Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 910409, 10 pages
http://dx.doi.org/10.1155/2013/910409

Research Article

Applying LU Decomposition of Matrices to Design Anonymity Bilateral Remote User Authentication Scheme

Xiong Li,[1,2] Jianwei Niu,[2] Muhammad Khurram Khan,[3] and Zhibo Wang[4]

[1] *School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China*
[2] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*
[3] *Center of Excellence in Information Assurance, King Saud University, Riyadh 11653, Saudi Arabia*
[4] *College of Software, East China Institute of Technology, Nanchang 330013, China*

Correspondence should be addressed to Jianwei Niu; niujianwei@buaa.edu.cn

Received 10 October 2012; Revised 9 January 2013; Accepted 9 January 2013

Academic Editor: Oleg V. Gendelman

We apply LU decomposition of matrices to present an anonymous bilateral authentication scheme. This paper aims at improving security and providing more excellent performances for remote user authentication scheme. The proposed scheme can provide bilateral authentication and session key agreement, can quickly check the validity of the input password, and can really protect the user anonymity. The security of the proposed scheme is based on the discrete logarithm problem (DLP), Diffie-Hellman problem (DHP), and the one-way hash function. It can resist various attacks such as insider attack, impersonation attack, server spoofing attack, and stolen smart card attack. Moreover, the presented scheme is computationally efficient for real-life implementation.

1. Introduction

The remote user authentication scheme allows the user and the remote server to mutually authenticate each other over public network environments, and then the authorized user can access the services and resources which are provided by the remote server. Generally, the password-based authentication scheme provides an efficient and secure way for mutual authentication and allows the user and the server to establish a shared session key for future secret communication after the mutual authentication process. In 1981, Lamport [1] first proposed a password-based remote user authentication scheme for the insecure communication. Since then, the researchers have proposed many password-based remote user authentication schemes [2–7] to ensure the secure communication through the public network, and also many studies [8–18] have been presented to enhance the security or improve the computation and communication costs of the remote user authentication scheme.

In the public network environments, it is important to ensure user anonymity such that the user's real identity can only be revealed by authorized entities. In 2000, Lee and Chang [19] proposed a user identification scheme with key distribution preserving user anonymity for the distributed computer network. However, Wu and Hsu [20] pointed out that Lee and Chang's scheme cannot protect user anonymity as they claimed, and they proposed an enhanced scheme. Later, Yang et al. [21] showed that Wu and Hsu's scheme cannot resist impersonation attack and proposed an improved scheme which is more secure and efficient. Unfortunately, Mangipudi and Katti [22] presented that Yang et al.'s protocol is vulnerable to a Denial-of-Service (DoS) attack and proposed a secure identification and key agreement protocol with user anonymity. Recently, Wang et al. [23] presented a secure and efficient identification and key agreement protocol with user anonymity based on the difficulty of computing the elliptic curve Diffie-Hellman. Their scheme's computation cost is lower and is suitable for applications in low power computing environments.

In 2004, Choi and Youn [24] proposed a novel data encryption and distribution approach using LU decomposition of matrices. Then, Pathan et al. [25, 26] proposed two efficient bilateral remote user authentication schemes based on LU decomposition of matrices. Nevertheless, these schemes have several weaknesses, such as they cannot resist replay attacks, they cannot preserve the user anonymity, the server and users cannot agree on a session key, and so forth. To address these issues, Tseng et al. [27] proposed a user authentication scheme based on LU decomposition of matrices. They claimed that their scheme can resist replay attack, forgery attack, and insider attack and provide user anonymity. Whereas, after careful analysis, we find that Tseng et al.'s scheme is still vulnerable to insider attack, stolen smart card attack and inefficient for wrong password login and does not provide user anonymity. To overcome these existed weaknesses of Tseng et al.'s scheme, we propose a novel bilateral authentication scheme with user anonymity using LU decomposition of matrices. Analysis shows that our scheme not only can provide better security properties but also is more efficient than the other authentication schemes.

The rest of this paper is organized as follows: Section 2 introduces the necessary preliminaries of this paper. The brief review of Tseng et al.'s scheme is provided in Section 3. Section 4 describes a cryptanalysis of Tseng et al.'s scheme. The proposed scheme and the corresponding analysis are presented in Sections 5 and 6, respectively. Finally, we conclude this paper in Section 7.

2. Preliminaries

In this section, we introduce some basic information about the LU decomposition of matrices and Discrete logarithm problem, and they are the mathematical basis of our proposed bilateral remote user authentication protocol with user anonymity.

2.1. LU Decomposition of Matrices. From the matrix theory, LU decomposition factorizes a matrix as the product of a lower triangular matrix and an upper triangular matrix. Let A be a square matrix; an LU decomposition of matrix A is the form $A = LU$, where L is a lower triangular matrix and U is an upper triangular matrix. This means that L has only zeros above the diagonal and U has only zeros below the diagonal. For example, for a 4×4 matrix A, its LU decomposition looks like

$$\begin{aligned} A &= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \\ &= \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}. \end{aligned} \tag{1}$$

If A is a singular matrix of rank k, it admits an LU decomposition if all the k-leading principal minors are nonzero.

In the identity authentication system, we assume that n is the number of users the system can support. We can introduce the LU decomposition into the user authentication system to ensure the security of the system. In the system initialization phase, the remote server generates a symmetric matrix $A_{n\times n}$ as his/her master secret key. With the LU decomposition, the server can separate the symmetric key matrix $A_{n\times n}$ to the product of a lower triangular matrix $L_{n\times n}$ and an upper triangular matrix $U_{n\times n}$, that is, $A_{n\times n} = L_{n\times n}U_{n\times n}$, and stores these matrices in other servers.

Since A is a symmetric matrix, we have that $a_{ij} = a_{ji}$, for $1 \le i \le n$ and $1 \le j \le n$, and the product of the xth row of matrix $L_{n\times n}$ and the yth column of matrix $U_{n\times n}$ is equal to the product of the yth row of matrix $L_{n\times n}$ and the xth column of matrix $U_{n\times n}$. For example, suppose A is a 4×4 symmetric matrix with LU decomposition as follows:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 8 & 7 \\ 3 & 8 & 11 & 9 \\ 4 & 7 & 9 & 6 \end{pmatrix}. \tag{2}$$

We can perform elementary row operations to get the lower matrix L and upper matrix U as follows:

$$\begin{aligned} L &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & -1 & \dfrac{1}{2} & 1 \end{pmatrix}, \\ U &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & -2 & -1 \\ 0 & 0 & 0 & -\dfrac{21}{2} \end{pmatrix}. \end{aligned} \tag{3}$$

Given $i = 3$ and $j = 4$, we can compute a_{34} and a_{43} as follows:

$$\begin{aligned} a_{34} &= L_R(3) \times U_C(4) \\ &= \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 4 & -1 & -1 & -\dfrac{21}{2} \end{pmatrix}^T = 9, \\ a_{43} &= L_R(4) \times U_C(3) \\ &= \begin{pmatrix} 4 & -1 & \dfrac{1}{2} & 1 \end{pmatrix} \times \begin{pmatrix} 3 & 2 & -2 & 0 \end{pmatrix}^T = 9, \end{aligned} \tag{4}$$

where $L_R(3)$ denotes the 3rd row of the matrix L and $U_C(4)$ denotes the 4th column of the matrix U, and we have that $a_{34} = a_{43}$.

