A Stubborn Security Model Based on Three-factor Authentication and Modified Public Key

Trung Thanh Ngo and Tae-Young Choe
(Corresponding author: Tae-Young Choe)

Department of Computer Engineering, Kumoh National Institute of Technology
61 Daehak-ro, Gumi-si, Gyeongsangbuk-do, 730-701, Republic of Korea
(Email: choety@kumoh.ac.kr)

(Received Aug. 15, 2015; revised and accepted Nov. 27, 2015)

Abstract

Single authentication methods such as password, smart card, or biometric authentication suffer from their own weaknesses. Thus, combined authentication methods have been proposed recently. Unfortunately, even combined authentication methods are exposed to replay attacks, user impersonation attacks, server masquerading attacks, or stolen smart card attacks. To minimize the range of such attacks, we propose a security model that combines smart card authentication and biometric authentication using a modified public key cryptography. The modified public key cryptography transfers a public key only to the opposite entity not to public. The proposed security model can withstand the above-mentioned attacks. In particular, the insider attack can be resisted even in cases where the secret values stored in any two of three parties of a system are compromised. Such tolerance is enabled by modified public keys which are not revealed to the third party.

Keywords: Biometric authentication, modified public key signature, smart card, three-factor authentication, three-entity security model

1 Introduction

The rapid growth in cloud computing has recently offered many benefits to customers, such as large-scale computations and data storage, virtualization, high expansibility, high-reliability, and low service costs. Although cloud computing offers several benefits, it has indisputably given rise to many security issues; this is because a customer's data is stored on a cloud for access and processing. Among these issues, the authentication between a client and server is very critical because an intruder can break into a cloud system, and steal or modify a customer's sensitive data if the authentication scheme is weak.

Traditional remote identity authentication schemes mainly rely on the use of passwords between clients and servers [7, 8, 14]. However, simple passwords have been revealed easily through simple dictionary attacks.

To overcome this problem, a smart card and a password are used together to verify the identity of a user [1, 2, 4, 10, 11, 16, 17, 18, 19, 20]. A problem with this method is that the scheme is unable to provide non-repudiation owing to the fact that: Smart cards and passwords can be lost, forgotten, or even shared with other people. Therefore, it is impossible to determine the actual owner.

Fortunately, owing to the personal identity property of biometric keys (fingerprints, irises, hand geometry, faces, and so on), biometric verification can be included to provide non-repudiation. The advantages of using the additional properties of the biometric keys described in [11] are as follows:

- Biometric keys cannot be lost or forgotten.
- Biometric keys are very difficult to copy or share.
- Biometric keys are extremely hard to forge or distribute.
- Biometric keys cannot be guessed easily.
- The biometrics of one individual is not easier to break than that of another individual.

In combination with smart card and password verification, biometric verification makes remote authentication more secure and reliable. As a result, several researches have been conducted along this direction [3, 11, 12, 13].

Li and Hwang proposed an efficient biometric-based remote user authentication scheme using smart cards [11]. They claimed that besides maintaining good properties, such as unneeded synchronized clock, easy to change passwords, low computation costs, and mutual authentication their method also provides non-repudiation because of the use of personal biometrics [14]. However, Das showed that Li and Hwang's scheme retains flaws in its login, authentication and password change phases, as well as in the verification of biometrics using a hash function [6]. Das proposed an improved scheme that resolves the flaws inherent to Li and Hwang's scheme. It maintains the scheme's good properties, and provides strong security against various attacks such as user impersonation attacks, server masquerading attacks, parallel session attacks, and stolen password attacks.

However, we found that some security flaws persist in Das scheme as well. In the case where the secret information stored in a smart card or on a server is lost, an attacker can conduct various kinds of attacks, such as a password change attack, server masquerading attack, or user impersonation attack. In this paper, we propose a new remote authentication scheme that improves on Das scheme, and shows how the new scheme can withstand the above-mentioned flaws. The proposed scheme sets the relations among the three entities: A biometric server, an authentication server, and a smart card representing a user. Each entity has single-side secrets to prevent an attack from dominating the system by leaking the secrets of the entities. In order to implement the single-side secrets, traditional public key distribution is modified. An entity has a private key and corresponding public key is assigned only to the pre-defined target entity. Although a message is encrypted with the private key, only the target entity can decrypt the message. Such modified public key system enables to detect whether the target entity is impersonated or not.

The rest of this paper is organized as follows. In Section 2, we briefly review Das scheme. The design flaws of Das scheme are discussed in Section 3. Section 4 presents our new scheme, which withstands the flaws discussed in Section 3. Next, the strength of our scheme is discussed in Section 5. Finally, we provide some concluding remarks in Section 6.

2 Review of Das Scheme

In Das scheme, there are four phases:

- Registration phase;
- Login phase;
- Authentication phase;
- Password change phase.

Notations used in this paper are shown in Table 1.

2.1 Registration Phase

To register with a trusted registration center TRC, a user must conform to the following steps:

1) User U_i provides TRC with their personal biometrics identity B_i and password PW_i personally.
2) Then, TRC computes the following:
   - $f_i \leftarrow h\ (B_i)$;
   - $r_i \leftarrow h\ (PW_i) \oplus f_i$;
   - $e_i \leftarrow h\ (ID_i \parallel X_i) \oplus r_i$.
3) TRC stores ID_i, $h(.)$, f_i, e_i, and r_i into the smart card SM_i and sends the smart card to the user in person.

