the difficulty of integer factorization. In addition, $K_2$ is not stored in the smart card. The attacker cannot derive h($PW$) from $R$ without $K_2$ and $C_1$ in our scheme. Consequently, our scheme is secure against an offline password guessing attack.

### 5.2 Unlimited Online Password Guessing Attack

In the proposed scheme, the password change phase can only be performed through the server; a user cannot change his/her password without the help of the server. Because the server is hardly tampered with as described in Sect. 3.1, the limitation of the number of password guesses is not removed in our scheme. Hence, our scheme is secure against an unlimited online password guessing attack.

### 5.3 Server Impersonation

In our scheme, even if an attacker obtains the user's password and secrets stored in the smart card, the attacker cannot impersonate the server. This is because our scheme adopts asymmetric cryptography. Without the knowledge of the server's private keys, the attacker cannot obtain values used for server authentication, such as $K_{SU}$.

### 5.4 User Impersonation

All the messages exchanged between a user and the server except $ID$ are refreshed for each authentication trial in our proposed scheme. Hence, an attacker cannot impersonate a user by replaying messages used for authentication after gathering these messages. Even if an attacker obtains a legitimate user's $ID$ which is not refreshed, he/she cannot masquerade as the user because of an unknown value which is the user's password. In our scheme, an attacker is unable to find the correct password as described in Sect. 5.1. Since an attacker cannot generate the valid messages required for authentication without the correct password, an attacker cannot impersonate a legitimate user. Note that our scheme cannot prevent a user impersonation attack if an attacker can obtain some secret information, such as the server's private keys, from the server. If an attacker would have the ability to extract secret keys from the server, however, server and user impersonation attacks might be possible not only in our scheme but also in most of the authentication schemes.

### 5.5 Replay Attack

To prevent a replay attack, all the authentication messages should be fresh for each login phase. Our scheme guarantees the freshness of the login messages because of a random number $K_{SU}$ and a timestamp $T_1$. Hence, our proposed scheme is able to withstand the replay attack.

### 5.6 Forward and Backward Secrecies

Forward secrecy means that the secrecy of previous established session keys should be maintained even if a session key is exposed [24]. Backward secrecy means that the secrecy of next session keys should be maintained even if an old session key is exposed [24]. In our scheme, a user and the server share a session key between them after mutual authentication. This session key is generated independently for each session. Hence, even if a session key is revealed, the previous and next sessions between the user and the server are secure.

### 5.7 Key Freshness

Key freshness means that neither party can predetermine the shared secret key being established. In our scheme, the session key is computed using two random numbers $KM_U$ and $KM_S$ generated by a user and the server, respectively. $KM_U$ is unknown to the server until a user sends it. Similarly, $KM_S$ is unknown to a user until the server transmits it. Hence, a user and the server cannot predetermine the session key, so we claim that our scheme guarantees key freshness.

### 5.8 Efficiency

Our scheme, which protects the server as well as the users,

is based on the Rabin cryptosystem. The cryptosystem can be adopted into our proposed scheme because a user should perform only one modular multiplication per mutual authentication. According to [25], a smart card has an ability to perform up to 3000 modular multiplications per second with the size of the modulus being 1024. 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.

### 5.9 Correctness Verification

We formally verify the correctness of the proposed scheme based on the BAN logic [26]. 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 $m$ is the public key of $S$; other notations follow those of the BAN logic [26]. We focus on the messages exchanged for mutual authentication and key agreement between a user and the server 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) $\boldsymbol{U}| \equiv \overset{\boldsymbol{m}}{\mapsto} \boldsymbol{S}$;
2) $\boldsymbol{S}| \equiv \overset{\boldsymbol{m}}{\mapsto} \boldsymbol{S}$;
3) $\boldsymbol{U}| \equiv \#(\boldsymbol{KM_U})$;
4) $\boldsymbol{S}| \equiv \#(\boldsymbol{KM_S})$;
5) $\boldsymbol{U}| \equiv \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S}$;
6) $\boldsymbol{S}| \equiv \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S}$;
7) $\boldsymbol{U}| \equiv \boldsymbol{S}| \equiv \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S}$;
8) $\boldsymbol{S}| \equiv \boldsymbol{U}| \equiv \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S}$;
9) $\boldsymbol{U}| \equiv \boldsymbol{S}| \Rightarrow \boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S}$;
10) $\boldsymbol{S}| \equiv \boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S}$;
11) $\boldsymbol{S}| \equiv \boldsymbol{U}| \sim (\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S})$.