2.2. Discrete Logarithm Problem. The detailed information about discrete logarithm problem can be found in the literature [28], and we briefly introduce the discrete logarithm problem as follow. In a multiplicative group $\langle g\rangle$ of order q, where $p = 2q + 1$ is the modulus for the group, both p and q are public large prime numbers. This implies

(1) $\langle g\rangle = \{X \mid X = g^x \bmod p, \text{ for } x = 1, 2, 3\}$ is a finite set of size q, where $2 \le g \le p - 1$ and $g^q \bmod p = 1$.

(2) Given x and g, computing the modular exponentiation $X = g^x \bmod p$ is relatively easy. However, given X and g, it is computationally infeasible to find x such that $X = g^x \bmod p$; namely, in $\langle g\rangle$, the discrete logarithm problem is intractable [28].

(3) Moreover, given g, p, $X = g^x \bmod p$ and $Y = g^y \bmod p$, computing $K = g^{xy} \bmod p$, which is known as the Diffie-Hellman problem, is also intractable [28].

3. Review of Tseng et al.'s Scheme

In this section, we briefly review Tseng et al.'s scheme, and more details can be found in [27]. Tseng et al.'s scheme contains four phases, that is, the registration phase, the login phase, the authentication phase, and the password change phase. The notations used throughout this paper are listed in Table 1.

Table 1: Notations used in this paper.

S	The server
U_i	User i
ID_i	The identity of user U_i
PW_i	The password of user U_i
K_S	The server's secret key
AK_i	The authenticated session key shared between the server and U_i
n	The number of users that could be supported by the system
A	A secret $n \times n$ symmetric matrix with LU decomposition generated by the server
L, U	Matrix A's LU decomposition such that $A = LU$
T	The timestamp
ΔT	The expected time interval for transmission delay
$h(\cdot)$	One-way hash function
p, q	Two big prime numbers generated according to the requirement as in Section 2.2
g	A generator of order q

3.1. Registration Phase. Suppose a new user U_i with the identity ID_i wants to register himself/herself with the server for access the remote services. U_i randomly chooses his/her password PW_i and sends ID_i, $h(\mathrm{PW}_i)$ to the server through a secure channel. Upon receiving the registration request message, the server takes the following steps:

(1) generates two random numbers x_i, y_i between 1 and n. Then selects the x_ith row from the matrix L (denoted as $L_R(x_i)$), the x_ith column from matrix U (denoted as $U_C(x_i)$), and the y_ith column from the matrix U (denoted as $U_C(y_i)$);

(2) computes $(K_{x_iy_i}, \theta_i, v_i)$ as follows:

$$K_{x_iy_i} = L_R(x_i) \times U_C(y_i),$$
$$\theta_i = h(\mathrm{ID}_i \oplus K_{x_iy_i}) \oplus h(\mathrm{PW}_i) \oplus h(K_S),$$
$$\text{and } v_i = h(K_S) \oplus y_i;$$

(3) stores $(K_{x_iy_i}, \theta_i, U_C(x_i), v_i, h(\cdot), g, p)$ into a smart card and submits the smart card to U_i via a secure channel.

3.2. Login Phase. When U_i wants to login into the system, U_i first inserts the smart card to the card reader and inputs his/her password PW_i^*. The smart card performs the following steps to generate the login request message:

(1) Generates a random number r;

(2) computes $H_i = K_{x_iy_i} \oplus h(r \oplus T)$ and $S_i = \theta_i \oplus h(\mathrm{PW}_i^*) \oplus r$, where T is the current timestamp;

(3) generates a random number a and then computes $r_i = g^a \bmod p$ and $R_i = h(\theta_i \oplus r_i)$;

(4) encrypts $(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ with R_i and computes $C_i = \theta_i \oplus h(\mathrm{ID}_i \oplus K_{x_iy_i}) \oplus h(\mathrm{PW}_i^*) \oplus R_i = h(K_S) \oplus R_i$;

(5) sends the login request message $M_i = (C_i, E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T), H_i, S_i, T)$ to the server.

3.3. Authentication Phase. Upon receiving the login request message M_i, the server and the user U_i perform the following operations for mutual authentication.

(1) The server computes $R_i = C_i \oplus h(K_S)$ and decrypts $E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ with R_i.

(2) The server checks the validity of ID_i. If ID_i is invalid, the server rejects the login request.

(3) The server verifies whether the time interval $(T' - T) \leq \Delta T$, where T' is the current timestamp when the server received the message. If $(T' - T) \geq \Delta T$, the login request is considered out of date and is rejected.

(4) The server computes $y_i = v_i \oplus h(K_S)$.

(5) The server computes $K_{y_ix_i} = L_R(y_i) \times U_C(x_i)$, $t = h(\mathrm{ID}_i \oplus K_{y_ix_i})$, and $r' = S_i \oplus T \oplus h(K_S) \oplus t$.

(6) The server verifies whether $K_{x_iy_i}$ equals $H_i \oplus h(r')$. If not, the server rejects the login request. Otherwise, it proceeds to the next steps.

(7) The server generates a random number b and computes $r_s = g^b \bmod p$.

(8) The server computes the authenticated session key $\mathrm{AK}_i = r_i^b = g^{ab} \bmod p$.

(9) At last, the server sends $E_{R_i}(K_{y_ix_i} \oplus r_s, r_i + 1, T'')$ to U_i, where T'' is the current timestamp.

(10) When receiving the message $E_{R_i}(K_{y_ix_i} \oplus r_s, r_i + 1, T'')$, U_i decrypts the message, gets $K_{y_ix_i} \oplus r_s$, and verifies whether $(T''' - T'') \leq \Delta T$, where T''' is the current timestamp. If so, U_i proceeds to the next steps.

(11) U_i checks whether decrypted data contain the value $r_i + 1$. If so, U_i uses $K_{y_ix_i}$ to compute $r_s = (K_{y_ix_i} \oplus r_s) \oplus K_{y_ix_i}$.

(12) U_i generates the authenticated session key AK_i as $\mathrm{AK}_i = r_s^a = (g^b)^a = g^{ab} \bmod p$. Then U_i can communicate with the server secretly by using AK_i.

