ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **15**, Number 3, 2012, 266–276

An Efficient RSA-Based Remote User Authentication Scheme

Wenbo SHI[1], Yitao CHEN[2]

[1]Department of Electronic Engineering,
Northeastern University at Qinhuangdao, Qinhuangdao, China
E-mail: swb319@hotmail.com

[2]School of Mathematics and Statistics,
Wuhan University, Wuhan, China
E mail: chenyitao.math@gmail.com

Abstract. Password authentication has been adopted as one of the most commonly used solutions in network environment to protect resources from unauthorized access. Recently, Awasthi et al. proposed an efficient RSA-based remote user authentication scheme. In this paper, we will point out that Awasthi et al.'s scheme is vulnerable to a privileged insider attack, a password guessing attack and an impersonation attack. To improve the security, we also propose a new RSA-based remote user authentication scheme. The analysis shows our scheme could overcome the weaknesses in Awasthi et al.'s scheme and has better performance than their scheme.

Key-words: Authentication; Smart card; RSA.

1. Introduction

Remote authentication is a method to authenticate remote users over insecure communication channel. Password-based authentication schemes have been widely deployed to verify the legitimacy of remote users. In 1981, Lamport [1] proposed the first password-based authentication scheme using password tables to authenticate remote users over insecure network. Since then, many password authentication schemes have been proposed to improve security or efficiency or cost.

Since RSA has been applied in the industry for decades, many companies may have invested in expensive hardware or software implementations of RSA. Then the

RSA-based remote user authentication scheme is very convenient for applications. In 1999, Yang et al. proposed the first RSA-based remote user authentication scheme. Compared with Lamport's scheme [1], Yang et al.'s scheme needs no password tables or verification tables. Then Yang et al.'s is more practical than Lamport's scheme. However, many scholars have pointed that Yang et al.'s scheme was vulnerable to the forged login attacks [3-5]. In 2003, Shen et al. [5] also proposed an improved scheme to overcome the security vulnerability in Yang et al.'s scheme. Yoon et al. [6] pointed out that Shen et al.'s scheme was still vulnerable to the forged login attacks. However, Yoon et al. did not give the solution to resist attacks. Later, Liu et al. [7] pointed out that Shen et al.'s scheme is vulnerable to another forged login attack. They also proposed an improved scheme to withstand the attack. Recently, Awasthi et al. [8] pointed that Liu et al.'s scheme is vulnerable to the replay attack. Then Awasthi et al. proposed an efficient RSA-based remote user authentication scheme. In this paper, we will show Awasthi et al.'s scheme is vulnerable to a privileged insider attack, a password guessing attack and an impersonation attack. To improve the security, we propose a new RSA-based remote user authentication scheme. The analysis shows our scheme not only overcome the weaknesses in Awasthi et al.'s scheme but also needs no extra operations.

The rest of this paper is organized as follows. In the next section, we will review Awasthi et al.'s scheme. We propose our attacks against Awasthi et al.'s scheme in Section 3. In Section 4, we propose our RSA-based remote user authentication scheme. The security and performance analysis is proposed in Section 5 and Section 6 separately. Finally, Section 7 concludes the paper.

2. Review of Awasthi et al.'s scheme

In Awasthi et al.'s scheme [8], the key information center (KIC), a trusted authority, is responsible for generating global parameters, computing users' secret information and providing smartcards to the new users. There are four phases in their scheme: **initialization**, **registration**, **login**, and **authentication**.

- **Initialization phase**

KIC performs the following steps:

1. Generate two large primes p and q and computes $n = pq$.
2. Choose a prime number e and an integer d such that $ed \equiv 1 \bmod (p-1)(q-1)$, where e is the system public key, d is the corresponding private key, which should be provided to the server in a safe way.
3. Find an integer g, which is a primitive element in both $GF(p)$ and $GF(q)$ and the public information in the system.