2.2 Login Phase

To log in to the system, the user must adhere to the following 6 steps:

1) User U_i inserts a smart card into a card reader and offers her/his biometrics identity B_i' on a specific device for verification.
2) Biometric identity B_i' is matched against the biometrics template B_i of the user stored in the system.
3) If B_i' matches successfully, U_i passes the biometrics verification step and continues to Step 4. Otherwise, abort the remote authentication.
4) User U_i inputs password PW_i. The smart card SM_i computes $r_i' = h(PW_i) \oplus f_i$ and compares the result with r_i.
   - If $r_i = r_i'$, continue Step 5.
   - Otherwise, terminate the remote authentication.
5) The smart card SM_i computes the following:
   - $M_1 \leftarrow e_i \oplus r_i' = h(ID_i \parallel X_i)$;
   - $M_2 \leftarrow M_1 \oplus R_u = h(ID_i \parallel X_i) \oplus R_u$;
   - $M_3 \leftarrow h(R_u)$.
6) User U_i sends the message (ID_i, M_2, M_3) to S.

2.3 Authentication Phase

After receiving (ID_i, M_2, M_3) from U_i, server S processes the following ten steps:

1) Server S checks the format of the message.
   - If it is valid, continue to Step 2.
   - Otherwise, abort the login request.
2) Server S computes the following:
   - $M_4 \leftarrow h(ID_i \parallel X_i)$;
   - $M_5 \leftarrow M_2 \oplus M_4 = R_u$.
3) Server S verifies whether $M_3 = h(M_5)$.

- If they are equal, continue to Step 4.
- Otherwise, abort the login request.

4) Server S computes the following:
   - $M_6 \leftarrow M_4 \oplus R_s$;
   - $M_7 \leftarrow h(M_2 \parallel M_5)$;
   - $M_8 \leftarrow h(R_s)$.
5) Server S sends (M_6, M_7, M_8) to U_i.
6) After receiving (M_6, M_7, M_8), U_i verifies whether $M_7 = h(M_2 \parallel R_u)$.
   - If they are equal, continue to Step 7.
   - Otherwise, abort the login request.
7) U_i computes: $M_9 \leftarrow M_6 \oplus M_1$.
8) U_i verifies whether $M_8 = h(M_9)$.
   - If they are equal, U_i computes $M_{10} \leftarrow h(M_6 \parallel M_9)$.
   - Otherwise, abort the login request.
9) U_i sends (M_{10}) to S.
10) After receiving (M_{10}), S verifies whether $M_{10} = h(M_6 \parallel R_s)$.
    - If they are equal, S accepts the login request.
    - Otherwise, abort the login request.

2.4 Password Change Phase

To change their password, the user must conduct the following six steps:

1) U_i inserts a smart card into a card reader and offers her/his biometric identity B_i' on a specific device for verification.
2) B_i' is matched against the biometric template B_i of the user stored in the system.
   - If B_i' is valid, continue to Step 3.
   - Otherwise, abort the password change request.
3) U_i inputs the old password ${PW_i}^{old}$ and a new password ${PW_i}^{new}$.
4) The smart card SM_i computes the following:
   - $r_i' \leftarrow h({PW_i}^{old}) \oplus f_i$;
   - If $r_i = r_i'$, continue to Step 5;
   - Otherwise, abort the change password request.
5) The smart card SM_i computes the following:
   - $r_i'' \leftarrow h({PW_i}^{new}) \oplus f_i$;
   - $e_i' \leftarrow e_i \oplus r_i' = h(ID_i \parallel X_i)$;
   - $e_i'' \leftarrow e_i' \oplus r_i''$.
6) The smart card SM_i replaces e_i and r_i with e_i'' and r_i'', respectively.

3 Security Analysis of Das Scheme

In this section, we analyze the security of Das scheme based on the assumption that one of the following conditions is satisfied:

- An attacker can obtain all secret values of a smart card using a specific device to monitor the power consumption if they have one [5].
- An attacker can obtain all secret values of the server with the help of an insider of the server.
- An attacker can eavesdrop, intercept and modify messages sent between a user and server.

Under these assumptions, we analyze the following critical attacks on Das scheme.

3.1 Stolen Smart Card Attack

If a smart card is stolen, there is a possibility that an attacker can extract all secret values ID_i, $h(.)$, f_i, e_i, and r_i stored on the card using a specific device to monitor the power consumption [5]. With these values, the attacker can easily impersonate a legal user, as explained in Section 3.4 or masquerade as the server by forging authentication messages, as described in Section 3.5. The attacker can even guess or change the password of the smart card by conducting a password guessing attack, as described in Section 3.2 and a password changing attack, as detailed in Section 3.3.

3.2 Password Guessing Attack

The password can also be guessed using the following steps:

1) Attacker A uses secret values on the smart card to computes: $r_i \oplus f_i = h(PW_i)$.
2) A guesses password PW_i' and repeatedly verifies whether $h(PW_i') = h(PW_i)$ until the equation is satisfied.

A dictionary attack speeds up the guessing process [15].

3.3 Password Change Attack

Assume that legal user A_j picks up user U_i's smart card, and becomes an attacker. Attacker A_j can impersonate U_i and change the password in the smart card using her/his biometric information B_j. This can occur because there is no relationship between the biometrics and password verification processes. In addition, the password of the smart card can be broken using a simple dictionary attack. A dictionary attack is possible using the local password verification of the smart card, as shown in Step 4 of Section 2.2, and in Section 3.2.

