SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2008; **1**:301–308
Published online 26 June 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.26

A bilateral remote user authentication scheme that preserves user anonymity

Huei-Ru Tseng, Rong-Hong Jan and Wuu Yang*,†
Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan

Summary

Smart card-based authentication is one of the most widely used and practical solutions to remote user authentication. Compared to other authentication schemes, our proposed scheme aims to provide more functionalities and to resist well-known attacks. These crucial merits include (1) a user can freely choose and change his passwords; (2) our scheme provides mutual authentication between a server and a user; (3) it achieves user anonymity; (4) a server and a user can generate authenticated sessions keys. Moreover, our scheme can resist replay attacks, forgery attacks, insider attacks, reflection attacks, and parallel session attacks. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: smart card; authentication; passwords; anonymity

1. Introduction

A remote user authentication scheme is a mechanism that authenticates remote users and allows legitimate users to access network services over an insecure communication network. In a distributed network, when a remote user requests for a service, the server should authenticate the user first. Due to high portability, low cost, and limited cryptographic capabilities of smart cards, a number of smart card-based remote authentication schemes have been proposed [1–22]. In 1981, Lamport [1] proposed the first password authentication scheme for remote users over an insecure channel. Since then, several schemes [2–22] have been proposed to improve security, efficiency, and functionality. Past experience has shown that constructing a secure user authentication scheme is not trivial because lots of proposed schemes were subsequently broken by well-known attacks [3,6–8,10,11,13,16].

Traditionally, if a remote user wants to log into a server, he has to submit his identity and password to the server. On receiving the login request, the server first checks the validity of the identity and computes a one-way hash value of the received password, and then checks the computed value against the server's verification table. Since this approach clearly incurs the risk of tampering and the cost of managing the table, several schemes [2,4,5,9,12,14,15,17–22] have been proposed that do not depend on a verification table.

Due to the constrained resources in smart cards, the computation and communication overhead must be low in practical implementation. Sun [19] proposed an efficient authentication scheme that adopts only simple hashing operations. In 2002, Chien *et al.* [4] proposed another authentication scheme that improves on Sun's in two ways: it achieves mutual authentication and it allows users to choose their passwords freely.

*Correspondence to: Wuu Yang, Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan.
†E-mail: wuuyang@cs.nctu.edu.tw

After a user is authenticated, the messages between the user and the server must be encrypted when transmitted over the public network. They have to agree on a session key. Juang [9] proposed an authentication scheme that provides a key agreement function. In various e-commerce applications, user anonymity is also crucial. Das *et al.* [5] first proposed a dynamic identity-based authentication scheme that preserves user anonymity. However, Chien and Chen [2] pointed out that Das *et al.*'s scheme [5] fails to protect user anonymity.

In order to reduce the risk of single-point failures, Choi and Youn [23] proposed a novel data encryption and distribution approach based on LU decomposition in 2004. The scheme allows higher security and availability compared with the mirroring scheme [24–26], and provides a solution for failures and malicious compromises of storage nodes, client systems, and user account. Pathan *et al.* [17,18] also proposed two bilateral authentication schemes based on LU decomposition. However, their schemes have several security weaknesses, including (1) they cannot resist replay attacks; (2) passwords could be revealed by the server; (3) they cannot preserve user anonymity; and (4) the server and users cannot agree on a session key.

To conquer these weaknesses, we propose a bilateral user-authentication scheme that not only fixes these weaknesses, but also aims to achieve more functionalities and resists well-known attacks. These crucial merits include (1) users can freely choose and change their passwords; (2) it provides mutual authentication between a server and a user; (3) it achieves user anonymity; (4) a server and a user can generate authenticated sessions keys. Moreover, the scheme is secure against replay attacks, forgery attacks, insider attacks, reflection attacks, and parallel session attacks.

The rest of this paper is organized as follows: In Section 2, we state the basic terms and preliminaries for our scheme. Our proposed scheme is presented in Section 3. Then, we shall analyze our proposed scheme, show that our scheme can resist several attacks, and provide a comparative study with other authentication schemes in Section 4. Finally, we will conclude our paper in Section 5.