- **Registration phase**

A new user U_i securely submits his identifier ID_i and password pw_i to the KIC. The KIC, as shown in Fig. 1, performs the following steps:

1. Generate the smart card's identifier CID_i for user U_i and h_i as $CID_i = f(ID_i \oplus d)$, $h_i = g^{pw_i \cdot d} \bmod n$, where $f(\cdot)$ is a one way function.
2. Calculate the user's secret information $S_i = CID_i^d \bmod n$.
3. Write n, e, g, ID_i, S_i and h_i into the smart card of U_i, and issue the smartcard to the user through a secure channel.

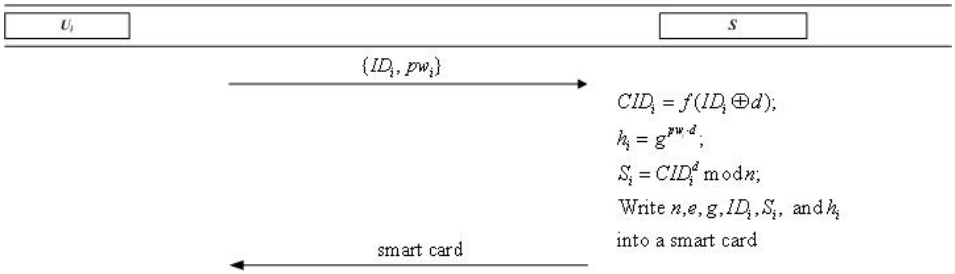

Fig. 1. Registration phase of Awasthi et al.'s scheme.

- **Login phase**

In this phase, as shown in Fig. 2, user U_i performs the following steps:

1. Inputs the password pw_i, choose a random number r_i and compute $X_i = g^{r_i \cdot pw_i} \bmod n$ and $Y_i = S_i \cdot h_i^{r_i \cdot f(ID_i, T_i)}$, where T_i is the timestamp at the login device.
2. Send the login request message $M_1 = \{ID_i, X_i, Y_i, n, e, g, T_i\}$ to the remote server S.

- **Authentication phase**

After receiving the login request message M_1 from U_i, as shown in Fig. 2, the remote server will perform the following steps to verify the correctness of M_1.

1. Verify that ID_i is a valid user identifier. If it is not then reject the login request.
2. Check the validity of T_c. If $T_s - T_i > \Delta T$, then the server rejects the login request, where T_s is the current timestamp at the remote server and ΔT is expected legitimate time interval for transmission delay.
3. Compute $CID_i = f(ID_i \oplus d)$.
4. Check the equation $Y_i^e = CID_i \cdot X_i^{f(ID_i, T_i)}$. If it holds, accept the login request, otherwise reject.
5. Compute $R = (f(ID_i, T_s'))^d \bmod n$ and send $M_2 = \{R, T_s'\}$, where T_s' is the current timestamp on the remote server. Upon receiving the message M_2 from the server, the user U_i verifies the server as follows.

6. Check the validity of T_s'. If $T_i' - T_s' > \Delta T$, then U_i rejects the login request, where T_i' is the current timestamp at the user and ΔT is expected legitimate time interval for transmission delay.