3.4. Password Change Phase. When U_i wants to change his password PW_i to PW'_i, he sends $(\mathrm{ID}_i, h(\mathrm{PW}_i),\ h(\mathrm{PW}'_i))$ to the server. Upon receiving the password-changing message, the server takes the following steps:

(1) computes $\theta'_i = \theta_i \oplus h(\mathrm{PW}_i) \oplus h(\mathrm{PW}'_i) = h(\mathrm{ID}_i \oplus K_{x_i y_i}) \oplus h(\mathrm{PW}'_i) \oplus h(K_S)$;

(2) replaces θ_i with θ'_i in the smart card.

4. Cryptanalysis of Tseng et al.'s Scheme

Tseng et al. claimed that their scheme can protect user anonymity and can resist various known attacks. However, after careful analysis, we find that their scheme cannot really protect the user anonymity and is vulnerable to insider attack, server spoofing attack. Besides, their scheme is inefficient for wrong password login. We analyze the security weaknesses of Tseng et al.'s scheme as below.

4.1. Attacks against the User Anonymity. In order to prevent the attacker from tracking the user's movements, it is important to ensure user anonymity such that the user's real identity can only be recognized by the server.

Kocher et al. [29] and Messerges et al. [30] have pointed out that the confidential information stored on the smart card can be extracted by physically monitoring its power consumption. So, in Tseng et al.'s scheme, a legal but malicious user U_A can extract information $(K_{x_A y_A}, \theta_A, U_C(x_A), v_A, h(\cdot), p, g)$ from his/her own smart card, and with his/her own identity ID_A and password PW_A, he/she can compute the value of $h(K_S) = \theta_A \oplus h(\mathrm{ID}_A \oplus K_{x_A y_A}) \oplus h(\mathrm{PW}_A)$. When the valid login request message $M_i = (C_i, E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T), H_i, S_i, T)$ of a legal user U_i was to be intercepted by this malicious user U_A from the public communication channel, the malicious user U_A can compute $R_i = C_i \oplus h(K_S)$, and then he/she can decrypts $E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ using R_i to obtain $(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ of the user U_i. Obviously, the malicious user U_A can obtain the real identity ID_i of the user U_i. From the above discussion, we can see that Tseng et al.'s scheme cannot really protect user anonymity.

4.2. Insider Attack. In the registration phase of Tseng et al.'s scheme, the user U_i sends $\mathrm{ID}_i, h(\mathrm{PW}_i)$ to the server for registration, and these information can be acquired by the privileged insider. However, in password change phase, the server simply not checks the validity of user U_i's ID_i and PW_i. So, this privileged insider of the remote system can masquerade as the user U_i to send the triple $(\mathrm{ID}_i, h(\mathrm{PW}_i), h(\mathrm{PW}'_i))$ to the server to perform the password-changing phase. Upon receiving the password-changing message, the server takes the following steps:

(1) computes $\theta'_i = \theta_i \oplus h(\mathrm{PW}_i) \oplus h(\mathrm{PW}'_i) = h(\mathrm{ID}_i \oplus K_{x_i y_i}) \oplus h(\mathrm{PW}'_i) \oplus h(K_S)$;

(2) replaces θ_i with θ'_i in the smart card.

Therefore, since the server does not check the validity of the user's identity and password when the user wants to change his/her password, Tseng et al.'s scheme is vulnerable to insider attack, and the privileged insider can easily change the legal user's password.

4.3. Stolen Smart Card Attack. Stolen smart card attack is that if the user's smart card is lost or stolen, the attacker can extract the information stored in the smart card and can easily change the password of the smart card, can guess the password of the user by using password guessing attacks, or can impersonate the user to login to the system.

In Tseng et al.'s scheme, a legal but malicious user U_A having his own smart card can gather information $(K_{x_A y_A}, \theta_A, U_C(x_A), v_A, h(\cdot), p, g)$ from his own smart card, and he/she can get the value of $h(K_S)$ as shown in Section 4.1. Now the malicious user U_A can intercept a valid login request message $M_i = (C_i, E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T), H_i, S_i, T)$ of the legal user U_i from the public communication channel. Then the malicious user U_A can compute $R_i = C_i \oplus h(K_S)$ and can get the identity ID_i of user U_i by decrypt $E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ using R_i. In case the user U_i's smart card is stolen by this malicious user U_A, he/she can extract the information $(K_{x_i y_i}, \theta_i, U_C(x_i), v_i, h(\cdot), g, p)$ from the memory of the smart card. With the information ID_i, $h(K_S)$, $K_{x_i y_i}$, and θ_i, the malicious user U_A can guess U_i's password by the following processes.

(1) The attacker computes $h(\mathrm{PW}_i) = \theta_i \oplus h(\mathrm{ID}_i \oplus K_{x_i y_i}) \oplus h(K_S)$.

(2) The attacker chooses a password PW'_i from a uniformly scattered dictionary.

(3) The attacker computes $h(\mathrm{PW}'_i)$ and verifies the correctness of password PW'_i by checking whether $h(\mathrm{PW}'_i)$ equals $h(\mathrm{PW}_i)$ or not.

(4) The attacker repeats steps (2) and (3) until $h(\mathrm{PW}'_i)$ equals $h(\mathrm{PW}_i)$ to guess a correct password.

After getting the correct ID_i, and PW_i, the malicious user U_A can easily change the password of user U_i and can impersonate U_i to login to the system.

4.4. Inefficient for Wrong Password Login. Generally speaking, in practical applications, the user U_i may keep different passwords for different applications to ensure security. Users are easy to confuse the password such that the user cannot match the application with the correct password; in other words, it is possible that the user enters a wrong password in the login phase.

In the login phase, the smart card does not verify the correctness of the entered password by the user. If the user U_i inputs a wrong password $\mathrm{PW}'_i (\neq \mathrm{PW}_i)$ by mistake, the smart card and the server will perform the following steps:

(1) generates a random number r, gets the current timestamp T, and computes $H_i = K_{x_i y_i} \oplus h(r \oplus T)$ and $S_i = \theta_i \oplus h(\mathrm{PW}'_i) \oplus r$;

(2) generates a random number a and computes $r_i = g^a \bmod p$ and $R_i = h(\theta_i \oplus r_i)$;

(3) encrypts $(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ with R_i and computes $C_i = \theta_i \oplus h(\mathrm{ID}_i \oplus K_{x_iy_i)}) \oplus h(\mathrm{PW}_i') \oplus R_i(= h(K_S) \oplus h(\mathrm{PW}_i) \oplus h(\mathrm{PW}_i') \oplus R_i)$;

(4) sends the login request message $M_i = (C_i, E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T), H_i, S_i, T)$ to the server;

(5) when the server receives the login request message M_i, the server computes $R_i' = C_i \oplus h(K_S) = h(K_S) \oplus h(\mathrm{PW}_i) \oplus h(\mathrm{PW}_i') \oplus R_i \oplus h(K_S) = h(\mathrm{PW}_i) \oplus h(\mathrm{PW}_i') \oplus R_i$. It is obvious that $R_i' \neq R_i$, since $\mathrm{PW}_i' \neq \mathrm{PW}_i$;

(6) when the server decrypts $E_{R_i}(\mathrm{ID}_i, r_i, U_C(x_i), v_i, T)$ using R_i', the server will find that the user U_i's identity is invalid. Thus, the server rejects the user U_i's login request.