2. Preliminaries

Our scheme is based on **LU** decomposition of matrices [27]. The decomposition re-writes a matrix as the product of a lower and an upper triangular matrices. In the **LU** decomposition, an $n \times n$ matrix $\mathbf{A}$ is written as

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U} \tag{1}$$

where $\mathbf{L}$ is a nonsingular lower triangular matrix, and $\mathbf{U}$ is a nonsingular upper triangular matrix.

In our scheme, a symmetric key matrix $\mathbf{A}_{n\times n}$ is generated by the server during system initialization, where n is the number of users that could be supported. This matrix is a secret of the server. In order to reduce the risk of single-point failures, with **LU** decomposition, the server can separate the symmetric key matrix $\mathbf{A}_{n\times n}$ to a lower and an upper triangular matrices and store these matrices in other servers.

Each element a_{ij} is a key from a key pool. We assume that $a_{ij} = a_{ji}$, for $1 \le i \le n$ and $1 \le j \le n$. Since $\mathbf{A}$ is symmetric, the product of the x-th row of matrix $\mathbf{L}$ and the y-th column of matrix $\mathbf{U}$ is as same as that of the y-th row of matrix $\mathbf{L}$ and the x-th column of matrix $\mathbf{U}$.

For example, given $\mathbf{A}$ as follows:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 4 & 5 \\ 2 & 5 & 8 & 9 \\ 4 & 8 & 15 & 17 \\ 5 & 9 & 17 & 20 \end{pmatrix} \tag{2}$$

we perform elementary row operations to get the lower matrix $\mathbf{L}$ and upper matrix $\mathbf{U}$ as follows:

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 0 & -1 & 0 \\ 5 & -1 & 0 & -3 \end{pmatrix} \text{ and}$$

$$\mathbf{U} = \begin{pmatrix} 1 & 2 & 4 & 5 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 2 \end{pmatrix} \tag{3}$$

Given $x = 2$ and $y = 3$, we can compute a_{23} and a_{32} as follows:

$$a_{23} = \mathbf{L}_R(2) \times \mathbf{U}_C(3) = \begin{pmatrix} 2 & 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 4 & 0 & 1 & 1 \end{pmatrix}^T = 8 \tag{4}$$

$$a_{32} = \mathbf{L}_R(3) \times \mathbf{U}_C(2) = \begin{pmatrix} 4 & 0 & -1 & 0 \end{pmatrix} \times \begin{pmatrix} 2 & 1 & 0 & 0 \end{pmatrix}^T = 8 \tag{5}$$

Since matrix $\mathbf{A}$ is symmetric, $a_{23} = a_{32}$. Note that $\mathbf{L}_R(2)$ denotes the 2nd row of matrix $\mathbf{L}$ and $\mathbf{U}_C(3)$ denotes the 3rd column of matrix $\mathbf{U}$.

DOI: 10.1002/sec

Table I. Notations.

Symbol	Definition
U_i	User i
ID_i	User i's identity
PW_i	User i's chosen password
K_s	The server's secret key
AK_i	The authenticated session key computed by the server and U_i
n	The number of users that could be supported by the system
$\mathbf{A}_{n\times n}$	A symmetric key matrix
T	The timestamp
$h(\cdot)$	A one-way hash function
p	A prime number and p is divisible by $q-1$
g	A generator of order q
$\oplus$	An XOR operation

Pathan *et al.* [17,18] proposed two bilateral authentication schemes based on LU decomposition. Their proposed schemes are divided into four phases: registration, login, authentication, and password-changing phases. However, their schemes have several security weaknesses, including (1) they cannot resist replay attacks; (2) passwords could be revealed by the server; (3) they cannot preserve user anonymity; and (4) the server and users cannot agree on a session key. Therefore, we propose a bilateral user-authentication scheme that not only fixes these weaknesses, but also aims to achieve more functionalities and resists well-known attacks.

3. Our Proposed Scheme

Our bilateral user authentication scheme is divided into four phases: registration, login, authentication, and password-changing phases. The notations and their corresponding definitions are listed in Table I.

3.1. Registration Phase

Suppose a new user U_i with the identity ID_i wants to register with a server for remote-access services. U_i randomly chooses his password PW_i and sends the pair $(ID_i, h(PW_i))$ to the server. Almost all existing user authentication schemes [3,7,10–13,15,17–20,22] presume the existence of a secure channel in the registration phase. This usually means the private registration data are submitted in person or through an existing secure channel. Upon receiving the registration message, the server takes the following steps:

1. Generate two random numbers x_i, y_i between 1 and n, and select the x_i-th row from matrix $\mathbf{L}$ (denoted as $\mathbf{L}_R(x_i)$), the x_i-th column from matrix $\mathbf{U}$ (denoted as $\mathbf{U}_C(x_i)$), and the y_ith column from matrix $\mathbf{U}$ (denoted as $\mathbf{U}_C(y_i)$).
2. Compute the pair $(K_{x_iy_i}, \theta_i)$ as follows: ($\oplus$ means the exclusive-or operation)

$$K_{x_iy_i} = \mathbf{L}_R(x_i) \times \mathbf{U}_C(y_i) \tag{6}$$

$$\theta_i = h(ID_i \oplus K_{x_iy_i}) \oplus h(PW_i) \oplus h(K_s) \tag{7}$$

3. Issue a smart card containing $(K_{x_iy_i}, \theta_i, \mathbf{U}_C(x_i), v_i, h(\cdot), g, p)$ to U_i, where $v_i = h(K_s) \oplus y_i$.

In the registration and password-changing phases, in order to keep a user's password secret and resist insider attacks, the user transmits his password in hashed form, rather than as plain text. Note that Pathan *et al.*'s schemes [17,18] make use of plain text for transmitting passwords. In addition, the system parameters g and p, used for computing a session key, have to be embedded in the smart card for later use.

3.2. Login Phase

When U_i wants to log in to the system, U_i first attaches the smart card and inputs his password PW_i^*. The smart card performs the following operations:

1. Generate a random number r.
2. Compute the pair (H_i, S_i) as follows:

$$H_i = K_{x_iy_i} \oplus h(r \oplus T) \tag{8}$$

$$S_i = \theta_i \oplus h(PW_i^*) \oplus r \tag{9}$$

where T is the current timestamp.
3. Generate a random number a and compute the pair (r_i, R_i) as follows:

$$r_i = g^a \bmod p. \tag{10}$$

$$R_i = h(\theta_i \oplus r_i) \tag{11}$$

4. Encrypt $(ID_i, r_i, U_C(x_i), v_i, T)$ with R_i and compute C_i as follows:

$$C_i = \theta_i \oplus h(ID_i \oplus K_{x_iy_i}) \oplus h(PW_i^*) \oplus R_i$$
$$= h(K_s) \oplus R_i \tag{12}$$

5. Send the login message $M_i = (C_i, E_{R_i}(ID_i, r_i, U_C(x_i), v_i, T), H_i, S_i, T)$ to the server.

DOI: 10.1002/sec

To achieve the requirements of key agreement and user anonymity, which are not provided in Pathan *et al.*'s schemes [17,18], the smart card has to compute the nonce r_i and encrypt the user's identity and other parameters as Equations (10) and (12), respectively. Moreover, since the user has already bound the timestamp T into the login message according to Equation (8), rather than only transmitting the timestamp in the login message, the proposed scheme can resist a replay attack. Note that Pathan *et al.*'s schemes [17,18] only transmit the timestamp in the login message without bounding it into the login message.

3.3. Authentication Phase

Upon receiving the login request M_i, the server performs the following operations:

1. Compute $R_i = C_i \oplus h(K_s)$, and decrypt $E_{R_i}(ID_i, r_i, \mathbf{U}_C(x_i), v_i, T)$ with R_i.
2. Check the validity of ID_i. If ID_i is invalid, the server rejects the login request.
3. Verify if the time interval $(T' - T) \leq \Delta T$, where T' is the current timestamp and ΔT is the allowed time interval for transmission delay. If $(T' - T) > \Delta T$, the login request is considered out-of-date and is rejected.
4. Compute $(v_i \oplus h(K_s))$, which is denoted as y_i.
5. Compute the triple $(K_{y_i x_i}, t, r')$ as follows:

$$K_{y_i x_i} = \mathbf{L}_R(y_i) \times \mathbf{U}_C(x_i) \tag{13}$$

$$t = h(ID_i \oplus K_{y_i x_i}) \tag{14}$$

$$r' = S_i \oplus T \oplus h(K_s) \oplus t \tag{15}$$

6. Verify if the following equation holds:

$$K_{x_i y_i} = H_i \oplus h(r') \tag{16}$$

   If not, the server rejects the login request. Otherwise, it proceeds to the next step.
7. Generate a random number b and compute r_s as follows:

$$r_s = g^b \bmod p. \tag{17}$$

8. Construct the authenticated session key AK_i:

$$AK_i = {r_i}^b = g^{ab} \bmod p. \tag{18}$$

9. Send

$$E_{R_i}(K_{y_i x_i} \oplus r_s, r_i + 1, T'')U_i.$$

After receiving the message $E_{R_i}(K_{y_i x_i} \oplus r_s, r_i + 1, T'')$, U_i performs following operations:

1. Decrypt the message, obtain $K_{y_i x_i} \oplus r_s$, and verify whether $(T''' - T'') \leq \Delta T$, where T''' is the current timestamp. If so, U_i proceeds to the next step.
2. Check whether decrypted data contains the value $r_i + 1$. If so, U_i uses $K_{x_i y_i}$ to compute r_s as follows:

$$r_s = (K_{y_i x_i} \oplus r_s) \oplus K_{x_i y_i} \tag{19}$$

3. Generate the authenticated session key AK_i as follows:

$$AK_i = {r_s}^a = g^{ba} = g^{ab} \bmod p. \tag{20}$$

Then U_i uses AK_i to communicate with the server.

In this authentication phase, the server has to generate a nonce r_s and compute a session key AK_i according to Equations (17) and (18), respectively. Furthermore, the user also needs to compute the session key as Equation (20). The session key computation mentioned above does not appear in Pathan *et al.*'s schemes [17,18] because their schemes did not provide key agreement.

3.4. Password-Changing Phase

When U_i wants to change his password PW_i to PW_i', he sends the triple $(ID_i, h(PW_i), h(PW_i'))$ to the server. As in the registration phase, these private data should be submitted in person or *via* a secure channel. Upon receiving the password-changing message, the server takes the following steps:

1. Compute θ_i' as follows:

$$\begin{aligned}\theta_i' &= \theta_i \oplus h(PW_i) \oplus h(PW_i') \\ &= h(ID_i \oplus K_{x_i y_i}) \oplus h(PW_i') \oplus h(K_s)\end{aligned} \tag{21}$$

2. Replace θ_i with θ_i' in the smart card.

As in the registration phase, the user has to transmit his password in hashed form in this phase to keep his password secret and withstand insider attacks.

DOI: 10.1002/sec

4. Analysis of Our Scheme

In this section, we analyze our scheme and show that our scheme can resist several well-known attacks. In addition, we provide a comparative study with other authentication schemes.

4.1. Correctness

According to Equation (15), we first derive the equation as follows:

$$\begin{aligned} r' &= S_i \oplus T \oplus h(K_s) \oplus t \\ &= \theta_i \oplus h(PW_i^*) \oplus r \oplus T \oplus h(K_s) \oplus t \\ &= h(ID_i \oplus K_{x_i y_i}) \oplus h(PW_i) \oplus h(K_s) \oplus h(PW_i^*) \\ &\quad \oplus r \oplus T \oplus h(K_s) \oplus t \\ &= h(ID_i \oplus K_{x_i y_i}) \oplus r \oplus T \oplus t \\ &= h(ID_i \oplus K_{x_i y_i}) \oplus r \oplus T \oplus h(ID_i \oplus K_{y_i x_i}) \\ &= r \oplus T \end{aligned} \tag{22}$$

Since the proposed scheme employs **LU** decomposition, $K_{x_i y_i} = K_{y_i x_i}$. That is, $h(ID_i \oplus K_{x_i y_i}) \oplus h(ID_i \oplus K_{y_i x_i}) = 0$. Therefore, $r' = r \oplus T$.

Using Equation (22), we verify Equation (16) as follows:

$$\begin{aligned} K_{x_i y_i} &= H_i \oplus h(r') \\ &= K_{x_i y_i} \oplus h(r \oplus T) \oplus h(r \oplus T) \\ &= K_{x_i y_i} \end{aligned} \tag{23}$$

4.2. Security Analysis

We now analyze the security properties of our scheme. We first introduce a few terms used in this paper [28].

Definition 1. *The discrete logarithm problem (DLP) is defined as follows: given a prime p, a generator g of Z_p^*, and an element $\beta \in Z_p^*$, find the integer α, $0 \leq \alpha \leq p-2$, such that $g^\alpha \equiv \beta \pmod p$.*

Definition 2. *The Diffie–Hellman problem (DHP) is defined as follows: given a prime p, a generator g of Z_p^*, and elements $g^c \pmod p$ and $g^s \pmod p$, find $g^{cs} \pmod p$.*

The security of the proposed scheme is based on the difficulty of DLP and DHP, which are believed infeasible to solve in polynomial time. We will show that our scheme can resist replay attack, forgery attack, insider attack, reflection attack, and parallel session attack. We will also analyze the following security properties: anonymity, mutual authentication, forward secrecy, and known-key security.

Theorem 1. *The proposed scheme can resist a replay attack.*

Proof. Assume an adversary eavesdrops the login message sent by U_i and uses it to impersonate U_i when logging into the system in a later session. However, the replay of U_i's previous login message will be detected by the server since the user has already bound the timestamp T into the login message according to Equation (8), and the server will verify the validity of the timestamp T used by U_i. Therefore, the adversary cannot replay the login message. However, there seems to be one potential security threat common to most existing timestamp-based user authentication schemes. That is, an adversary could impersonate a legitimate user by replaying that user's previous login message within the allowed time interval ΔT. This threat can be solved by the additional requirement that T is not reused by U_i within ΔT.

Theorem 2. *The proposed scheme can resist a forgery attack.*

Proof. If the adversary wants to impersonate U_i, he has to create a valid login message $(C_i^*, E_{R_i^*}(ID_i, r_i^*, U_C(x_i), v_i, T^*), H_i^*, S_i^*, T^*)$, where T^* is the current timestamp. First he has to choose a random number r^* and compute the pair (H_i^*, S_i^*)as follows:

$$H_i^* = K_{x_i y_i} \oplus h(r^* \oplus T^*) \tag{24}$$

$$S_i^* = \theta_i \oplus h(PW_i) \oplus r^* \tag{25}$$

Because having no idea about $K_{x_i y_i}$, θ_i, and PW_i, the adversary cannot forge a valid login message and hence cannot launch a forgery attack.

Theorem 3. *The proposed scheme can resist an insider attack.*

Proof. In our proposed scheme, when U_i wants to resigter with a server for remote-access services, he has to submit $(ID_i, h(PW_i))$ instead of (ID_i, PW_i), as in Pathan *et al.*'s schemes [17,18]. Due to the employment of the one-way hash function h, it is considered practically impossible for the server to derive the user's password PW_i from the hashed value

DOI: 10.1002/sec

[29]. That is, even the server does not know PW_i. Obviously, the proposed scheme can prevent the insider attack.

Theorem 4. *The proposed scheme can resist a reflection attack.*

Proof. A reflection attack is one in which, when a user sends a login message to a server, the adversary eavesdrops the message and sends it (or a modified version of the message) back to the user. In the proposed scheme, the adversary cannot fool the server since he has to know the server's secret key K_s in computing R_i, which is used to decrypt the ciphertext $E_{R_i}(ID_i, r_i, \mathbf{U}_C(x_i), v_i, T)$ sent by U_i. Therefore, it is ensured that our scheme can withstand the reflect attack.

Theorem 5. *The proposed scheme can resist a parallel-session attack.*

Proof. In the proposed scheme, an adversary cannot impersonate a legitimate user by creating a valid login message in another on-going run from the honest run since the server's response message $E_{R_i}(K_{y_i x_i} \oplus r_s, r_i + 1, T'')$ is encrypted with R_i, which is unknown to the adversary. Therefore, the proposed scheme can resist the parallel-session attack.

Theorem 6. *The proposed scheme can provide user anonymity.*

Proof. If an adversary eavesdrops the login message, he cannot extract the user's identity from the ciphertext $E_{R_i}(ID_i, r_i, \mathbf{U}_C(x_i), v_i, T)$ since it is encrypted with R_i, which is unknown to the adversary. In addition, due to the use of the nonce and the timestamp in the login phase, the login messages submitted to the server are different in the login sessions. Hence, it is difficult for the adversary to discover a user's identity. Clearly, the proposed scheme can provide user anonymity.

Theorem 7. *The proposed scheme can provide mutual authentication.*

Proof. The proposed scheme uses the Diffie–Hellman key exchange algorithm to achieve mutual authentication between the server and a user. U_i and the server securely exchange r_i and r_s in the login and authentication phases, respectively. As a result, the authenticated session key is established as follows:

$$AK_i = r_i^b = r_s^a = g^{ab} \bmod p \tag{26}$$

Therefore, U_i and the server can use the authenticated session key AK_i in subsequent communications.

Theorem 8. *The proposed scheme can provide perfect forward secrecy.*

Proof. Perfect forward secrecy means that the disclosure of the long-term secret key material (e.g., server's secret key K_s and user's password PW_i) does not compromise the secrecy of the agreed keys in earlier runs. In the proposed scheme, perfect forward secrecy is ensured since the Diffie–Hellman key exchange algorithm is used to establish the authenticated session key g^{ab}. Even if the adversary knows the server's secret key K_s, he is only able to obtain g^a and g^b from earlier runs. However, based on the difficulty of the discrete logarithm problem and the Diffie–Hellman problem, it is computationally infeasible to compute the authenticated session key g^{ab} from g^a and g^b. Thus, our proposed scheme provides perfect forward secrecy.

Theorem 9. *The proposed scheme can provide known-key security.*

Proof. Known-key security means that the compromise of a session key will not lead to further compromise of other secret keys or session keys. Even if a session key g^{ab} is revealed to an adversary, he still cannot derive other session keys since they are generated from the random numbers $g^{a'}$ and $g^{b'}$ based on Diffie–Hellman key exchange algorithm. Hence, the proposed scheme can achieve known-key security.

4.3. Functionality

We summarize the functionality of our proposed scheme in this subsection. The crucial criteria in a user authentication scheme are listed below:

- **C1.** *Freely chosen password*: a user can choose his password freely in the registration phase.
- **C2.** *Mutual authentication*: the server and a user can authenticate each other.
- **C3.** *User anonymity*: a user's identity is protected when he logs into the system. No one knows the user's identity except the server.
- **C4.** *Session key agreement*: while mutual authentication is established between the server and a user, they can agree on a session key for use in subsequent communications.
- **C5.** *Secure password change*: after the registration, a user can change his password freely.

DOI: 10.1002/sec

Table II. Comparison of authentication schemes.

	C1	C2	C3	C4	C5
Our scheme	Yes	Yes	Yes	Yes	Yes
Pathan *et al.* [18]	Yes	Yes	No	No	No
Hu *et al.* [8]	Yes	Yes	Yes	Yes	Yes
Pathan and Hong [17]	Yes	Yes	No	No	Yes*
Chien and Chen [2]	Yes	Yes*	Yes	Yes	No
Das *et al.* [5]	Yes	No	Yes*	No	No
Juang [9]	Yes	Yes	No	Yes	No
Chien *et al.* [4]	Yes	Yes	No	No	No

C1: freely chosen password; C2: mutual authentication; C3: user anonymity; C4: session key agreement; C5: secure password change.
*Authors claimed such a security property but the property actually failed.

Table III. Evaluation parameters.

Symbol	Definition
T_{H}	Time for performing a one-way hash function
T_{M}	Time for performing a vector multiplication operation
T_{XOR}	Time for performing an XOR operation
T_{EXP}	Time for performing an exponentiation operation
T_{ENC}	Time for performing a symmetric encryption operation
T_{DEC}	Time for performing a symmetric decryption operation

We summarized the functionality of related authentication and key distribution protocols in Table II.