7. Compute $R' = R^e \bmod n$. If $R' = f(ID_i, T_s')$, U_i accept the server otherwise reject server and disconnect it.

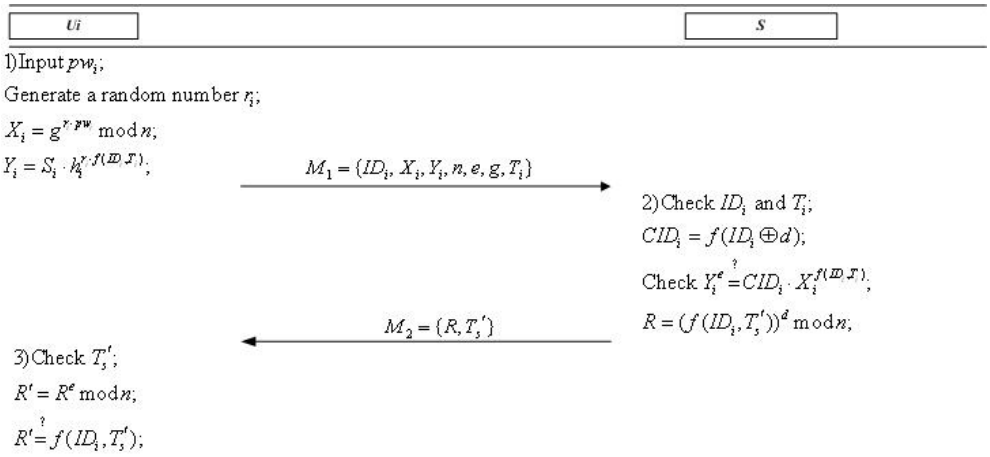

Fig. 2. Login and authentication phase of Awasthi et al.'s scheme.

3. Weaknesses of Awasthi et al.'s scheme

Kocher et al. [9] and Messerges et al. [10] have pointed out that all existent smart cards are vulnerable in that the confidential information stored in the device could be extracted by physically monitoring its power consumption; once a card is lost, all secrets in it may be revealed. To evaluate the security of smart card based user authentication, we assume the capabilities that an adversary $\mathscr{A}$ may have as follows:

1. The adversary has total control over the communication channel between the users and the server in the login and authentication phases. That is, $\mathscr{A}$ may intercept, insert, delete, or modify any message in the channel.

2. $\mathscr{A}$ may (i) either steal a user's smart card and then extract the information from it, (ii) or obtain a user's password, (iii) but not both (i) and (ii).

In this section, we shall prove that Awasthi et al.'s scheme is vulnerable to a privileged insider attack, a password guessing attacks and an impersonation attack. A more detailed description of attacks is as follows.

3.1. Privileged Insider attack

In a real environment, it is a common practice that many users use same passwords to access different applications or servers for their convenience of remembering long

passwords and ease-of-use whenever required [11]. However, if the system manager or a privileged insider $\mathscr{A}$ of the server S knows the passwords of user U_i, he may try to impersonate U_i by accessing other servers where U_i could be a registered user. In the user registration phase of Awasthi et al.'s scheme, U_i sends his identity ID_i, the password pw_i to S directly. Then the privileged insider $\mathscr{A}$ could get U_i's password. Therefore, Awasthi et al.'s scheme is vulnerable to the privileged insider attack.

3.2. Password guessing attack

In remote user authentication schemes that the user is allowed to choose his password, the client tends to choose a password that can be easily remembered for his convenience [11]. However, these easy-to-remember passwords are potentially vulnerable to password guessing attack, in which an adversary can try to guess the client's password and then verify his guess.

Suppose an adversary $\mathscr{A}$ has stolen U_i's smart card and extracted the stored values n, e, g, ID_i, S_i and h_i through someway[9, 10], where $h_i = g^{pw_i \cdot d}$, $CID_i = f(ID_i \oplus d)$ and $S_i = CID_i^d \bmod n$. Then the attacker $\mathscr{A}$ can successfully find out U_i's password pw_i by performing the following procedure.

1) $\mathscr{A}$ computes $l_i \equiv h_i^e \equiv (g^{pw_i \cdot d})^e \equiv g^{pw_i} \bmod n$.

2) $\mathscr{A}$ guesses a password pw_i' and computes $l_i' \equiv g^{pw_i'} \bmod n$.

3) $\mathscr{A}$ checks whether l_i and l' are equal. If they are equal, then $\mathscr{A}$ finds the correct password. Otherwise, $\mathscr{A}$ repeats 2) and 3) until the correct password is found.

From the above description, we know the adversary can get the password. Therefore, Awasthi et al.'s scheme is vulnerable to the offline password guessing attack.

3.3. Impersonation attack

Suppose an adversary $\mathscr{A}$ has stolen U_i's smart card and extracted the stored values n, e, g, ID_i, S_i and h_i, where $h_i = g^{pw_i \cdot d}$, $CID_i = f(ID_i \oplus d)$ and $S_i = CID_i^d \bmod n$. Then the attacker $\mathscr{A}$ could impersonate U_i to login in the server by performing the following procedure.

1) $\mathscr{A}$ computes $CID_i = S_i^e = (CID_i^d)^e \bmod n$.

2) Since the public key e is a prime number, then the greatest common divisor of e and $f(ID_i, T_c)$ is one. Then $\mathscr{A}$ could find two integer a and b through Extended Euclidean algorithm such that $ae + bf(ID_i, T_c) = 1$, where T_c is the timestamp at the login device.

3) $\mathscr{A}$ computes $X_i = CID_i^{-b} \bmod n$ and $Y_i = CID_i^a \bmod n$.

4) At last, $\mathscr{A}$ sends the login request message $M_1 = \{ID_i, X_i, Y_i, n, e, g, T_c\}$ to the remote server S.

Since

$$\begin{aligned} &CID_i \cdot X_i^{f(ID_i,T_c)} = CID_i \cdot (CID_i^{-b})^{f(ID_i,T_c)} \\ &= CID_i^{1-b\cdot f(ID_i,T_c)} = CID_i^{a\cdot e} \\ &= Y_i^e \end{aligned} \tag{1}$$

then M_1 could pass the verification of the server. Therefore, $\mathscr{A}$ could impersonate U_i successfully and Awasthi et al.'s scheme is vulnerable to the impersonation attack.

4. The improved scheme

Like Awasthi et al.'s scheme, our scheme also has four phases: **initialization**, **registration**, **login**, and **authentication**.

- **Initialization phase**

KIC performs the following steps:

1. Generate two large primes p and q and computes $n = pq$.
2. Choose a prime number e and an integer d such that $ed \equiv 1 \bmod (p-1)(q-1)$, where e is the system public key, d is the corresponding private key, which should be provided to the server in a safe way.

- **Registration phase**

A new user U_i generates a random number N_i, securely submits his identifier ID_i and $f(pw_i \oplus N_i)$ to the KIC. The KIC, as shown in Fig. 3, performs the following steps:

1. Generate the smart card's identifier $CID_i = f(ID_i \oplus d)$.
2. Calculate the user's secret information $S_i = CID_i \oplus f(pw_i \oplus N_i)$.
3. Write n, e, ID_i and S_i into the smart card of U_i, and issue the smartcard to the user through a secure channel.
4. After receiving the smart card, U_i inserts N_i into it and finishes the registration.

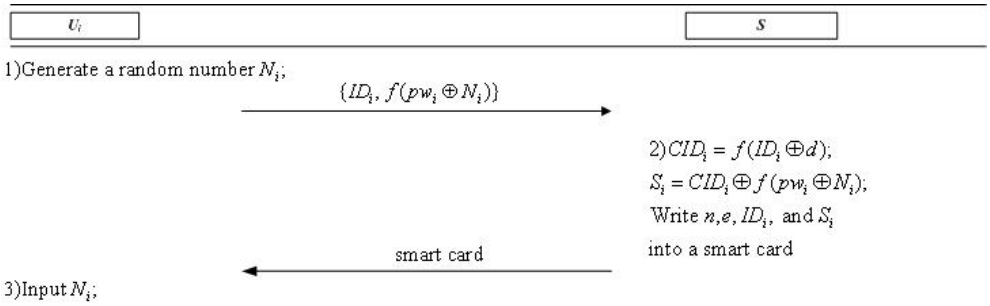

Fig. 3. Registration phase of our scheme.

- **Login phase**

In this phase, as shown in Fig. 4, user U_i performs the following steps:

1. Inputs the password pw_i, compute $CID_i = S_i \oplus f(pw_i \oplus N_i)$, choose a random number r_i and compute $A_i = f(CID_i||r_i||T_i)$ and $X_i = (A_i||r_i)^e \bmod n$, where T_i is the timestamp at the login device and $f(\cdot)$ is a one way function.
2. Send the login request message $M_1 = \{ID_i, X_i, n, e, T_i\}$ to the remote server S.

- **Authentication phase**

After receiving the login request message M_1 from U_i, as shown in Fig. 4, the remote server will perform the following steps to verify the correctness of M.

1. Verify that ID_i is a valid user identifier. If it is not then reject the login request.
2. Check the validity of T_i. If $T_s - T_i > \Delta T$, then the server rejects the login request, where T_s is the current timestamp at the remote server and ΔT is expected legitimate time interval for transmission delay.
3. Compute $CID_i = f(ID_i \oplus d)$ and $A_i'||r_i' = X_i^d \bmod n$.
4. Check the equation $A_i' = f(CID_i||r_i'||T_i)$. If it holds, accept the login request, otherwise reject.
5. Compute $R = f(ID_i, T_i, r_i', T_s')$ and send $M_2 = \{R, T_s'\}$, where T_s' is the current timestamp on the remote server. Upon receiving the message M_2 from the server, the user U_i verifies the server as follows.
6. Check the validity of T_s'. If $T_i' - T_s' > \Delta T$, then U_i rejects the login request, where T_i' is the current timestamp at the user and ΔT is expected legitimate time interval for transmission delay.
7. Compute $R' = f(ID_i, T_i, r_i, T_s')$. If $R' = R$, U_i accept the server otherwise reject server and disconnect it.

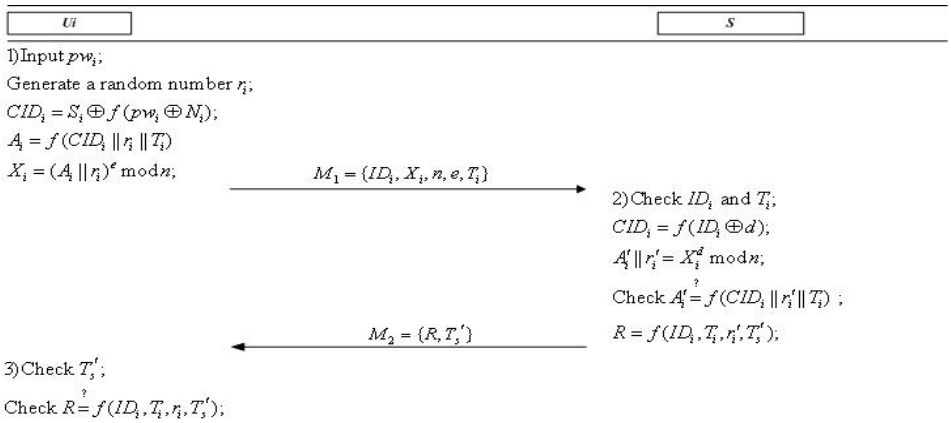

Fig. 4. Login and authentication phase of our scheme.

5. Security analysis

In this section, we will analyze the security of our scheme. We will show our scheme could withstand various attacks. The details are described as follows.

Theorem 1. *Our scheme could withstand the user impersonation attack.*

Proof. To impersonate a legal user to login in the server, the adversary could generate a legal message $M_1 = \{ID_i, X_i, n, e, T_i\}$, where $A_i = f(CID_i||r_i||T_i)$, $X_i = (A_i||r_i)^e \bmod n$ and $CID_i = f(ID_i \oplus d)$. However, the adversary could not generate the message since he does not the server's secret d. Therefore, our scheme could withstand the user impersonation attack. □

Theorem 2. *Our scheme could withstand the server masquerading attack.*

Proof. To impersonate the server to the user, the adversary could generate a legal message $M_2 = \{R, T_s'\}$ after receiving the message $M_1 = \{ID_i, X_i, n, e, T_i\}$, where$A_i = f(CID_i||r_i||T_i)$, $X_i = (A_i||r_i)^e \bmod n$, $CID_i = f(ID_i \oplus d)$ $R = = f(ID_i, T_i, r_i, T_s')$. However, the adversary cannot get r_i from X_i without the server's secret d. Then, he cannot generate the message R. Therefore, our scheme could withstand the server masquerading attack. □