In this case, the user U_i is unaware of the fact that he/she has entered his/her password incorrectly in the login phase, which results in unnecessary extra communication and computation costs.

5. The Proposed Scheme

In this section, we apply the LU decomposition of matrices to design a novel bilateral remote user authentication scheme with user anonymity, where LU decomposition of matrices ensures secretly information exchange between the user and the server, and enhances the security of the authentication scheme. To initiate the scheme, the server chooses a $n \times n$ symmetric matrix A with LU decomposition as $A = LU$ and secretly stores these matrices as his/her secret key in other servers, where n is the number of users the system can support. The server chooses a secret key K_S with 256 bits, which makes the K_S have a high entropy and can resist brutal force attack. The proposed scheme also contains four phases, that is, the registration phase, the login phase, the authentication and key agreement phase, and the password change phase. The proposed scheme contains the timestamps, so the authentication system needs to deploy a mechanism such as NTP (Network Time Protocol) to ensure clock synchronization between the user and the remote server. The detailed information about these phases are described as follows and also shown in Figure 1.

5.1. Registration Phase. When a user U_i wants to become a legal user of the system, U_i generates his own identity ID_i and easy-to-remember password PW_i and selects and remembers a random number b_i (the bit length of b_i is assumed to be 128). Then, U_i computes $\mathrm{RPW}_i = h(b_i \| \mathrm{PW}_i)$ and submits ID_i and RPW_i to the server over a secure communication channel for registration. $h(\cdot)$ used throughout the proposed scheme is a collision-free one-way hash function such as SHA-1 [31], which maps any message with the length less than 2^{64} bit to a 160-bit message digest. Upon receiving the registration request message, the server S and the user U_i take the following steps.

(1) The server S computes $A_i = h(\mathrm{ID}_i \| K_S)$, $B_i = A_i \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i)$, and $C_i = h(A_i)$.

(2) The server S chooses two random numbers $x_i, y_i \in [1, n]$ and computes $K_{x_iy_i} = L_R(x_i) \times U_C(y_i)$, $D_i = h(K_{x_iy_i} \oplus K_S) \oplus h(\mathrm{ID}_i \| \mathrm{RPW}_i)$, $E_i = h(K_S) \oplus y_i$, where the meaning of symbols $L_R(x_i)$ and $U_C(y_i)$ are the same as in Section 3.1.

(3) The server stores $\{B_i, C_i, K_{x_iy_i}, D_i, E_i, U_C(x_i), g, p\}$ into a smart card and issues the smart card to U_i via a secure channel, where p is a big prime and $p = 2q + 1$, q is also a big prime, and g is a primitive element with $g^q = 1 \bmod p$.

(4) At last, in order to facilitate the subsequent verification, the user U_i enters the remembered random number b_i into the smart card, and the smart card contains $\{B_i, C_i, K_{x_iy_i}, D_i, E_i, U_C(x_i), g, p, b_i\}$.

5.2. Login Phase. When the user U_i wants to login to the system, U_i inserts his/her smart card into the card reader and inputs his/her identity ID_i, password PW_i. Then the smart card performs the following operations:

(1) computes $\mathrm{RPW}_i = h(b_i \| \mathrm{PW}_i)$, $A_i = B_i \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i)$, and $C_i' = h(A_i)$ and compares $C_i' \overset{?}{=} C_i$. If they are equal, it means the user inputs the right identity and password. Otherwise, the input identity or password is not valid, and the smart card terminates the session;

(2) generates a random number r, gets current timestamp T, and computes $F_i = D_i \oplus h(\mathrm{ID}_i \| \mathrm{RPW}_i)$, $G_i = F_i \oplus r$, $\mathrm{CID}_i = r \oplus \mathrm{ID}_i$, and $H_i = E_i \oplus T$;

(3) generates a random number $r_i \in (1, q)$ and computes $R_i = g^{r_i} \bmod p$, $J_i = R_i \oplus h(r \oplus T)$, and $M_i = h(h(\mathrm{ID}_i \| K_{x_iy_i}) \| R_i \| r \| T)$, where M_i is used to resist the forgery attack such that any change of the login request message is invalid login message;

(4) submits the login request message $\{G_i, \mathrm{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$ to the server.

5.3. Verification and Key Agreement Phase. Upon receiving the login request message $\{G_i, \mathrm{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$, the server performs the following steps for mutual authentication and key agreement.

(1) The server S verifies the validity of the time interval between T' and T. If $(T' - T) \geq \Delta T$, S rejects the login request. Here T' is the timestamp, when the login request message was received, and ΔT is the expected valid time interval for transmission delay.

(2) The server S computes $y_i = H_i \oplus h(K_S) \oplus T$, $K_{y_ix_i} = L_R(y_i) \times U_C(x_i)$.

(3) The server S computes $r' = h(K_{y_ix_i} \oplus K_S) \oplus G_i$, $R_i' = J_i \oplus h(r' \oplus T)$, and $\mathrm{ID}_i = \mathrm{CID}_i \oplus r'$ and checks whether ID_i is the registered identity of a valid user. If so, the server S performs the following steps. Otherwise, the session is terminated.