In Li and Hwang's scheme [11], there is a relationship between the biometrics and password verification processes. After the biometrics verification, at the beginning of the login phase, it checks whether $f_i = h(B'_i)$ where B'_i is the biometrics template inputted by the user. Therefore, if another legal user steals a smart card, the hashed value of their biometric template B_j does not match the value of f_i stored in the smart card, and they cannot use the smart card to log in to the system. Unfortunately, this step was discarded in Das scheme [6], because of the flaws in the biometrics verification using a hash function. As the result, Das scheme is exposed to the password change attack easily.

3.4 User Impersonation Attack

If attacker A knows all secret values $(ID_i, h(.), f_i, e_i, r_i)$ of user U_i stored in a smart card, A can easily impersonate U_i through the following steps:

1) Attacker A generates and sends a message (ID_i, M_2^f, M_3^f) to server S, where
   - $M_1 \leftarrow e_i \oplus r_i$;
   - $M_2^f \leftarrow M_1 \oplus R_f$;
   - $M_3^f \leftarrow h(R_f)$;
   - R_f is a random number generated by A.
2) Server S receives (ID_i, M_2^f, M_3^f) and checks whether ID_i is valid. Because ID_i is valid, S computes the following:
   - $M_4 \leftarrow h(ID_i \parallel X_i)$;
   - $M_5^f \leftarrow M_2^f \oplus M_4$.
3) Server S checks and sees that $M_3^f = h(M_5^f)$, and then sends (M_6, M_7^f, M_8) to A, where
   - $M_6 \leftarrow M_4 \oplus R_s$;
   - $M_7^f \leftarrow h(M_2^f \parallel M_5^f)$;
   - $M_8 \leftarrow h(R_s)$.
4) Attacker A receives the message (M_6, M_7^f, M_8) and computes the following:
   - $M_9 \leftarrow M_6^f \oplus M_1$;
   - $M_{10} \leftarrow h(M_6 \parallel M_9)$.
5) Attacker A sends M_{10} to S.
6) After receiving (M_{10}), S verifies whether $M_{10} = h(M_6 \parallel R_s)$.
7) Server S grants access to the attacker.

At this point, the attacker has successfully impersonated user U_i.

3.5 Server Masquerading Attack

If attacker A knows all the secret values stored in a smart card, A can easily masquerade as the server by conducting the following steps:

1) Attacker A sends a message (M_6^f, M_7, M_8^f) to U_i, where
   - $M_4 \leftarrow e_i \oplus r_i$;
   - $M_5 \leftarrow M_2 \oplus M_4$;
   - $M_6^f \leftarrow M_4 \oplus R_f$;
   - $M_7 \leftarrow h(M_2 \parallel M_5)$;
   - $M_8^f \leftarrow h(R_f)$;
   - R_f is a random number generated by the attacker.
2) User U_i receives (M_6^f, M_7, M_8^f), verifies that $M_7 = h(M_2 \parallel R_u)$, and computes M_9^f where, $M_9^f \leftarrow M_6^f \oplus M_1$.
3) User U_i verifies that $M_8^f = h(M_9^f)$, and then it computes and sends (M_{10}) to the attacker, where $M_{10} \leftarrow h(M_6^f \parallel M_9^f)$.

At this point, the attacker has successfully masqueraded as the server.

3.6 Insider Attack

If attacker A knows the secret value X_i of user U_i stored on the server with the help of an insider, attacker A can impersonate user U_i by conducting the following steps:

1) Attacker A sends a message (ID_i, M_2^f, M_3^f) to the server S, where
   - $M_2^f \leftarrow h(ID_i \parallel X_i) \oplus R_f$;
   - $M_3^f \leftarrow h(R_f)$;
   - R_f is a random number generated by the attacker.
2) Attacker A receives the reply message (M_6, M_7, M_8) from the server, computes M_{10}^f, and sends it to S, where
   - $M_9 \leftarrow M_6 \oplus h(ID_i \parallel X_i)$;
   - $M_{10}^f \leftarrow h(M_6 \parallel M_9)$.
3) Server S verifies that $M_{10}^f \leftarrow h(M_6 \parallel R_s)$. Attacker A then grants login access to server S.

Table 1: Notations

Notation	Description
U_i	User i
TRC	Trusted Registration Center
SM_i	Smart card of user i
S	Server
A	Attacker
PW_i	Password of the user i
ID_i	Identity of the user i
B_i	Biometric template of the user i
$h(.)$	A secure hash function
X_i	A secret information for user U_i maintained by the server S
$a \parallel b$	a concatenates b
$a \oplus b$	a Exclusive-OR b
R_u	A random number generated by the user U_i
R_s	A random number generated by the server S
R_{bs}	A random number generated by the biometric server BS
$\{M\}_U$	Encrypts message M using public key of user U
$[M]_U$	Encrypts message M using private key of user U

4 Enhanced Scheme

In this section, we propose an enhanced scheme that overcomes the flaws of Das scheme and resists the attacks discussed in Section 3.

To begin with, we describe the system model used in the enhanced scheme. Das scheme uses a client-server model consisting of a remote server and many clients. The client side includes a terminal, smart card reader, and local biometric verification system that takes the user's biometric template using a specific device and compares it with the user's biometric template that is stored in a local database.

However, the local biometrics verification system is not scalable for cloud computing systems because a local biometrics database cannot store the biometrics templates of innumerable cloud users. Therefore, biometrics verification must be processed by remote servers.

In our scheme, the system model consists of smart card SM_i, a biometrics server BS, and an authentication server S, as depicted in the Figure 1.