4.4. Efficiency Analysis

Now we examine the performance of our proposed scheme. The evaluation parameters are defined in Table III. The time requirement of the proposed scheme is summarized in Table IV. We use the computational overhead as the metrics to evaluate the performance of the proposed scheme. In our scheme, only one hashing operation is required for a user to register and get his smart card. In the login phase, three hashing operations, nine exclusive-or operations, one exponentiation operation, and one symmetric encryption operation are needed for a user. For authentication, two exclusive-or operations, one symmetric decryption operation, and one exponentiation operation are needed for a user. We can see from Table IV that the exponentiation operations are required by the server and the user due to the requirements of key agreement and perfect forward secrecy. These operations might be expensive for smart cards nowadays. However, with an increasing demand for information security as today's security systems still have plenty of room for improvement, it is expected that the complicated computations will be widely adopted as a necessary security measure and hardware security enhancement for smart cards will become prevalent in the near future.

5. Conclusions

In this paper, we present a bilateral user authentication scheme based on **LU** decomposition. The scheme can withstand well-known attacks and possesses many merits, including freely changeable passwords, mutual authentication, user anonymity, and session key agreement. In addition, the proposed scheme is secure against replay attacks, forgery attacks, insider attacks, reflection attacks, and parallel session attacks. Moreover, compared with other authentication schemes, our scheme achieves more functionalities.

Acknowledgement

This work was supported by the National Science Council, Taiwan, Republic of China, under grant NSC

Table IV. Performance of the proposed scheme.

Phase	The server	A user
Registration	$1T_{\mathrm{M}} + 2T_{\mathrm{H}} + 4T_{\mathrm{XOR}}$	$1T_{\mathrm{H}}$
Login	—	$3T_{\mathrm{H}} + 9T_{\mathrm{XOR}} + 1T_{\mathrm{EXP}} + 1T_{\mathrm{ENC}}$
Authentication	$1T_{\mathrm{M}} + 2T_{\mathrm{H}} + 8T_{\mathrm{XOR}} + 2T_{\mathrm{EXP}} + 1T_{\mathrm{ENC}} + 1T_{\mathrm{DEC}}$	$2T_{\mathrm{XOR}} + 1T_{\mathrm{DEC}} + 1T_{\mathrm{EXP}}$
Total	$2T_{\mathrm{M}} + 4T_{\mathrm{H}} + 12T_{\mathrm{XOR}} + 2T_{\mathrm{EXP}} + 1T_{\mathrm{ENC}} + 1T_{\mathrm{DEC}}$	$4T_{\mathrm{H}} + 11T_{\mathrm{XOR}} + 2T_{\mathrm{EXP}} + 1T_{\mathrm{ENC}} + 1T_{\mathrm{DEC}}$

DOI: 10.1002/sec

96-2752-E-009-005-PAE, NSC 96-2219-E-009-012, NSC 96-2219-E-009-006, NSC 96-2219-E-009-008, and NSC-96-3114-P-001-002-Y.

References

1. Lamport L. Password authentication with insecure communication. *Communications of the ACM* 1981; **24**(11): 770–772.
2. Chien HY, Chen CC. A remote authentication scheme preserving user anonymity. In *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA'05)*, March 2005, pp. 245–248.
3. Chang YF, Chang CC, Su YW. A secure improvement on the user-friendly remote authentication scheme with no time concurrency mechanism. In *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA'06)*, Vol. 2, April 2006.
4. Chien HY, Jan JK, Tseng YM. An efficient and practical solution to remote authentication: smart card. *Computers and Security* 2002; **21**(4): 372–375.
5. Das ML, Saxena A, Gulati VP. A dynamic ID-based remote user authentication scheme. *IEEE Transactions on Consumer Electronics* 2004; **50**(2): 629–631.
6. Hsu CL. Security of Chien et al.'s remote user authentication scheme using smart cards. *Computer Standards and Interfaces* 2004; **26**(3): 167–169.
7. Hsu CL. A user friendly remote authentication scheme with smart cards against impersonation attacks. *Applied Mathematics and Computation* 2005; **170**(1): 135–143.
8. Hu L, Yang Y, Niu X. Improved remote user authenticatoin scheme preserving user anonymity. In *Proceedings of the IEEE International Conference on Communication Networks and Services Research (CNSR'07)*, May 2007, pp. 323–328.
9. Juang WS. Efficient password authenticated key agreement using smart cards. *Computers and Security* 2004; **23**(2): 167–173.
10. Ku WC, Chuang HM, Tsaur MJ. Vulnerabilities of Wu-Chieu's improved password authentication scheme using smart cards. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 2005; **E88-A**(11): 3241–3243.
11. Ku WC, Chang ST, Chen HH, Tsaur MJ. Weakness and simple improvement of a password authentication scheme based on geometric approach. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN'05)*, November 2005, pp. 472–473.
12. Kim KW, Jeon JC, Yoo KY. Efficient and secure password authentication schemes for low-power devices. In *Proceedings of International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2005)*, December 2005, pp. 73–82.
13. Lee SW, Kim HS, Yoo KY. Improvement of Chien et al.'s remote user authentication scheme using smart cards. *Computer Standards and Interfaces* 2005; **27**(2): 181–183.
14. Liaw HT, Lin JF, Wu WC. An efficient and complete remote user authentication scheme using smart card. *Mathematical and Computer Modelling* 2006; **44**(1–2): 223–228.
15. Lee Y, Nam J, Kim S, Won D. Two efficient and secure authentication schemes using smart cards. In *Proceedings of International Conference on Computational Science and its Applications (ICCSA 2006)*, May 2006, pp. 858–866.
16. Mitchell CJ, Tang Q. Security of the Lin–Lai smart card based user authentication scheme. *Technical Report* RHUL-MA-2005-1, Royal Holloway, University of London, January 2005.
17. Pathan AK, Hong CS. An efficient bilateral remote user authentication scheme with smart cards. In *Proceedings of the 33rd Korea Information Science Society Fall Conference*, October 2006, pp. 132–134.
18. Pathan AK, Hong CS, Suda T. A novel and efficient bilateral remote user authentication scheme using smart cards. In *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE'07)*, January 2007, pp. 1–2.
19. Sun HM. An efficient remote use authenticaiton scheme using smart cards. *IEEE Transactions on Consumer Electronics* 2000; **46**(4): 958–961.
20. Wu ST, Chieu BC. A user friendly remote authentication scheme with smart cards. *Computers and Security* 2003; **22**(6): 547–550.
21. Yoon EJ, Yoo KY. Robust secret key based authentication scheme using smart cards. In *Proceedings of Pacific Rim Conference on Multimedia (PCM 2005)*, November 2005, pp. 723–734.
22. Yoon EJ, Yoo KY. New authentication scheme based on a one-way hash function and Diffie–Hellman key exchange. In *Proceedings of International Conference on Cryptology and Network Security (CANS 2005)*, December 2005, pp. 147–160.
23. Choi SJ, Youn HY. A novel data encryption and distribution approach for high security and availability using LU decomposition. In *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA'04)*, May 2004, pp. 637–646.
24. Hsiao HI, DeWitt DJ. A performance study of three high availability data replication strategies. In *Proceedings of the First International Conference on Parallel and Distributed Information Systems (ICPDIS)*, December 1991, pp. 18–28.
25. Long DDE. A technique for managing mirrored disks. In *Proceedings of the IEEE International Conference on Performance, Computing, and Communications*, April 2001, pp. 272–277.
26. Menon J, Riegel J, Wyllie J. Algorithms for software and low-cost hardware RAIDs. In *Proceedings of the 40th IEEE Computer Society International Conference (COMPCON)*, March 1995, pp. 411–418.
27. Zarowski CJ. *An Introduction to Numerical Analysis for Electrical And Computer Engineers*. John Wiley & Sons, Inc.: Hoboken, NJ, 2004; 148.
28. Menezes AJ, Oorschot PC, Vanstone SA. *Handbook of Applied Cryptography*. CRC Press Boca Raton, Florida, 1997.
29. Schneier B. *Applied Cryptography* (2nd edn). John Wiley & Sons Inc. Publication: New York, 1996.

DOI: 10.1002/sec