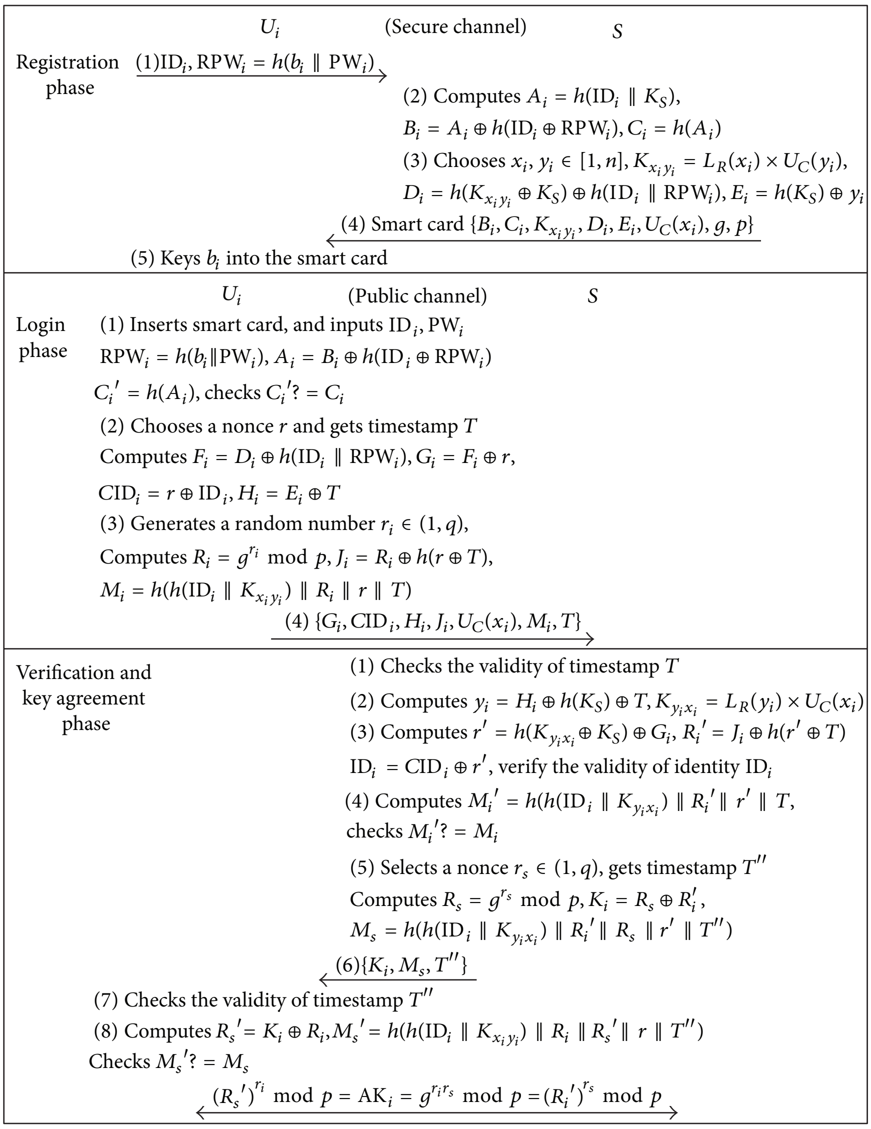

U_i
(Secure channel)
S
Registration phase
$(1)\mathrm{ID}_i, \mathrm{RPW}_i = h(b_i \parallel \mathrm{PW}_i)$
(2) Computes $A_i = h(\mathrm{ID}_i \parallel K_S)$,
$B_i = A_i \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i), C_i = h(A_i)$
(3) Chooses $x_i, y_i \in [1, n], K_{x_iy_i} = L_R(x_i) \times U_C(y_i)$,
$D_i = h(K_{x_iy_i} \oplus K_S) \oplus h(\mathrm{ID}_i \parallel \mathrm{RPW}_i), E_i = h(K_S) \oplus y_i$
(4) Smart card $\{B_i, C_i, K_{x_iy_i}, D_i, E_i, U_C(x_i), g, p\}$
(5) Keys b_i into the smart card
U_i
(Public channel)
S
Login phase
(1) Inserts smart card, and inputs $\mathrm{ID}_i, \mathrm{PW}_i$
$\mathrm{RPW}_i = h(b_i \| \mathrm{PW}_i), A_i = B_i \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i)$
$C_i{}' = h(A_i)$, checks $C_i{}'? = C_i$
(2) Chooses a nonce r and gets timestamp T
Computes $F_i = D_i \oplus h(\mathrm{ID}_i \parallel \mathrm{RPW}_i), G_i = F_i \oplus r$,
$\mathrm{CID}_i = r \oplus \mathrm{ID}_i, H_i = E_i \oplus T$
(3) Generates a random number $r_i \in (1, q)$,
Computes $R_i = g^{r_i} \bmod p, J_i = R_i \oplus h(r \oplus T)$,
$M_i = h(h(\mathrm{ID}_i \parallel K_{x_iy_i}) \parallel R_i \parallel r \parallel T)$
(4) $\{G_i, \mathrm{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$
Verification and key agreement phase
(1) Checks the validity of timestamp T
(2) Computes $y_i = H_i \oplus h(K_S) \oplus T, K_{y_ix_i} = L_R(y_i) \times U_C(x_i)$
(3) Computes $r' = h(K_{y_ix_i} \oplus K_S) \oplus G_i, R_i{}' = J_i \oplus h(r' \oplus T)$
$\mathrm{ID}_i = \mathrm{CID}_i \oplus r'$, verify the validity of identity ID_i
(4) Computes $M_i{}' = h(h(\mathrm{ID}_i \parallel K_{y_ix_i}) \parallel R_i{}' \parallel r' \parallel T$,
checks $M_i{}'? = M_i$
(5) Selects a nonce $r_s \in (1, q)$, gets timestamp T''
Computes $R_s = g^{r_s} \bmod p, K_i = R_s \oplus R_i'$,
$M_s = h(h(\mathrm{ID}_i \parallel K_{y_ix_i}) \parallel R_i{}' \parallel R_s \parallel r' \parallel T'')$
$(6)\{K_i, M_s, T''\}$
(7) Checks the validity of timestamp T''
(8) Computes $R_s{}' = K_i \oplus R_i, M_s{}' = h(h(\mathrm{ID}_i \parallel K_{x_iy_i}) \parallel R_i \parallel R_s{}' \parallel r \parallel T'')$
Checks $M_s{}'? = M_s$
$(R_s{}')^{r_i} \bmod p = \mathrm{AK}_i = g^{r_ir_s} \bmod p = (R_i{}')^{r_s} \bmod p$

FIGURE 1: The proposed scheme.

(4) The server S computes $M_i' = h(h(\mathrm{ID}_i \| K_{y_ix_i}) \| R_i' \| r' \| T)$ and checks $M_i' \overset{?}{=} M_i$. If they are equal, the validity of the user U_i is authenticated by the server S. Otherwise, the session is terminated by the server S.

(5) For achieving mutual authentication, the server chooses a random number $r_s \in (1, q)$, gets the current timestamp T'', and computes $R_s = g^{r_s} \bmod p$, $K_i = R_s \oplus R_i'$, and $M_s = h(h(\mathrm{ID}_i \| K_{y_ix_i}) \| R_i' \| R_s \| r' \| T'')$.

(6) The server S submits the reply message $\{K_i, M_s, T''\}$ to the user U_i for mutual authentication.

(7) After receiving the mutual authentication message (K_i, M_s, T''), the smart card verifies the validity of the time interval between T''' and T''. If $(T''' - T'') \geq \Delta T$, the user U_i terminates the session. Here T''' is the timestamp, when the mutual authentication message was received.

(8) The smart card computes $R_s' = K_i \oplus R_i$, $M_s' = h(h(\mathrm{ID}_i \| K_{x_iy_i}) \| R_i \| R_s' \| r \| T'')$ and checks $M_s' \overset{?}{=} M_s$. If they are equal, the server S is authenticated by the user U_i, and the server S and user achieve mutual authentication. Otherwise, the smart card terminates this session.

(9) At last, the user U_i and the server S can compute $\mathrm{AK}_i = R_s'^{r_i} \bmod p = g^{r_ir_s} \bmod p$ and $\mathrm{AK}_i = R_i'^{r_s} \bmod p = g^{r_ir_s} \bmod p$, respectively, as their shared session key for future secret communication.