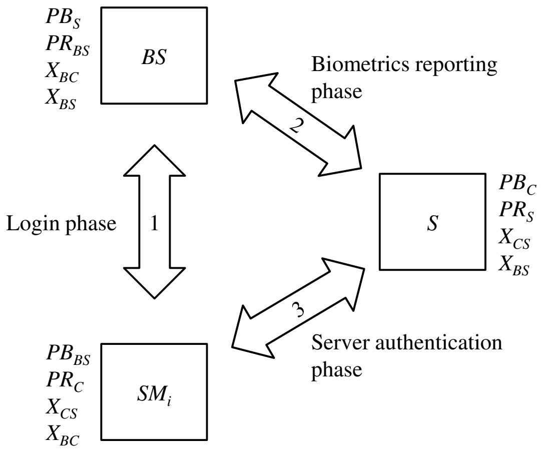

Figure 1: Authentication model in three entities

Further, we improve the security using public key cryptography along with random nonces. The public and private key pairs are distributed securely as depicted in Figure 1. These key pairs are used for the encryption and mutual authentication in each phase of the scheme. The public and private keys are used differently than a general public/private key pair. In general, a public key is freely accessible, and is used for encryption. However, our public key is only a partially secure key used to encrypt a users' plaintext and generate a ciphertext. The corresponding private key is used to decrypt the ciphertext. For example, user U generates private key PR_U and public key PB_U. U sends PB_U to another user W securely. If U sends an encrypted message $[M]_U$, only W can decrypt it using PB_U and W knows that M is sent from U. In addition, if W sends an encrypted message M_U, only U can decrypt the message using PR_U and know that N was sent from W. One thing we must remember is that a standard such as $PKCS$ by RSA Security Inc. cannot be used because a public key is included in a private key format [9]. The enhanced scheme consists of four phases which are Registration phase, Login phase, Biometrics reporting phase, and Server authentication phase. In order to access the server, the user must follow these phases sequentially.

4.1 Registration Phase

To log in to the system, the user must visit a registration center in person and register for an account. At the registration center, the user and center perform the following steps:

1) The trusted registration center TRC generates the following keys:
   - Private/public key pair PR_{BS}/PB_{BS} for use between BS and SM_i.
   - Private/public key pair PR_C/PB_C for use between SM_i and S.
   - Private/Public key pair PR_S/PB_S for use between S and BS.
   - Secret Key X_{BC} is shared between BS and SM_i.
   - Secret Key X_{CS} is shared between S and SM_i.
   - Secret Key X_{BS} is shared between BS and S.
2) The user inputs their biometric identity B_i on a specific device and offers their password PW_i.
3) TRC computes the following:
   - $f_i \leftarrow h(B_i)$;
   - $r_i \leftarrow h(PW_i) \oplus f_i$;
   - $e_i \leftarrow h(ID_i \parallel X_{CS}) \oplus r_i$.
4) TRC stores ID_i, $h(.)$, e_i, r_i, X_{BC}, X_{CS}, PB_B, and PR_C into a smart card SM_i and delivers the smart card to user U_i in person.
5) Finally, TRC distributes the remaining keys and information through a secure channel:
   - $PR_{BS}, PB_S, X_{BC}, X_{BS}, B_i$, and ID_i to biometric server BS.
   - $PR_S, PB_C, X_{CS}, X_{BS}, ID_i, PW_i$, and $h(ID_i \parallel h(PW_i))$ to authentication server S.

4.2 Login Phase

After registering for an account, user U_i can log in to the system by performing the following steps:

1) U_i inserts a smart card SM_i into a card reader and inputs their biometrics identity B'_i on a biometrics reading device.
2) The smart card SM_i uses the public key PB_{BS} to encrypt ID_i, and sends $\{ID_i\}_{BS}$ to the biometrics server BS.
3) When BS receives the message from SM_i, it uses the private key PR_{BS} to decrypt the message to obtain ID_i.
4) BS computes the following messages:
   - $M_1 \leftarrow h(ID_i \parallel X_{BC}) \oplus R_{bs}$;
   - $M_2 \leftarrow h(R_{bs})$;
   - R_{bs} is a random number generated by BS.
5) BS uses the private key PR_{BS} to encrypt and sends the message $[(M_1, M_2)]_{BS}$ to SM_i. Because its corresponding public key is not revealed to the public, the encrypted message is transferred securely. Simultaneously, the sender is authenticated as BS.
6) When SM_i receives $[(M_1, M_2)]_{BS}$ from BS, it uses the public key PB_{BS} to decrypt the message and then computes the following messages:
   - $M_3 \leftarrow h(ID_i \parallel X_{BC})$;
   - $M_4 \leftarrow M_1 \oplus M_3$.
7) SM_i verifies whether $h(M_4) = M_2$:
   - If this is not true, SM_i terminates the login phase.
   - Otherwise, the process continues to Step 8.
8) SM_i computes M_5 and sends a message $\{(M_5, B'_i)\}_{BS}$ encrypted with the public key PB_{BS} to BS, where $M_5 \leftarrow h(M_1||M_4)$.
9) After receiving $\{(M_5, B'_i)\}_{BS}$, BS uses its private key PR_{BS} to decrypt the message and then verifies whether $M_5 = h(M_1 \parallel R_{bs})$
   - If these are equals, BS compares B'_i with the biometric template B_i of user ID_i in the database.
     - If B'_i is valid, BS accepts the login request of SM_i, sends an acceptance message to SM_i and moves on to the biometrics report phase.
     - Otherwise, BS rejects the login request of SM_i.
   - Otherwise, BS rejects the login request of SM_i.