Theorem 3. *Our scheme could withstand the privileged insider attack.*

Proof. In the registration phase of our scheme, the user sends $f(pw_i \oplus N_i)$ to the server. The privileged insider of the server could get $f(pw_i \oplus N_i)$. However, he cannot get the user's password pw_i since it is protected by a secure hash function. Therefore, our scheme could withstand the privileged insider attack. □

Theorem 4. *Our scheme is secure against password guessing attacks.*

Proof. The adversary could extract the stored data n, e, ID_i, S_i and N_i in the smart card through physically monitoring its power consumption, where $S_i = = CID_i \oplus f(pw_i \oplus N_i)$ and $CID_i = f(ID_i \oplus d)$. He could also get the message $M_1 = \{ID_i, X_i, n, e, T_i\}$, $A_i = f(CID_i||r_i||T_i)$ and $X_i = (A_i||r_i)^e \bmod n$. The adversary could pw' and computes $CID_i' = S_i \oplus f(pw_i' \oplus N_i)$. However, he cannot verify the correctness of pw' without the server's secret d. Therefore, our scheme could withstand the password guessing attack. □

Theorem 5. *Our scheme could provide the mutual authentication.*

Proof. From Theorem 1 and Theorem 2, we know that only the legal user and the server could generate $M_1 = \{ID_i, X_i, n, e, T_i\}$ and $M_2 = \{R, T_s'\}$ separately. Therefore, the server and the user could authenticate each other by checking the legality of M_1 and M_2 separately. Therefore, our scheme could provide the mutual authentication. □

Theorem 6. *Our scheme could withstand the replay attack.*

Proof. Suppose that the adversary intercept the login message $M_1 = \{ID_i, X_i, n, e, T_i\}$ and replay it to the server. However, the server could find the attack easily by

checking the freshness of T_i. Through the same analysis, we know that the user also could find the replay attack by checking the freshness of T_s'. Therefore, our scheme could withstand the replay attack. □

6. Performance analysis

In this section, we will compare the performance of our scheme with that of Awasthi et al.'s scheme. To evaluate the performance, we assume the output size of the cryptographic hash function is 160 bits. We also assume that the public key e and the modulus n in RSA algorithm are of 32 bits and 1024 bits separately.

In the storage cost concern, our scheme requires the smart card to store the parameters n, e, ID_i, S_i and N_i instead of the parameters n, e, g, ID_i, S_i and h_i in Awasthi et al.'s scheme. We can further estimate that the parameters n, e, g, ID_i, S_i and h_i in Awasthi et al.'s scheme need 1024 + 32 + 1024 + 32 + 1024 + 1024 = 4160 bits of storage space, where the identifier ID_i can be 32 bits. Correspondingly, the parameters n, e, ID_i, S_i and N_i in our scheme need 1024 + 32 + 32 + 1024 + 64 = 2176 bits of storage space, where the identifier ID_i can be 32 bits, the random number N_i can be 64 bits. Besides, both of the server S in our scheme and Awasthi et al.'s scheme need a 1024 bits storage space for the secret parameter d. In Table 1, we show the storage costs of Awasthi et al.'s scheme and our scheme.

Table 1. The storage cost of Awasthi et al.'s scheme and our scheme

	Awasthi et al.'s scheme	our scheme
Smart Card	4160 bits	2176 bits
Server	1024 bits	1024 bits

In the communication cost concern, our scheme exchanges data ID_i, X_i, n, e, T_i, R and T_s', while Awasthi et al.'s scheme exchanges data $ID_i, X_i, Y_i, n, e, g, T_c, R$ and T_s'. Let the length of timestamp T_c and T_s' be 32 bits. The communication cost of our scheme is 32 + 160 + 1024 + 32 + 32 + 1024 + 32 = 2336 bits and that of Awasthi et al.'s scheme is 32 + 160 + 1024 + 1024 + 32 + 1024 + 32 + 1024 + 32 = 4384 bits. In Table 2, we show the communication costs of Awasthi et al.'s scheme and our scheme.

Table 2. The communication cost of Awasthi et al.'s scheme and our scheme

Awasthi et al.'s scheme	our scheme
4384 bits	2336 bits

For the convenience of evaluating the computation cost, we define some notations as follows.

- T_h :The time of executing a one-way hash function operation.
- T_m :The time of executing a modular multiplication operation.

- T_e :The time of executing a modular exponentiation operation.

In Table 3, we tabulate the computation costs of Awasthi et al.'s scheme and our scheme.

Table 3. The computation cost of Awasthi et al.'s scheme and our scheme

	Awasthi et al.'s scheme	our scheme
Smart Card	$3T_e+1T_m+2T_h$	$1T_e+3T_h$
Server	$3T_e+1T_m+3T_h$	$1T_e+3T_h$

Compared with the computation cost of modular exponentiation operation, the cost of the one-way hash function operation and the modular multiplication operation is ignored. At the same time, the one-way hash function operation and the modular multiplication operation have similar computation cost.

From the above discussion, we know our scheme has much better performance than Awasthi et al.'s scheme. Besides, our scheme has better storage cost and communication cost than Awasthi et al.'s scheme. Moreover, Awasthi et al.'s scheme is vulnerable to a privileged insider attack, a password guessing attack and an impersonation attack, and our scheme could overcome the weaknesses. Then we can conclude that our scheme is more suitable for practical applications.

7. Conclusion

In this paper, we review Awasthi et al.'s RSA-based remote authentication scheme and analyze its security. We show their scheme is vulnerable to three types of attacks by proposing concrete attacks. To enhance security, we proposed an improved RSA-based remote authentication scheme. The security analysis shows our scheme could over weaknesses in Awasthi et al.'s. Moreover, performance analysis shows our scheme also has better performance than their scheme. Therefore, our scheme is more suitable for practical applications. It is easy to say that the proposed scheme needs a true random number generator in order to work. This implies an additional cost for each user. We will solve the problem in the future.

Acknowledgements. The authors wish to thank the anonymous reviewers for their valuable comments. This research was supported by the National Natural Science Foundation of China (No.61202447).

References

[1] Lamport L., *Password Authentication with Insecure Communication*, Communications of the ACM, 1981, Vol. **24**(11), pp. 770–772.

[2] Yang W., Shieh S., *Password authentication scheme with smard cards*, Computers and Security, 1999, Vol **18**(8), pp. 727–733.

[3] Chan C., Cheng L., *Cryptanalysis of timestamp-based password authentication scheme*, Computers and Security, 2002, Vol. **21**(1), pp.74–76.

[4] Fan L., Li J., Zhu H., *An enhancement of timestamp-based password authentication scheme*, Computers and Security, 2002, Vol. **21**(7), pp. 665–667.

[5] Shen J., Lin C., Hwang M., *Security enhancement for the timestamp-based password authentication*, Computers and Security, 2003, Vol. **22**(7), pp. 591–595.

[6] Yoon E., Ryu E., Yoo K., *Attacks on the Shen et al.'s timestamp-based password authentication scheme using smart card*s, IEICE Transactions on Fundamentals, 2005, Vol. **E88-A** (1), pp. 319–321.

[7] Liu J., Zhou A., Gao M., *A new mutual authentication scheme based on nonce and smart cards*, Computer Communications, 2008, Vol. **31**, pp. 2205–2209.

[8] Awasthi A., Srivastava K., Mittal R., *An improved timestamp-based remote user authentication scheme*, Computers and Electrical Engineering, 2011, Vol. **37**(6), pp. 869–874.

[9] Kocher P., Jaffe J., Jun B., *Differential power analysis*, *Proceedings of Advances in Cryptology (CRYPTO 99)*, 1999, pp. 388–397.

[10] Messerges T., Dabbish E., Sloan R., *Examining smart-card security under the threat of power analysis attacks*, IEEE Transactions on Computers, 2002, Vol. **51**(5), pp. 541–552.

[11] He D., Wu S., Chen J., *Note on "Design of improved password authentication and update scheme based on elliptic curve cryptography"*, Mathematical and Computer Modelling, 2012, Vol. **55**(3–4), pp. 1661–1664.