5.4. Password Change Phase. When the user U_i wants to renew his/her password to $\mathrm{PW}_i^{\mathrm{new}}$, the user U_i can update his/her password by performing the following steps without communicating with the server S.

(1) The user U_i inserts his smart card into a card reader and inputs his identity ID_i and old password PW_i and requests to change his/her password.

(2) The smart card computes $\text{RPW}_i = h(b_i \| \text{PW}_i)$, $A_i = B_i \oplus h(\text{ID}_i \oplus \text{RPW}_i)$, $C_i' = h(A_i')$ and compares $C_i' \overset{?}{=} C_i$. If they are not equal, the password change request is rejected. Otherwise the user U_i inputs a new password PW_i^{new}.

(3) The smart card computes $\text{RPW}_i^* = h(b_i \| \text{PW}_i^{\text{new}})$, $B_i^* = A_i \oplus h(\text{ID}_i \oplus \text{RPW}_i^*)$, and $D_i^* = D_i^* \oplus h(\text{ID}_i \| \text{RPW}_i) \oplus h(\text{ID}_i \| \text{RPW}_i^*)$.

(4) Finally, the smart card replaces B_i and D_i with B_i^* and D_i^*, respectively, to update his/her password.

6. Analysis of the Proposed Scheme

In this section, we first discuss the security features of the proposed anonymity bilateral authentication scheme. Then we evaluate the performance and functionality of our proposed scheme and make comparisons with Tseng et al.' scheme.

6.1. Security of Session Key

6.1.1. Known-Key Secrecy. Known-key secrecy means that compromise of one session key should not compromise other session keys. In our scheme, the session key $\text{AK}_i = g^{r_i r_s} \bmod p$ is associated with r_i, $g^{r_i} \bmod p$, r_s, and $g^{r_s} \bmod p$. According to discrete logarithm problem (DLP) and Diffie-Hellman problem (DHP), knowing a session key $\text{AK}_i = g^{r_i r_s} \bmod p$ and the random number r_i, r_s is useless for computing the other session keys $\text{AK}_i' = g^{r_i' r_s'} \bmod p$ without knowing r_i' and r_s'. It is impossible for an attacker to compute the other session key AK_i', and the proposed scheme provides known-key security.

6.1.2. Forward Secrecy. Forward secrecy means that if the long-term secret keys (e.g., the server's secret key K_S and user's password PW_i) are compromised, the secrecy of previously established session keys should not be affected. In our scheme, we assume that the master secret key K_S and the password PW_i of user U_i are compromised for some reasons, and the attacker gets the previous communication message $\{G_i, \text{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$ and $\{K_i, M_s, T''\}$ from the public channel; then the attacker can get $y_i = H_i \oplus h(K_S) \oplus T$. However, since the secret matrix L has been maintained only by the server S, the attacker cannot compute $K_{x_i y_i}$, r and has no way to know r_i, R_i, r_s, and R_s. Therefore, the attacker has no way to get the previous session key $\text{AK}_i = g^{r_i r_s} \bmod p$, and our scheme can ensure perfect forward secrecy.

6.2. Protect User Anonymity. In the login phase and authentication phase of the proposed scheme, the real identity ID_i of user U_i is not transmit via plain text form. If the login request message $\{G_i, \text{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$ and the mutual authentication message $\{K_i, M_s, T''\}$ are eavesdropped by an attacker from the public channel, the attacker has to get the random number r to compute the real identity ID_i. However, the attacker has no way to know $K_{x_i y_i}$ and K_S, so he/she has no valid method to get the random number r and cannot reveal the real identity ID_i of the user U_i. Therefore, our scheme can really protect user anonymity.

6.3. Resist Impersonation Attack. In this type of attack, in order to impersonate as a legitimate user, the attacker or a malicious user has to forge a valid login request message $\{G_i, \text{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$ using the previously eavesdropped messages or the information obtained from the lost smart card. However, in the proposed scheme, the attacker and any malicious user U_A cannot forge a valid login request message, since he/she has no knowledge of ID_i, RPW_i, F_i, and $K_{x_i y_i}$, so he/she cannot impersonate as the legitimate user U_i.

In addition, even if the adversary or a malicious user has obtained the smart card of user U_i and extracts the parameters $\{B_i, C_i, K_{x_i y_i}, D_i, E_i, U_C(x_i), g, p, b_i\}$ which are stored in the smart card by some way, he/she still cannot forge a valid login request message, since he/she have no way to get the valid ID_i, PW_i, where they are all protected by the one-way hash function.

Therefore, the proposed protocol is secure against impersonation attack. At the same time, the attacker cannot get the valid ID_i, PW_i, so the proposed protocol can resist the denial of service attack.

6.4. Resist Insider Attack. In the registration phase of the proposed scheme, the user U_i freely selects his/her password PW_i and submits the masked password $\text{RPW}_i = h(b_i \oplus \text{PW}_i)$ instead of $h(\text{PW}_i)$ to the server for registration. In the proposed scheme, the password must first be verified by the smart card in login and password change phase, only the adversary gets the valid password PW_i of the user U_i, and he/she can impersonate the user U_i to access service. However, if the insider of the remote system gets the information ID_i and $\text{RPW}_i = h(b_i \oplus \text{PW}_i)$, he/she cannot obtain the password PW_i since it is protected by the one-way hash function and cannot impersonate the user U_i to login to the system or change the user's password. Therefore, the proposed scheme can resist insider attack properly.

6.5. Resist Stolen Smart Card Attack. Assume that the user U_i's smart card has been lost or stolen, the attacker can extract the stored information $\{B_i, C_i, K_{x_i y_i}, D_i, E_i, U_C(x_i), g, p, b_i\}$ from the smart card using differential power analysis [29] and simple power analysis [30]. Even after gathering these information, in order to change the user's password or login into the system by using the lost smart card, the attacker has to get real identity ID_i and the password PW_i correctly at the same time. However, because the attacker has not the knowledge of the master secret key K_S and meanwhile the ID_i and the PW_i are protected by one-way hash function, it is not possible for an attacker to guess the ID_i and the PW_i correctly

Table 2: Performance comparison of our scheme and Tseng et al.'s scheme.

	Our scheme	Tseng et al.'s scheme [27]
Registration phase	$1T_M + 6T_H$	$1T_M + 3T_H$
Login phase	$6T_H + 1T_{\text{EXP}}$	$4T_H + 1T_{\text{EXP}} + 1T_{\text{ENC}}$
Verification and key agreement phase	$1T_M + 6T_H + 3T_{\text{EXP}}$	$1T_M + 2T_H + 3T_{\text{EXP}} + 1T_{\text{ENC}} + 2T_{\text{DEC}}$
Total	$2T_M + 18T_H + 4T_{\text{EXP}}$	$2T_M + 9T_H + 4T_{\text{EXP}} + 2T_{\text{ENC}} + 2T_{\text{DEC}}$

Table 3: Functionality comparison between our scheme and Tseng et al.'s scheme.