The login phase is summarized in Login phase box of Figure 2.

4.3 Biometrics Reporting Phase

After accepting the login request, the biometrics server BS must report the login result to server S by performing the following steps:

1) BS uses the public key PB_S to encrypt the user's id ID_i obtained in the login phase and sends $\{(ID_i, R_{bs})\}_S$ to S, where R_{bs} is a random number generated by BS.
2) S receives $\{(ID_i, R_{bs})\}_S$ and uses the private key PR_S to decrypt it.
3) S computes the following messages:
   - $M_1 \leftarrow h(ID_i \parallel X_{BS}) \oplus R_{ss}$;
   - $M_2 \leftarrow h(R_{ss})$;

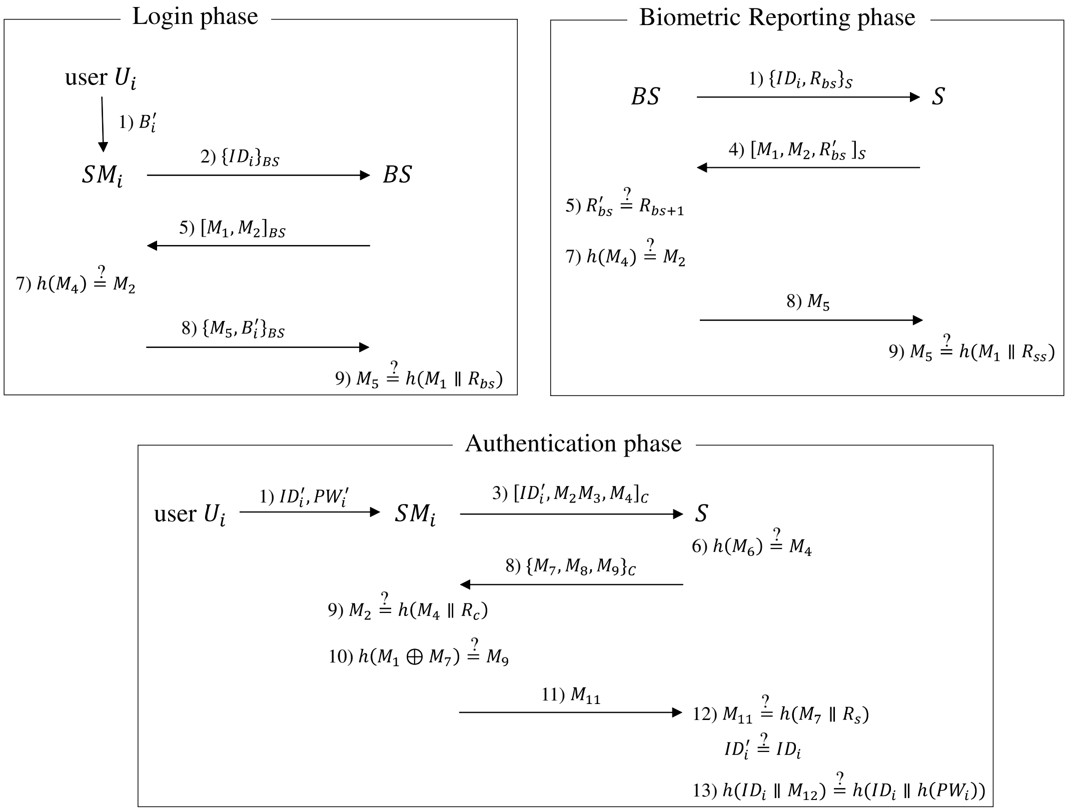
Login phase
user U_i
1) B_i'
SM_i
2) $\{ID_i\}_{BS}$
BS
5) $[M_1, M_2]_{BS}$
7) $h(M_4) \overset{?}{=} M_2$
8) $\{M_5, B_i'\}_{BS}$
9) $M_5 \overset{?}{=} h(M_1 \parallel R_{bs})$
Biometric Reporting phase
BS
1) $\{ID_i, R_{bs}\}_S$
S
4) $[M_1, M_2, R_{bs}']_S$
5) $R_{bs}' \overset{?}{=} R_{bs+1}$
7) $h(M_4) \overset{?}{=} M_2$
8) M_5
9) $M_5 \overset{?}{=} h(M_1 \parallel R_{ss})$
Authentication phase
user U_i
1) ID_i', PW_i'
SM_i
3) $[ID_i', M_2 M_3, M_4]_C$
S
6) $h(M_6) \overset{?}{=} M_4$
8) $\{M_7, M_8, M_9\}_C$
9) $M_2 \overset{?}{=} h(M_4 \parallel R_c)$
10) $h(M_1 \oplus M_7) \overset{?}{=} M_9$
11) M_{11}
12) $M_{11} \overset{?}{=} h(M_7 \parallel R_s)$
$ID_i' \overset{?}{=} ID_i$
13) $h(ID_i \parallel M_{12}) \overset{?}{=} h(ID_i \parallel h(PW_i))$

Figure 2: Three phases for remote user authentication

- R_{ss} is a random number generated by S;
- $R_{bs}' \leftarrow R_{bs} + 1$.

4) S uses the private key PR_S to encrypt and sends the message $[(M_1, M_2, R_{bs}')]_S$ to BS to authenticate BS.