Assumptions 1) and 2) state that $U$ and $S$ believe that $S$ possesses a public key $m$. Assumptions 3) and 4) mean that $U$ and $S$ generate two fresh random numbers $KM_U$ and $KM_S$ 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 $KM_U$ and sends it to $S$ with the shared secret $C_1$, $K_{SU}$ is finally determined by the random number $KM_S$ 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, respectively.

The verification is shown as follows:

Message 1 $\boldsymbol{U} \rightarrow \boldsymbol{S} : \{\boldsymbol{KM}_U, \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S}\}_m$.

12) $\boldsymbol{S} \triangleleft (\boldsymbol{KM}_U, \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S})$;
13) $\boldsymbol{S}| \equiv \#(\boldsymbol{KM}_U, \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S})$;
14) $\boldsymbol{S}| \equiv \boldsymbol{U}| \sim (\boldsymbol{KM}_U, \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S})$;
15) $\boldsymbol{S}| \equiv \boldsymbol{U}| \equiv (\boldsymbol{KM}_U, \boldsymbol{U} \overset{C_1}{\rightleftharpoons} \boldsymbol{S})$.

Message 2 $\boldsymbol{S} \rightarrow \boldsymbol{U} : \{\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S}\}\boldsymbol{K}_{SU}$.

16) $\boldsymbol{U} \triangleleft \{\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S}\}\boldsymbol{K}_{SU}$;
17) $\boldsymbol{U}| \equiv \boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S}$;
18) $\boldsymbol{U}| \equiv \#(\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S})$;
19) $\boldsymbol{U}| \equiv \boldsymbol{S}| \sim (\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S})$;
20) $\boldsymbol{U}| \equiv \boldsymbol{S}| \equiv (\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S})$;
21) $\boldsymbol{S}| \equiv \boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S}$;
22) $\boldsymbol{S}| \equiv \#(\boldsymbol{U} \overset{\boldsymbol{K_{SU}}}{\longleftrightarrow} \boldsymbol{S})$.

In the login phase (Message 1), a user calculates the shared secret $C_1$ using $R$ and $PW$ and then securely sends $C_1$ and a fresh random number $KM_U$ to $S$. The Message 1 is fresh for each authentication attempt because of the random number $KM_U$. Because of the shared secret $C_1$, $S$ can authenticate $U$. In the verification phase (Message 2), $S$ generates a fresh random number $KM_S$ and calculates the session key $K_{SU}$ shared between $S$ and $U$ using $KM_U$ and $KM_S$. Then, $S$ proves that it can generate $K_{SU}$ by sending $D_2$ which is generated using $K_{SU}$. Note that only $S$ can generate $K_{SU} = \mathrm{h}(KM_U \| KM_S)$, because only the entity that has the corresponding private key of $S$'s public key $m$ can find $KM_U$ from the Message 1.

## 6. Discussion

We discuss two issues important for practical implementation of our scheme in this section; we should consider the use of the XOR operation on inputs of different lengths and communication cost of our scheme.

In the proposed scheme, the XOR operation is used. This operation needs to have the inputs with the same length and takes the output of the same length as the input length. In our scheme, however, the lengths of two input strings are different in some computations. Furthermore, $ID$ and $PW$, which are chosen by a user, do not always take the same length, and thus the lengths of $ID$ and $PW$ require to be adjusted.

In our scheme, the XOR operation can be performed even if the lengths of two strings are different by lengthening the shorter one as follows.

If a 4-bit string $A$ is 0100 and a 6-bit string $B$ is 110010, we can compute $C = A \oplus B$ by adding the first two bits of $A$ into the end of $A$. That is, $C = 010001 \oplus 110010 = 100011$ holds.

If a 4-bit string $A$ is 0100 and a 10-bit string $B$ is 1100100101, we can compute $C = A \oplus B$ by concatenating $A$ into the end of $A$ and then adding the first two bits of $A$ into the end of $A$. That is, $C = 0100010001 \oplus 1100100101 = 1000110100$ holds.

The lengths of $ID$ and $PW$ can be uniformly adjusted using the above-mentioned method. In this paper, we de-

fine *ID* and *PW* are 128-bit lengths, respectively. If a user chooses and enters 48-bit *ID*, the smart card concatenates *ID* into the end of *ID* and then adds the first 32 bits of *ID* into the end of *ID* automatically to make *ID* a 128-bit string.

In the login phase of our scheme, a user sends 128 + 32 + 1024 bits, where the lengths of *ID*, $T_1$, and *m* are 128 bits, 32 bits, and 1024 bits, respectively. In the verification phase, a server sends 32 + 128 + 256 bits, where the lengths of $T_2$, $KM_S$, and a hash output are 32, 128, and 256 bits, respectively. It requires about 16 milliseconds to send all the messages at 100 kbps. In Kim and Chung's scheme, it requires about 9.6 milliseconds to send all the messages at the same rate. Users can hardly be aware of this trivial difference, because this delay is too short to recognize. Note that the Rabin cryptosystem with 512-bit private keys might offer the same security as the RSA cryptosystem with a 1024-bit modulus, because the Rabin cryptosystem with 512-bit private keys uses a 1024-bit modulus *m*.

To change a password in our scheme, a user and the server should exchange totally 1856 bits. This requires about 18.5 milliseconds at 100 kbps. In Kim and Chung's scheme, they do not have to exchange any message. However, Kim and Chung's scheme has a security flaw against an unlimited online password guessing attack caused by the insecure password change phase. To remove this vulnerability, the password change phase should be performed through an online server. Although our scheme requires additional delay for the password change compared with Kim and Chung's scheme, the delay is tolerable. To eliminate the security weakness of Kim and Chung's scheme, this additional cost is unavoidable.

## 7. Conclusion

We demonstrated that Kim and Chung's scheme is vulnerable to an offline password guessing attack, an unlimited online password guessing attack, and server impersonation. We proposed an improved scheme to overcome the weaknesses based on the Rabin cryptosystem and analyzed the security of the proposed scheme. Our scheme is secure against the offline password guessing attack, the unlimited online password guessing attack, and the server impersonation. We provided the correctness verification using the BAN logic.

### References

[1] L. Lamport, "Password authentication with insecure communication," Commun. ACM, vol.24, no.11, pp.770–772, Nov. 1981.
[2] M.S. Hwang and L.H. Li, "A new remote user authentication scheme using smart cards," IEEE Trans. Consum. Electron., vol.46, no.1, pp.28–30, 2000.
[3] H.Y. Chien, J.K. Jan, and Y.M. Tseng, "An efficient and practical solution to remote authentication smart card," Comput. Secur., vol.21, no.4, pp.372–375, 2000.
[4] C.L. Hsu, "Security of Chien et al.'s remote user authentication scheme using smart cards," Computer Standards & Interfaces, vol.26, no.3, pp.167–169, 2004.
[5] S.W. Lee, H.S. Kim, and K.Y. Yoo, "Improved efficient remote user authentication scheme using smart cards," IEEE Trans. Consum. Electron., vol.50, no.2, pp.565–567, 2004.
[6] S.W. Lee, H.S. Kim, and K.Y. Yoo, "Improvement of Chien et al.'s remote user authentication scheme using smart cards," Computer Standards & Interfaces, vol.27, no.2, pp.181–183, 2005.
[7] E. Yoon and K. Yoo, "More efficient and secure remote user authentication scheme using smart cards," Proc. 11th International Conference on Parallel and Distributed System, vol.2, pp.73–77, 2005.
[8] S.K. Kim and M.G. Chung, "More secure remote user authentication scheme," Comput. Commun., vol.32, no.6, pp.1018–1021, 2009.
[9] W.C. Ku, "Weaknesses and drawbacks of a password authentication scheme using neural networks for multiserver architecture," IEEE Trans. Neural Netw., vol.16, no.4, pp.1002–1005, 2005.
[10] C.C. Yang and R.C. Wang, "Cryptanalysis of improvement of password authenticated key exchange based on RSA for imbalanced wireless networks," IEICE Trans. Commun., vol.E88-B, no.11, pp.4370–4372, Nov. 2005.
[11] T. Cao and D. Lin, "Cryptanalysis of two password authenticated key exchange protocols based on RSA," IEEE Commun. Lett., vol.10, no.8, pp.623–625, 2006.
[12] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," Proc. 9th ACM Conference on Computer and Communications Security (CCS'02), pp.161–170, 2002.
[13] C.L. Lin and T. Hwang, "A password authentication scheme with secure password updating," Comput. Secur., vol.22, no.1, pp.68–72, 2003.
[14] B.T. Hsieh, H.M. Sun, and T. Hwang, "On the security of some password authentication protocols," Informatica, vol.14, no.2, pp.195–204, 2003.
[15] C.L. Lin and T. Hwang, "A password authentication scheme with secure password updating," Comput. Secur., vol.22, no.1, pp.68–72, 2003.
[16] L. Gong, M.A. Lomas, R.M. Needham, and J.H. Saltzer, "Protecting poorly chosen secrets from guessing attacks," IEEE J. Sel. Areas Commun., vol.11, no.5, pp.648–656, 1993.
[17] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Proc. Advances in Cryptology (CRYPTO'99), pp.388–397, 1999.
[18] T.S. Messerges, E.A. Dabbish, and R.H. Sloan, "Examining smart-card security under the threat of power analysis attacks," IEEE Trans. Comput., vol.51, no.5, pp.541–552, 2002.
[19] V. Taponen, "Tamper-resistant smart cards — Too much to ask for?," Proc. Helsinki University of Technology Seminar on Network Security, Fall 2000.
[20] R. Anderson and M. Kuhn, "Low cost attacks on tamper resistant devices," Proc. 5th International Workshop on Security Protocols, LNCS 1361, pp.125–136, April 1997.
[21] R. Anderson and M. Kuhn, "Tamper resistance: A cautionary note," Proc. 2nd USENIX Workshop on Electronic Commerce (WOEC'96), vol.2, pp.18–21, 1996.
[22] Z. Chen, S. Guo, K. Zheng, and Y. Yang, "Modeling of man-in-the-middle attack in the wireless networks," Proc. International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'07), pp.2255–2258, 2007.
[23] M.O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Technical Report: TR-212, Massachusetts Institute of Technology, 1979.
[24] P. Sakarindr and N. Ansari, "Security services in group communications over wireless infrastructure, mobile ad hoc, and wireless sensor networks," IEEE Wirel. Commun., vol.14, no.5, pp.8–20, 2007.
[25] C. Zouridaki, B.L. Mark, K. Gaj, and R.K. Thomas, "Distributed CA-based PKI for mobile ad hoc networks using elliptic curve cryptography," Lect. Notes Comput. Sci., 3093, pp.232–245, 2004.
[26] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," Proc. Royal Soc. London A, vol.426, pp.233–271, 1989.

**Jung-Yoon Kim** is a Ph.D. student in Mobile Systems Engineering at Sungkyunkwan University in South Korea. He received his B.S. degree (2006) in Computer Engineering and M.S. degree (2008) in Electrical and Computer Engineering from Sungkyunkwan University in South 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.

**John A. Copeland** received the B.S., M.S., and Ph.D. degrees in physics from the Georgia Institute of Technology (Georgia Tech). He holds the John H. Weitnauer, Jr., Chair as a professor in the School of Electrical and Computer Engineering at Georgia Tech, and is a Georgia Research Alliance Eminent scholar. He was the Vice President of Technology at Hayes (1985–1993) and the Vice President of Engineering Technology at Sangamo Weston, Inc. (1982–1985), and served at Bell Labs (1965–1982). He founded Lancope, Inc. (2000) and invented the StealthWatch network security monitoring system. He has been awarded 48 patents and has published more than 100 technical articles. In 1970, he received the IEEE's Morris N. Liebmann Award. He is a fellow of the IEEE.