	Our scheme	Tseng et al.'s scheme [27]
Protect user anonymity	Yes	No
Resist impersonation attack	Yes	Yes
Resist insider attack	Yes	No
Resist stolen smart card attack	Yes	No
Resist server spoofing attack	Yes	Yes
Efficient for password verification	Yes	No
Mutual authentication	Yes	Yes
Session key agreement	Yes	Yes

at the same time in real polynomial time. Therefore, the proposed scheme is secure against stolen smart card attack.

6.6. Resist Server Spoofing Attack. In the proposed scheme, in order to masquerade as the remote server to cheat the user U_i, the attacker has to get the secret information L and K_S to compute the valid reply mutual authentication message. However, the secret matrix L is only maintained by the server S such that the attacker has no way to recover the information $K_{y_i x_i}$. On the other hand, even if the malicious user U_A has got his own smart card information $\{B_A, C_A, K_{x_A y_A}, D_A, E_A, U_C(x_A), g, p, b_A\}$ and other users' communication messages $\{G_i, \text{CID}_i, H_i, J_i, U_C(x_i), M_i, T\}$ and $\{K_i, M_s, T''\}$, he/she still has no way to get K_S since it is protected by the one-way hash function. So the attacker cannot get the required information L and K_S, and the proposed scheme can resist the server spoofing attack.

6.7. Efficient for Wrong Password Verification. In the login and password change phase of the proposed scheme, the validity of the password PW_i can quickly be verified by the smart card, when the user U_i inputs his/her password. If the user U_i inputs a wrong password $\text{PW}_i'(\neq \text{PW}_i)$, the smart card computes $\text{RPW}_i' = h(b_i \| \text{PW}_i')(\neq h(b_i \| \text{PW}_i) = \text{RPW}_i$, $A_i' = B_i \oplus h(\text{ID}_i \| \text{RPW}_i') = A_i \oplus h(\text{ID}_i \oplus \text{RPW}_i) \oplus h(\text{ID}_i \| \text{RPW}_i')(\neq A_i)$ and gets $C_i' = h(A_i') \neq h(A_i) = C_i$. So, the wrong password can quickly be checked by the smart card, and the server does not need to waste unnecessary communication and computation cost to verify the validity of the password. Thus, the proposed scheme is efficient for wrong password verification.

6.8. Performance and Functionality Analysis. In this section, we evaluate the performance and functionality of our proposed scheme and make comparisons with Tseng et al.'s scheme. In order to facilitate the computational complexity analysis of the scheme, we define the following notations.

T_h: the time for executing a one-way hash function $h(\cdot)$,

T_M: the time for performing a vector multiplication operation,

T_{EXP}: the time for performing an exponentiation operation,

T_{ENC}: the time for performing a symmetric encryption operation, and

T_{DEC}: the time for performing a symmetric decryption operation.

Because exclusion-OR operation requires very few computations, we neglect considering its computational cost in this paper. We list the result of performance comparison in Table 2, and we can see that the total computational cost of our scheme and Tesng et al.'s scheme are $2T_M + 18T_H + 4T_{\text{EXP}}$ and $2T_M + 9T_H + 4T_{\text{EXP}} + 2T_{\text{ENC}} + 2T_{\text{DEC}}$, respectively. Since the symmetric cryptosystem needs more computational costs than the one-way hash functions, our scheme is more efficient than Tseng et al.'s scheme.

Table 3 shows the functional comparison of our proposed scheme and Tseng et al.'s scheme. Compared with Tseng et al.'s scheme, our scheme can resist various attacks and can really protect user anonymity. Besides, our scheme can quickly check the validity of the password in the very beginning of login phase. Therefore, our scheme is more secure and efficient than Tseng et al.'s scheme.

7. Conclusions

In this paper, we have applied the LU decomposition of matrices to present a novel anonymity bilateral authentication scheme. First, we pointed out the security weaknesses of Tseng et al.'s scheme, that is, their scheme is vulnerable to insider attack and stolen smart card attack, is inefficient for wrong password login, and does not really provide user anonymity. To surmount these identified weaknesses, we have proposed a novel scheme using the LU decomposition of matrices to reduce computational complexity and improve security, where LU decomposition of matrices ensures secretly information exchange between the user and the server, and enhances the security of the authentication scheme. Hence, our proposed protocol is more efficient and practical.

Acknowledgments

The authors are grateful to the editor and anonymous reviewers for their valuable suggestions which improved the paper. This work was supported by the Research Fund of the State Key Laboratory of Software Development Environment under Grant no. BUAA SKLSDE-2012ZX-17, the National Natural Science Foundation of China under Grant no. 61271041, 61170296, and 61190120, and the Program for New Century Excellent Talents in University under Grant no. NECT-09-0028.

References

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

[2] W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards," *Computers & Security*, vol. 18, no. 8, pp. 727–733, 1999.

[3] J. K. Lee, S. R. Ryu, and K. Y. Yoo, "Fingerprint-based remote user authentication scheme using smart cards," *Electronics Letters*, vol. 38, no. 12, pp. 554–555, 2002.

[4] J. L. Tsai, T. C. Wu, and K. Y. Tsai, "New dynamic ID authentication scheme using smart cards," *International Journal of Communication Systems*, vol. 23, no. 12, pp. 1449–1462, 2010.

[5] R. C. Wang, W. S. Juang, and C. L. Lei, "Robust authentication and key agreement scheme preserving the privacy of secret key," *Computer Communications*, vol. 34, no. 3, pp. 274–280, 2011.

[6] 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.

[7] H. J. Jeong, D. G. Won, and S. J. Kim, "Weaknesses and improvement of secure hash-based strong-password authentication protocol," *Journal of Information Science and Engineering*, vol. 26, no. 5, pp. 1845–1858, 2010.

[8] A. S. K. Pathan, "A review and cryptanalysis of similar timestamp-based passwordauthentication schemes using smart cards," *International Journal of Communication Networks and Information Security*, vol. 2, no. 1, pp. 15–20, 2010.

[9] C. K. Chan and L. M. Cheng, "Cryptanalysis of a timestamp-based password authentication scheme," *Computers and Security*, vol. 21, no. 1, pp. 74–76, 1998.

[10] H. M. Sun and H. T. Yeh, "Further cryptanalysis of a password authentication scheme with smart cards," *IEICE Transactions on Communications*, vol. E86-B, no. 4, pp. 1412–1415, 2003.

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

[12] W. C. Ku, S. T. Chang, and M. H. Chiang, "Further cryptanalysis of fingerprint-based remote user authentication scheme using smartcards," *Electronics Letters*, vol. 41, no. 5, pp. 240–241, 2005.

[13] S. H. Wu, Y. F. Zhu, and Q. Pu, "Robust smart-cards-based user authentication schemewith user anonymity," *Security and Communication Networks*, vol. 5, no. 2, pp. 236–248, 2012.