5) When BS receives $[(M_1, M_2, R_{bs}')]_S$ from S, it uses the public key PB_S to decrypt the message and then verifies whether $R_{bs}' = R_{bs} + 1$.
   - If this is false, BS terminates the biometrics reporting phase.
   - Otherwise, the process continues to Step 6.

6) BS computes the following messages:
   - $M_3 \leftarrow h(ID_i \parallel X_{BS})$;
   - $M_4 \leftarrow M_1 \oplus M_3$.

7) BS verifies whether $h(M_4) = M_2$.
   - If this is false, BS terminates the biometrics reporting phase.
   - Otherwise, the process continues to Step 8.

8) BS computes M_5 and sends it to S, where $M_5 \leftarrow h(M_1||M_4) = h((h(ID_i \parallel X_{BS}) \oplus R_{ss}) ||R_{ss})$.

9) After receiving M_5, S verifies whether $M_5 = h(M_1 \parallel R_{ss})$.
   - If this is true, move on to the authentication phase.
   - Otherwise, S terminates the session.

The summary of the biometric reporting phase is shown in Biometric Reporting phase box of Figure 2.

4.4 Authentication Phase

After receiving the acceptance message from BS, server S authenticates user U_i by conducting the following steps:

1) U_i inputs their password PW_i' into the smart card SM_i.

2) SM_i computes the following messages:
   - $r_i' \leftarrow h(PW_i')$;
   - $M_1 \leftarrow h(ID_i' \oplus X_{CS})$;
   - $M_2 \leftarrow M_1 \oplus R_c$;
   - $M_3 \leftarrow r_i' \oplus R_c$;
   - $M_4 \leftarrow h(R_c)$;
   - R_c is a random number generated by SM_i.

3) SM_i uses the private key PR_C to encrypt and sends the message $[(ID_i', M_2, M_3, M_4)]_C$ to server S. PR_C can be used to encrypt the message is because its corresponding public key PB_C is not revealed to the public. PB_C is stored only in the server S.

4) S receives the message and uses the public key PB_C to decrypt it to obtain (ID'_i, M_2, M_3, M_4).

5) S computes the following messages:
   - $M_5 \leftarrow h(ID'_i \oplus X_{CS})$;
   - $M_6 \leftarrow M_5 \oplus M_2$.

6) S verifies whether $h(M_6) = M_4$. If this is false, S terminates the session. Otherwise, the process continues to Step 7.

7) S computes the following messages:
   - $M_7 \leftarrow M_5 \oplus R_s$;
   - $M_8 \leftarrow h(M_2 \parallel M_6)$;
   - $M_9 \leftarrow h(R_s)$;
   - R_s is a random number generated by S.

8) S uses the public key PB_C to encrypt and sends the message $\{(M_7, M_8, M_9)\}_C$ to SM_i.

9) SM_i receives $\{(M_7, M_8, M_9)\}_C$, and uses the private key PR_C to decrypt it. Then SM_i verifies whether $M_8 = h(M_2||R_c)$.

10) If this is false, SM_i terminates the session. Otherwise, SM_i verifies whether $h(M_{10}) = M_9$ where $M_{10} \leftarrow M_1 \oplus M_7$.

11) If this is false, SM_i terminates the session. Otherwise, SM_i computes and sends M_{11} to S where $M_{11} \leftarrow h(M_7 \parallel M_{10})$.

12) S receives M_{11} and verifies whether $M_{11} = h(M_7 \parallel R_s)$.
   - If this is false, S terminates the session.
   - Otherwise, S compares ID_i received from the biometrics server during the biometrics reporting phase with ID'_i received from the smart card in Step 4. If the two values of ID'_i do not match, S terminates the session. Otherwise, the process moves to Step 13.

13) S computes $h(ID_i \parallel M_{12})$ and compares the result with $h(ID_i \parallel h(PW_i))$ in the database, where $M_{12} \leftarrow M_3 \oplus M_6$.
   - If this is true, S accepts the login request.
   - Otherwise, S rejects the login request.

In this phase, a mutual authentication process between smart card SM_i and server S is used along with public key cryptography and random nonces. From Steps 1 through 3, smart card SM_i computes, encrypts, and sends messages containing its random nonce R_c to server S. From Steps 4 through 8, server S verifies the random nonce R_c, and computes, encrypts, and sends reply messages containing its random nonce R_s to smart card SM_i. From Steps 9 through 11, smart card SM_i verifies the random nonce R_s and, replies to server S. Finally, server S verifies the last reply message from SM_i, and conducts the smart card password verification in Steps 12 and 13, respectively. The authentication phase is summarized in Authentication phase box of Figure 2.

5 Security Analysis of the Enhanced Scheme

In this section, we analyze the security of the enhanced scheme based on the assumptions stated in Section 3 and show that our scheme can withstand all the mentioned attacks on Das scheme.

5.1 User Impersonation Attack

As mentioned in Section 3.1, Das scheme can be attacked easily if an attacker knows all the secret values stored in a user's smart card. However, this attack cannot succeed in our scheme. When attacker A tries to send the message $[(ID_i, M_2, M_3, M_4)]_C$ to S during the authentication phase, A uses the secret values of the smart card to compute fake messages M_2 and M_3, where

- $M_1 \leftarrow e_i \oplus r_i = h(ID_i \oplus X_{CS})$;
- $M_2 \leftarrow M_1 \oplus R_a$;
- $M_4 \leftarrow h(R_a)$;
- R_a is a random number generated by an attacker.

However, attacker A cannot generate M_3 because $M_3 = h(PW'_i) \oplus R_c$ where PW'_i is not stolen by attacker A. In addition, attacker A cannot obtain $h(PW'_i)$ after intercepting $[(ID_i, M_2, M_3, M_4)]_C$ because PB_C is needed to decrypt the message.

5.2 Server Masquerading Attack

As mentioned in Section 3.2, Das scheme can be attacked easily if an attacker knows all the secret values stored in a user's smart card. However, such an attack cannot work in our scheme because an attacker does not know the server's public key PB_C used to decrypt the message $[(ID_i, M_2, M_3, M_4)]_C$.

5.3 Password Guessing Attack

Unlike Das scheme, our proposed scheme does not store the secret value $f_i = h(B_i)$ into a smart card during the registration phase. Therefore, it is impossible for an attacker to execute this type of an attack.

5.4 Password Changing Attack

As pointed out in Section 3.3, there is no relationship between the biometrics and password verification processes in Das scheme. Thus, if attacker A obtains smart card SM_i of user U_i, A can impersonate U_i using U_i's biometric information.

In our scheme, however, this attack is impossible because after passing the login phase, the biometrics server reports the login user's id to the server in the biometrics reporting phase, as described in Section 4.3. Therefore, the server knows who has passed the login phase and is able to verify whether the smart card belongs to that person. This is conducted in the Steps 12 and 13 of the server authentication phase.

Moreover, to prevent this kind of attack, our scheme does not allow the user to change their smart card's password locally. If a user loses or forgets their smart card's password, they must contact the registration center to retrieve the password or create a new account.

5.5 Insider Attack

This attack is based on the assumption with the help of an insider, an attacker can acquire all the secret values stored in biometrics server BS, the server S or both. This assumption and the previous assumption on a loss of the smart card's secret values, motivated us to design a scheme that remains secure despite losing all secret values stored in any one or two parties of a system.

This expectation has been achieved in our scheme using public key cryptography. To demonstrate this, we consider the following cases in which secret values are lost.

Case 1: All secret values stored in BS and SM_i are compromised. Attacker A can pass the login and biometrics reporting phase, but not the server authentication phase. During the login phase, an attacker can easily impersonate a legal user by conducting the following steps:

1) Attacker A eavesdrops on an encrypted message $\{(M_5, B_i')\}_{BS}$ sent from SM_i to BS in Step 2 of the login phase (Section 4.1).

2) Attacker A uses the private key PR_{BS} to decrypt the message to obtain B_i'.

3) Later, A starts the login phase by sending message $\{ID_i\}_{BS}$ to BS to impersonate SM_i.

4) BS uses private key PR_{BS} to decrypt the message to obtain ID_i, and sends the message $[(M_1, M_2)]_{BS}$ back to A.
   - $M_1 \leftarrow h(ID_i \parallel X_{BC}) \oplus R_{bs}$;
   - $M_2 \leftarrow h(R_{bs})$;
   - R_{bs} is a random number generated by BS.

5) Attacker A uses the public key PB_{BS} to decrypt $[(M_1, M_2)]_{BS}$, forges M_5, and sends message $\{(M_5, B_i')\}_{BS}$ encrypted with public key PB_{BS} to BS, where
   - $M_3 \leftarrow h(ID_i \parallel X_{BC})$;
   - $M_4 \leftarrow M_1 \oplus M_3$;
   - $M_5 \leftarrow h(M_1 \parallel M_4) = h((h(ID_i \parallel X_{BC}) \oplus R_a) \parallel R_a)$;
   - R_a is a random number generated by A.

6) Biometric server BS verifies that M_5 and B_i are valid. BS then allows A to proceed to the server authentication phase. Next, BS automatically proceeds to the biometrics reporting phase with the ID_i provided by the attacker. Therefore, the attacker does not need to carry out an attack on the biometrics reporting phase.

In the server authentication phase, as explained in the Section 5.1, the attacker cannot impersonate the user if he has only the secret values of the user's smart card SM_i without password PW_i.

Case 2: All secret values stored in BS and S are compromised. Attacker A can pass only the biometrics reporting phase, but not the login phase or the server authentication phase.

During the biometrics reporting phase, the attacker A does not need to impersonate BS because the phase is processed automatically by both BS and S.

During the login phase, in order to impersonate the user U_i, A must have the public key PB_{BS} to decrypt the message $[(M_1, M_2)]_{BS}$ sent by BS in Step 6 of Section 4.2. After decrypting the message, A can use M_1 to forge reply message M_5 to trick BS into recognizing him as a legal user. However, A does not know PB_{BS}, which is stored only in SM_i. Consequently, A fails to impersonate U_i.

During the authentication phase, attacker A intercepts message $[(ID_i', M_2, M_3, M_4)]_{PR_C}$, and forwards it to S between Steps 3 and 4 of the authentication phase in order to impersonate the user U_i. When A receives the message $\{(M_8, M_9, M_{10})\}_{PB_C}$ in Step 9, A cannot decrypt it because PR_C is stored in smart card SM_i. In addition, A cannot generate M_{12}, which must be authenticated by S.

Case 3: All secret values stored in SM_i and S are compromised. Attacker A can pass both the biometrics reporting phase and the authentication phase.

During the login phase (Section 4.2), attacker A cannot impersonate a legal user because A does not know the biometric identity B_i' of the user U_i sent to the biometric server BS in Step 8. Moreover, it is impossible for A to retrieve B_i' without knowing the private key PR_{BS} stored in the biometric server.

However, attacker A knows all the secret values stored in smart card SM_i and server S, and can easily forge fake

Table 2: Analysis results of the insider attack

Compromised Sites			**Cracked Phases**			**Guard Values**
SM	*BS*	*S*	**Login**	**Biometrics Reporting**	**Authentication**	
O	O	X	O	O	X	PB_C
X	O	O	X	O	X	PB_{BS}, PR_C
O	X	O	X	O	O	PR_{BS}

messages needed to impersonate a legal user during the authentication phase. In addition, A can bypass the password verification in Step 13 of the authentication phase by conducting the following steps:

- Attacker A eavesdrops on message $[(ID_i, M_2, M_3, M_4)]_C$ sent from smart card SM_i in Step 3 of the authentication phase.
- A uses the public key PB_C to decrypt the message.
- A computes the following:
  - $M_1 \leftarrow e_i \oplus r_i = h(ID_i \oplus X_{CS})$;
  - $R_c \leftarrow M_1 \oplus M_2$;
  - $r'_i \leftarrow M_3 \oplus R_c = h(PW'_i)$.
- At this point, A can compute $h(ID_i \parallel h(PW'_i))$, and use it to pass Step 13 of the authentication phase.

During the biometrics reporting phase, attacker A does not need to impersonate BS because the phase is processed automatically by both BS and S. The Table 2 summarizes the analysis results of an insider attack.

6 Conclusion

In this paper, we reviewed and analyzed the security of Das scheme. We showed that this scheme still retains certain flaws that make it insecure against various types of attacks, particularly an insider attack. Therefore, we redesigned the system model and added the use of public key cryptography to overcome these security weaknesses, and made the system more secure against various types of attacks. For an insider attack, even when secret information stored in a smart card, in the biometrics server, or in the authentication server is revealed, an attacker still cannot pass the authentication process.

Acknowledgments

This research was supported by research fund, Kumoh National Institute of Technology.

References

[1] R. Amin, "Cryptanalysis and efficient dynamic ID based remote user authentication scheme in multi-server environment using smart card," *International Journal of Network Security*, vol. 18, pp. 172–181, January 2016.

[2] R. Amin and G. P. Biswas, "An improved rsa based user authentication and session key agreement protocol usable in tmis," *Journal of Medical Systems*, vol. 39, no. 8, pp. 1–14, 2015.

[3] R. Amin and G. P. Biswas, "A secure three-factor user authentication and key agreement protocol for tmis with user anonymity," *Journal of medical systems*, vol. 39, no. 8, pp. 1–19, 2015.

[4] R. Amin, T. Maitra and S. P. Rana, "An improvement of wang. et. al.'s remote user authentication scheme against smart card security breach," *International Journal of Computer Applications*, vol. 75, no. 13, pp. 37–42, 2013.

[5] Y. An, "Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards," *BioMed Research International*, vol. 2012, 2012.

[6] A. K. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *IET Information Security*, vol. 5, no. 3, pp. 145–151, 2011.

[7] L. Fan, J. H. Li and H. W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Computers & Security*, vol. 21, no. 7, pp. 665–667, 2002.

[8] 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, 2000.

[9] J. Jonsson and B. Kaliski, "Public-key cryptography standards (pkcs)# 1: Rsa cryptography specifications version 2.1," 2003. (`https://tools.ietf.org/html/rfc3447`) referenced at 3 March 2015.

[10] N. Y. Lee and Y. C. Chiu, "Improved remote authentication scheme with smart card," *Computer Standards & Interfaces*, vol. 27, no. 2, pp. 177–180, 2005.

[11] C. T. Li and M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[12] X. Li, J. W. Niu, J. Ma, W. D. Wang and C. L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.

[13] C. H. Lin and Y. Y. Lai, "A flexible biometrics remote user authentication scheme," *Computer Standards & Interfaces*, vol. 27, no. 1, pp. 19–23, 2004.

[14] J. J. Shen, C. W. Lin and M. S. Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Computers & Security*, vol. 22, no. 7, pp. 591–595, 2003.

[15] R. Shirey, "RFC 2828: Internet security glossary," 2000. (`http://tools.ietf.org/html/rfc2828`) referenced at 5 March 2015.

[16] S. K. Sood, A. K. Sarje and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[17] H. M. Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958–961, 2000.

[18] D. Wang, C. G. Ma, Q. M. Zhang and S. Zhao, "Secure password-based remote user authentication scheme against smart card security breach," *Journal of Networks*, vol. 8, no. 1, pp. 148–155, 2013.

[19] S. T. Wu and B. C. Chieu, "A user friendly remote authentication scheme with smart cards," *Computers & Security*, vol. 22, no. 6, pp. 547–550, 2003.

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

Trung Thanh Ngo received his bachelor's degree in Computer Science from Troy University, Alabama, USA in 2012. He received his MS degree in Computer Engineering from Kumoh National Institute of Technology, Gumi, South Korea in 2015. Currently, he is working as a system engineer for a telecommunication company in South Korea. His main research interest is secure authentication protocol in cloud system.

Tae-Young Choe received his bachelor's degree in Mathematical Education from Korea University, Seoul, South Korea in 1991. He received his MS and PhD degree in Computer Engineering from POSTECH, Pohang, South Korea in 1996 and 2002, respectively. Since 2002, he has been with KIT, Kumoh National Institute of Technology, Kumi, South Korea, where he is a professor in the Department of Computer Engineering. His main research interests are task management and secure authentication in cloud system.