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

[15] H. Jung and H. S. Kim, "Secure hash-based password authentication protocol using smartcards," in *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA '11)*, vol. 6786 of *Lecture Notes in Computer Science*, pp. 593–606, Springer, 2011.

[16] 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.

[17] X. Li, Y. P. Xiong, J. Ma, and W. D. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

[18] X. Li, J. Ma, W. D. Wang, Y. P. Xiong, and J. S. Zhang, "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments," *Mathematical and Computer Modelling*. In press.

[19] W. B. Lee and C. C. Chang, "User identification and key distribution maintaining anonymity for distributed computer networks," *Computer Systems Science and Engineering*, vol. 15, no. 4, pp. 211–214, 2000.

[20] T. S. Wu and C. L. Hsu, "Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks," *Computers and Security*, vol. 23, no. 2, pp. 120–125, 2004.

[21] Y. Yang, S. Wang, F. Bao, J. Wang, and R. H. Deng, "New efficient user identification and key distribution scheme providing enhanced security," *Computers & Security*, vol. 23, no. 8, pp. 697–704, 2004.

[22] K. Mangipudi and R. Katti, "A Secure Identification and Key agreement protocol with user Anonymity (SIKA)," *Computers & Security*, vol. 25, no. 6, pp. 420–425, 2006.

[23] R. C. Wang, W. S. Juang, and C. L. Lei, "Provably secure and efficient identification and key agreement protocol with user anonymity," *Journal of Computer and System Sciences*, vol. 77, no. 4, pp. 790–798, 2011.

[24] S. J. Choi and H. Y. Youn, "A novel data encryption and distribution approach for high security and availability using LU decomposition," in *Proceedings of the International Conference on Computation Science and Its Application (ICCSA '04)*, vol. 3046 of *Lecture Notes in Computer Science*, pp. 637–646, May 2004.

[25] A. S. K. Pathan and C. S. Hong, "An efficient bilateral remote user authenticationscheme with smart cards," in *Proceedings of the 33rd Korea Information Science Society Fall Conference*, vol. 33, no. 2(D), pp. 132–134, October 2006.

[26] A. S. K. Pathan, C. S. Hong, and T. Suda, "A novel and efficient bilateral remote user authentication scheme using smart cards," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '07)*, pp. 1–2, January 2007.

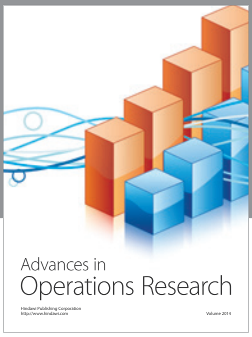
[27] H. R. Tseng, R. H. Jan, and W. Yang, "A bilateral remote user authentication scheme that preserves user anonymity," *Security and Communication Networks*, vol. 1, no. 4, pp. 301–308, 2008.

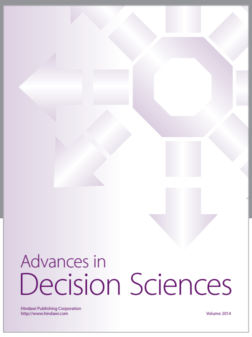
[28] B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

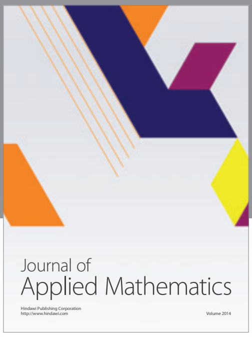
[29] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99)*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, 1999.

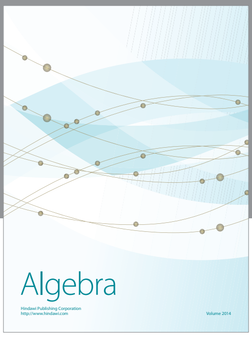
[30] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

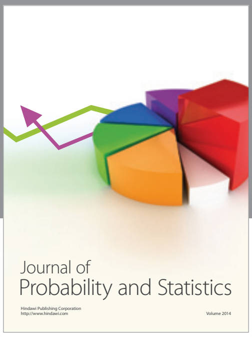
[31] National Institute of Standards and Technology, US Department of Commerce, Secure Hash Standard, US Federal Information Processing Standard Publication 180-2, 2002.


Advances in
Operations Research

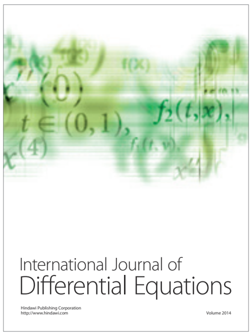

Advances in
Decision Sciences


Journal of
Applied Mathematics

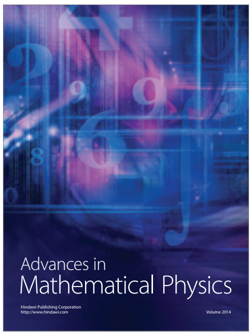

Algebra

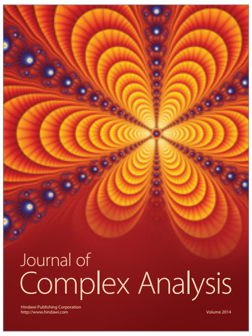

Journal of
Probability and Statistics

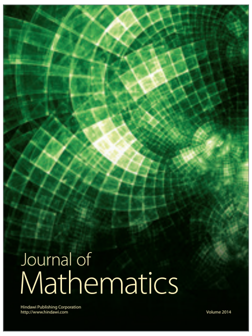
The Scientific
World Journal

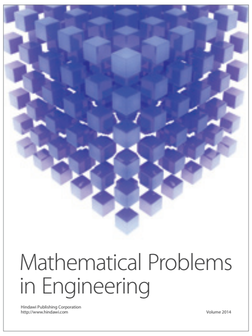

International Journal of
Differential Equations

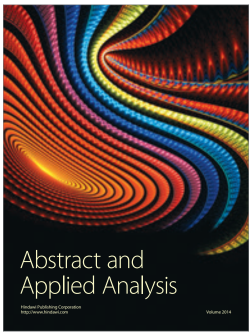
International Journal of
Combinatorics

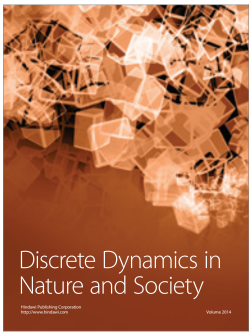

Advances in
Mathematical Physics

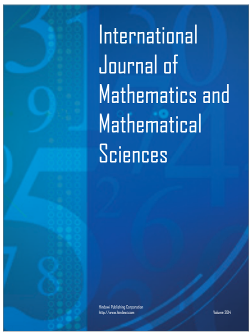

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

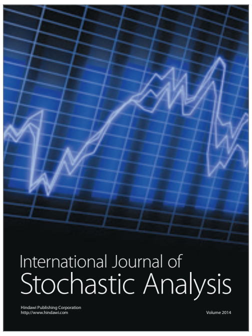

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization