Selective Timestamp-Nonce Based Authentication Scheme

Wei-Chen Wu[1]*, Horng-Twu Liaw[2] and Yi-Ming Chen[3]
[1]*Computer Center, Hsin Sheng College of Medical Care and Management, Taoyuan County, Taiwan, R.O.C.*
wwu@hsc.edu.tw
[2]*Department of Information Management, Shih Hsin University, Taipei, Taiwan, R.O.C.*
htliaw@cc.shu.edu.tw
[3]*Department of Information Management, National Central University, Jhongli City, Taoyuan County, Taiwan, R.O.C.*
cym@cc.ncu.edu.tw

Abstract

In this paper, we improve an efficient and complete remote user authentication scheme and propose an adaptive timestamp-nonce based authentication scheme using portable storage devices. Compared with other smart card-based, timestamp-based and nonce-based schemes, our scheme achieves more functionality. The new importance merits are: An adaptive timestamp-nonce structure is proposed; portable device stores authentication data not only smart card; all transactions through non-secure channel, especially in the registration phase, and batch of portable storage devices is issued. Besides, the basic merits include a dictionary of verification tables is not required to authenticate users, users can choose their password freely, mutual authentication is provided between a user and the remote system, the communication cost and the computational cost are very low, a user can update their password after the registration phase, a session key agreed by a user and the remote system is generated in every session and the serious time synchronization problem are solved.

Keywords:*Authentication; Security; Session Key; Portable Storage Device; Adaptive Timestamp-Nonce.*

1 Introduction

A remote password authentication scheme was proposed by Lamport first over an insecure channel in 1981 [1]. Lately, many schemes using smart cards had been proposed to improve security, functionality and efficiency [2, 3, 4, 5, 6, 7, 8, 9, 10]. Especially, in Liaw et al.'s scheme [9], they summarized the following criteria are important for user authentication schemes.

C1:No verification table: No verification or password table is stored in the remote system.

C2:Freely chosen password: Users can choose their password freely.

C3:Mutual authentication: Whether the users and the remote system can authenticate

each other.
C4:Lower communication and computation cost: Due to hardware constraints of a smart card, it usually does not support power communication cost and higher bandwidth.
C5:Updated password: Users can update their passwords after the registration phase.
C6:Session key agreement: A session key agreed by a user and the remote system generated in every session.
C7: Time synchronization: Discard the timestamp to solve the serious time synchronization problem.

However, most secure authentication information is through secure channel. Due to computer systems and their interconnections via networks or internet have increased, it is unreasonable over a closed network system. For this reason, we propose the authentication scheme through opened channel or internet to exchange data in all transactions. Especially, in the registration phase all data are transmitted to another through opened channel or internet. Hence, a new criterion will be referred: All transactions through non-secure channel.

Since then, several mechanisms have been proposed to achieve more functionality and efficiency [2, 3, 4, 5, 6, 7, 8, 9, 10]. They claimed that their schemes have the merits of providing mutual authentication, freely choosing password, no verification table, updating their password after the registration phase, generating a session key agreed by a user and the remote system in every session, solving he serious time synchronization problem and involving only few hashing operations. To resist replay attacks, several timestamp-based schemes were used [4, 6, 8, 10]. Unfortunately, such schemes can lead to serious clock synchronization problems. Moreover, to avoid these synchronization problems, many nonce-based schemes were proposed without timestamps [2, 3, 5, 7, 9]. However, they need a user table to store used nonce values for freshness checking. In this paper, we propose an adaptive timestamp-nonce based to solve these problems. Hence, a new criterion will be referred: Adaptive Timestamp-Nonce.

Most proposed schemes store secure data in smart card [2, 3, 4, 5, 6, 7, 8, 9, 10]. It is not convenient to issue and revoke a smart card. A user lost his/her a smart card or is stolen that brings he/she must register again. Thus, our scheme proposes that a user just registers the remote system once and then saves a backup for registration information on any storage devices. Suppose a storage device is lost or stolen, a user can recover backup data into a new storage device. As regards the smart card, it is difficult because the smart card perform specialized software and hardware. Furthermore, due to there are many mobile devices today, a user needs to carry a smart card reader everywhere. If people login the remote system using mobile computer, it is not suitable for use. Therefore, we propose the authentication scheme using portable storage devices such as USB flash drives, iPods, PDAs or disk. Hence, a new criterion will be referred: Portable device stores authentication data.

Suppose some mobile devices or USB flash drives may make mass production by manufacturer. Therefore, a method for a batch of storage devices is issued. If the remote system wants to issue many storage devices, the remote system can set default passwords for them and send a batch of these devices to many users. Hence, a new criterion will be referred: Batch of storage devices is issued.

As mentioned, either Liaw et al. [9] summarized the criteria are important or the following criteria also must be satisfied:
C1:Adaptive Timestamp-Nonce: An ATN does not only have a user table but also no

serious time synchronization problem and avoid replay attack.
C2:Portable device stores authentication data: Not only smart cards but also any storage devices are supported in our proposed scheme.
C3:All transactions through non-secure channel: Especially, in the registration phase all data are transmitted to another through insecure channel or internet.
C4:Batch of storage devices is issued: If the remote system wants to issue many storage devices, the remote system can set default passwords for them and send a batch of these devices to many users.

The current scheme integrates all the advantages proposed by the previous schemes. In Section 2 , we propose an adaptive timestamp-nonce based authentication scheme using portable storage devices. This scheme has several merits: The remote system does not need a dictionary of verification tables to authenticate users; a user can choose their password freely; mutual authentication is provided between a user and the remote system; the communication cost and the computational cost are very low; a user can update their password after the registration phase; a session key agreed by a user and the remote system can be generated in every session; the serious time synchronization problem is avoid; a user table is not be used; all transactions are through non-secure channel; batch of storage devices is issued and portable device stores authentication data. In Section 3, we examine the security. In Section , we evaluate the efficiency of our scheme and a comparison is given. Finally, we conclude this paper in Section 5.

2 Our scheme

First, we propose a new Adaptive Timestamp-Nonce (ATN) structure. See the Fig. 1. For security issues, some schemes [4, 6, 8, 10] proposed timestamp and some schemes [2, 3, 5, 7, 9] proposed nonce to avoid replay attack. But, both of them have some advantages and disadvantages depend on which authentication environment.

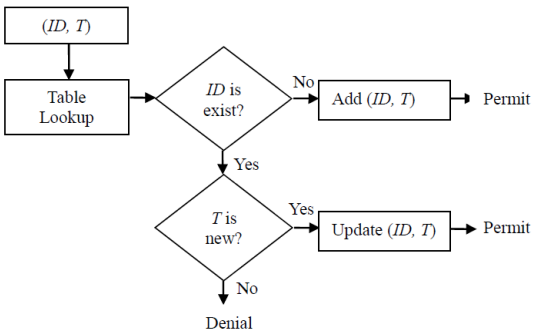

Figure 1. The structure of an Adaptive Timestamp-Nonce

Timestamp: A time sequence is generated according to time currently.

- Advantages: Avoid replay attack and no need a user table.
- Disadvantages: Incur the serious time synchronization problem that the client's time and the server's time must be the same or must differ in a reasonably small range.

Note that if the client and server's clocks are not well synchronized, or the transmission delay is too long and unpredictable in a network environment.

Nonce: A random number is generated and the value that not be used before.

- Advantages: Avoid replay attack and no serious time synchronization problem.
- Disadvantages: Need to create a user table to store used nonce value for freshness checking.

For this reason, we shall propose an ATN structure. The ATN is a compromise between Timestamp and Nonce. Suppose T denote the Timestamp-Nonce. We can tune ΔT range depend on which authentication environment. When ΔT is short, it is similar to Timestamp scheme. Otherwise, when ΔT is long, it is similar to Nonce scheme. Hence, an ATN would not only no user table but also no serious time synchronization problem.

In this section, we propose an improved remote user authentication scheme using portable storage devices. Let x be a secret key maintained by the remote system, S denotes the remote system, U_i denote the ith user who submits his public identity ID_i and his secret password PW_i to the remote system, "$X \rightarrow Y : Z$" denotes a sender X sending a message Z to a receiver Y. Also, "$\|$" denotes the conventional string concatenation operator, $h(\bullet)$ denote a secure one-way hash function with fixed-length output and Storage[$\bullet$] denote a portable storage device saves some information in its memory. Our schemes are divided into six phases: (1) the registration phase, (2) the login phase, (3) the verification phase, (4) the session phase, (5) the updated password phase and (6) batch of storage devices is issued. In the registration phase, a new user chooses his/her own ID and PW. In the login phase, when a legitimate user wants to log into the servers, he/she must enter the ID and PW and generate related arguments to the remote system for verification. Then the remote system will verify the legitimacy and access right of the user in the verification phase. We demonstrate our scheme as follows:

2.1 Registration phase

When a new user wants to log into the remote system, a user must register with the remote system and perform the following steps:

- $U_i \rightarrow S : ID_i, h(PW_i, T_r), T_r$
  A new user U_i inputs his/her own ID_i and PW_i, and then generates a timestamp-nonce T_r, and sends $ID_i, h(PW_i, T_r), T_r$ to the S for registration through insecure channel or Internet.
- $S \rightarrow ATN : ID_i, T_r$
  The S shall deliver ID_i, T_r to an Adaptive Timestamp-Nonce to check whether the time interval between T_r and T_r' is greater than ΔT_r. If $(T_r' - T_r) \geq \Delta T_r, S$ will reject the registration request. If accepted, S checks the correctness of ID_i. If the format of ID_i is incorrect, S will also reject the registration request.
- $S \rightarrow U_i : Storage[h(\bullet), e_i]$
  Once the S receives ID_i and $h(PW_i, T_r)$, S computes U_i's secret information $v_i = h(ID_i, x)$ and $e_i = v_i \oplus h(PW_i, T_r)$. Then writes $h(\bullet)$ and e_i into the memory of a portable storage device and issue the device to U_i.

2.2 Login phase

When U_i wishes to login the remote system. He must insert the portable storage device into the terminal or personal computer and type his identity ID_i and his password PW_i in the jth login for a user.

- $U_i \rightarrow S : ID_i, C_{i,j}, T_{i,j}$
  U_i then generates a timestamp-nonce $T_{i,j}$ and use T_r, and computes $C_{i,j} = h(e_i \oplus h(PW_i, T_r), T_{i,j})$. And then sends the message $ID_i, C_{i,j}, T_{i,j}$ to the remote system.

2.3 Verification phase

After receiving the authentication request message $ID_i, C_{i,j}, T_{i,j}$, the remote system and a user execute the following steps to facilitate a mutual authentication between a user and the remote system in the login phase at $T_{i,j}$.

- $S \rightarrow ATN : ID_i, T_{i,j}$
  The S shall deliver $ID_i, T_{i,j}$ to an Adaptive Timestamp-Nonce to check whether the time interval between $T_{i,j}$ and $T'_{i,j}$ is greater than $\Delta T_{i,j}$. If $(T'_{i,j} - T_{i,j}) \geq \Delta T_{i,j}, S$ will reject the login request. If accepted, S checks the correctness of ID_i. If the format of ID_i is incorrect, S will also reject the login request.
- $S \rightarrow U_i : Reject\ Request\ or\ (msg)$
  The S computes $v'_i = h(ID_i, x)$ and check whether $C_{i,j} = h(v'_i, T_{i,j})$. If not, the request is rejected; otherwise, generates a timestamp-nonce N_s and encrypts the message $msg = E_{v_i}(T_{i,j}, N_s)$ and send back it to the user.
- $U_i \rightarrow S : Reject\ Services\ or\ (N'_s)$

- $S \rightarrow U_i : Accept\ or\ Reject$
  After receiving the message msg, U_i decrypts the message $D_{e_i \oplus h(PW_i, T_r)}(msg)$ to derive $(T'_{i,j}, N'_s)$ and verify whether $T'_{i,j} = T_{i,j}$. If yes, N'_s is sent to S. If not, the connection is disconnected. Finally, checks whether $N'_s = N_s$. If yes, the mutual authentication is done.

2.4 Session phase

It is used a common session key $C_{i,j}$ is generated in this protocol to encrypt individual conversation between the client and the remote system within a session after the jth login. The following operations are performed.

- $S \rightarrow U_i : E_{e'_i \oplus v_i}(M_s \oplus C_{i,j})$
  If S wants to send private data or message M_s to U_i , it encrypts message $E_{e'_i \oplus v_i}(M_s \oplus C_{i,j})$ with e_i and v_i, and sends it to U_i. After U_i receives the message, the U_i decrypts the message with PW_i and T_r, and makes an exclusive operation using session key $C_{i,j}$ to derive M_s.
- $U_i \rightarrow S : E_{h(PW_i, T_r)}(Mu \oplus C_{i,j})$
  If U_i wants to send private data or message M_u to S, it encrypts message $E_{h(PW_i, T_r)}(M_u \oplus C_{i,j})$ and send it to S. After the S receives the message, it decrypts the message with e_i and v_i and makes an exclusive operation using session key $C_{i,j}$ to derive M_u.

2.5 Updated password phase

If U_i wants to change his password from PW_i into PW_i' after registration, the following procedure is performed.

- Generate a new timestamp-nonce T_r'.
- Input old PW_i and new PW_i'.
- Calculate $e_i' = e_i \oplus h(PW_i, T_r) \oplus h(PW_i', T_r')$.
- Update e_i on the memory of portable storage device to set e_i'. That is because:

$$\begin{aligned} e_i' &= v_i \oplus h(PW_i', T_r') \\ &= h(ID_i, x) \oplus h(PW_i', T_r') \\ &= e_i \oplus h(PW_i, T_r) \oplus h(PW_i', T_r') \end{aligned}$$

Our scheme just proposes a function that provides to change password off-line. The practical implement need an additional method to achieve updated password off-line. For instance, use double columns to type two new passwords and check whether the same for spelling check.

2.6 Batch of storage devices is issued

If S wants to issue many storage devices to many U_i, the following steps are performed.

- Set a default PW_i for each U_i.
- Implement registration phase for each U_i.
- Batch and issue portable storage device to U_i.
- Implement updated password phase by U_i chooses individually.

3 Security analysis

In this section, we analyze the security of our scheme. The strength of our scheme can be demonstrated as follows:

3.1 Impersonate attack

The finder of the lost portable storage device may not impersonate to compute session key $C_{i,j} = h(e_i \oplus PW_i, T_{i,j})$ when not know PW_i of U_i and $T_{i,j}$ is fresh that not be used before, even e_i is gained by finder in the lost portable storage device. When a user loses the portable storage device, the intruder can not update PW_i because he dose not know the owner's PW_i. A secret x of the remote system can not be obtained by anyone; it is infeasible to because the one-way property of $v_i = h(ID_i, x)$. Suppose the portable storage device is stolen and e_i is gained by finder, no one can derive x, which is protected by the one-way hash function. The masqueraded the remote system cannot succeed because an attacker cannot compute $msg = E_{v_i}(T_{i,j}, N_s)$ unless he knows the secret v_i, which then lead to $T_{i,j}' \neq T_{i,j}$ and $N_s' \neq N_s$.

3.2 Replay attack

In the login phase, an old login message was eavesdropped by an attacker. He may try to replay the old login message $(ID_i, C, T'_{i,j})$ to the remote system for a new login request. Verifying whether $C_{i,j} = h(v'_i, T'_{i,j})$ may fail because $T'_{i,j}$ is not always the same every time and it is fresh that not be used before. The login request thus cannot succeed. Similarly in the verification phase, when an attacker eavesdropped an old verification message, he may try to replay the old verification message $msg = E_{v_i}(T_{i,j}, N_s)$ to the user for verification. The verification of $T'_{i,j} = T_{i,j}$ and $N'_s = N_s$ can not succeed because $T'_{i,j}$ and N'_s are timestamps or nonce and used only once.

3.3 Eavesdropping attack

An attacker who eavesdropped $E_{e'_i \oplus v_i}(M_s \oplus C_{i,j})$ and $E_{PW_i}(M_u \oplus C_{i,j})$ can not decrypt the message and make an exclusive operation to derive M_s and M_u because v_i, PW_i and session key $C_{i,j}$ can never be found. An attacker cannot steal $T_{i,j}$ and N_s both in the same time to accumulate plaintext/ciphertext pairs $(T_{i,j}, N_s)/E_{v_i}(T_{i,j}, N_s)$ and mount a known-ciphertext attack on v_i because $T_{i,j}$ appear during login phase and N_s will appear during verification phase if $T'_{i,j} = T_{i,j}$. Besides, nonce is generated and the value that not be used before.

No one can forge a valid $C_{i,j} = h(e_i \oplus PW_i, T_{i,j})$ because it must be derived from PW_i. On the other hand, given a valid $C_{i,j} = h(e_i \oplus PWi, T_{i,j})$, no one can compute PW_i because $h(\bullet)$ is a one-way hash function. Even given some valid request messages $(ID_i, C_{i,j}, T_{i,j})$, the attacker has no way to derive another valid message because of the one-way property of the secure one-way hash function.

3.4 Stolen verifier attack

Due to the proposed scheme does not store the verification table in the remote system, the attack may not be held.

3.5 Denial of Service attack

Due to the remote system has not any verification table, it is hard for the attacker to make failures by any modification.

3.6 Man-in-the-middle attack

Due to the verification phase has been done, any illegal users or the remote system will cannot establish this session and the session key are always changed because nonce are freshness.

4 Efficiency

In this section, we summarize performance and criteria for authentication schemes. As a protection mechanism on user authentication, the following criteria are crucial.
C1:Adaptive Timestamp-Nonce: An ATN does not only have a user table but also no

Table 1: Give comparisons among the smart card-based schemes.

	Our proposed scheme	**Timestamp Based** [3, 5, 8, 10]	**Nonce Based** [1-2, 4, 7, 9]
C1	Yes: Propose ATN Structure	No: Serious time synchronization problem	No: have a user table
C2	Yes: Use storage devices such as USB flash drives, iPods, PDAs or disk.	No: Use smart card	No: Use smart card
C3	Yes: Transmit to another through insecure channel or internet	No: Perform over a secure channel in the registration.	No: Perform over a secure channel in the registration.
C4	Yes: Use storage devices	No: Cannot support	No: Cannot support
C5	Yes	Yes	Partial Yes; [2] cannot support
C6	Yes	Yes	Yes
C7	Yes	Yes	Yes
C8	Yes	Yes	Yes
C9	Yes	Yes	Partial Yes; [1-2] not specify
C10	Yes	Partial Yes; [5,8,10] not specify	Partial Yes; [7] not specify

serious time synchronization problem and avoid replay attack.
C2:Portable device stores authentication data: Not only smart cards but also any storage devices are supported in our proposed scheme.
C3:All transactions through non-secure channel: Especially, in the registration phase all data are transmitted to another through insecure channel or internet.
C4:Batch of storage devices is issued: If the remote system wants to issue many storage devices, the remote system can set default passwords for them and send a batch of these devices to many users.
C5:No verification table: No verification or password table is stored in the remote system.
C6:Freely chosen password: Users can choose their password freely.
C7:Mutual authentication: Whether the users and the remote system can authenticate each other.
C8:Lower communication and computation cost: Due to hardware constraints of a smart card, it usually does not support power communication cost and higher bandwidth.
C9:Updated password: Users can update their passwords after the registration phase.
C10:Session key agreement: A session key agreed by a user and the remote system generated in every session.

In this section, the efficiency of our scheme is examined and compares with other remote authentication schemes, Chen et al.'s scheme [2], Chien et al.'s scheme [3, 4], Juang's scheme [5], Ku et al.'s scheme [6], Lee et al.'s scheme [7, 8], Liaw et al.'s scheme [9] and Yoon et al.'s scheme [10], known as secure so far. The schemes [4, 6, 8, 10] are timestamp based and the schemes [2, 3, 5, 7, 9] are nonce based. Thus, as shown in Table 1, our scheme is much more efficient than timestamp and nonce based schemes. Moreover, the scheme [3] has verification or password table is stored in the remote system, the schemes [2, 3] do not specify how users change their passwords and the schemes [6, 7, 8, 10] do not specify session

key agreement. Thus, we give comparisons among the previous smart card schemes and our proposed scheme. Table 1 is illustrated to show that our scheme satisfies all criteria. Table 2 gives several comparisons among various methods in the registration, login, verification, session and updated password phases.

Table 2: Comparisons of computation costs.

	Registration	**Login**	**Verification**	**Session**	**Updated Password**	**Batch of storage devices is issued**
Our scheme	$T(f)$ $T(\oplus)$ $T(r)$	$T(f)$ $T(\oplus)$ $T(r)$	$3T(f)$ $T(\oplus)$ $2T(S)$ $T(r)$	$T(f)$ $3T(\oplus)$ $2T(S)$	$2T(f)$ $2T(\oplus)$ $T(r)$	n multiplied by costs of (Registration + Update password) , where n is amount of one batch
Chen et al. [1]	$T(f)$ $T(\oplus)$ $T(r)$	$2T(f)$ $3T(\oplus)$ $T(r)$	$14T(f)$ $9T(\oplus)$ $10T(\|\|)$ $T(r)$		Not specify	Not support
Chien et al. [2]	$4T(f)$ $T(\oplus)$ $T(r)$ $2T(\|\|)$	$7T(f)$ $4T(\oplus)$ $T(r)$ $3T(\|\|)$	$8T(f)$ $4T(\oplus)$ $T(\|\|)$		Not specify	Not support
Chien et al. [3]	$T(f)$ $2T(\oplus)$ z	$T(f)$ $2T(\oplus)$ $T(r)$	$4T(f)$ $4T(\oplus)$ $T(r)$	Not specify	Not specify	Not support
Juang [4]	$T(f)$ $T(\oplus)$	$T(f)$ $T(\oplus)$ $T(S)$ $2T(r)$ $T(\|\|)$	$3T(f)$ $4T(S)$ $4T(r)$ $T(\|\|)$		Not specify	Not support
Ku et al. [5]	$2T(f)$ $3T(\oplus)$ $T(\|\|)$	$2T(f)$ $3T(\oplus)$ $T(r)$	$5T(f)$ $5T(\oplus)$ $T(r)$	Not specify	$2T(f)$ $4T(\oplus)$	Not support
Lee et al. [7]	$T(f)$ $2T(\oplus)$	$2T(\oplus)$ $T(\|\|)$	$5T(f)$ $5T(\oplus)$ $T(r)$	Not specify	$2T(\oplus)$	Not support
Lee et al. [8]	$T(f)$ $2T(\oplus)$	$T(f)$ $2T(\oplus)$ $T(\|\|)$	$6T(f)$ $4T(\oplus)$ $T(r)$	Not specify	$2T(\oplus)$	Not support
Liaw et al. [9]	$T(f)$ $T(\oplus)$	$T(f)$ $T(\oplus)$ $T(r)$	$2T(f)$ $2T(S)$ $T(\oplus)$ $T(r)$	$2T(ME)$ $2T(S)$ $T(\oplus)$	$2T(\oplus)$	Not support
Yoon et al. [10]	$4T(f)$ $5T(\oplus)$ $T(r)$ $T(\|\|)$	$2T(f)$ $3T(\oplus)$ $T(r)$	$5T(f)$ $5T(\oplus)$ $T(r)$	Not specify	$2T(f)$ $4T(\oplus)$	Not support

$T(f)$ is computation cost of one-way function; $T(\oplus)$ is computation cost of Exclusive-OR operation; $T(S)$ is computation cost of symmetric encryption; $T(||)$ is computation cost of string concatenation; $T(r)$ is computation cost of random number, nonce and timestamp generation; $T(ME)$ is computation cost of modular exponentiation.

5 Conclusion

In this paper, we have proposed an adaptive timestamp-nonce based authentication scheme using portable storage devices. Besides the basic merits, the new importance merits are: An adaptive timestamp-nonce structure is proposed; portable device stores authentication data not only smart card; all transactions through non-secure channel, especially in the registration phase, and batch of portable storage devices is issued. Compared with other smart card-based schemes, our proposed scheme achieves more functionality and satisfies all criteria.

References

[1] Lamport, L.: Password authentication with insecure communication. Communications of the ACM **24** (1981) 770–772

[2] Chen, Y., Yeh, L.: An efficient nonce-based authentication scheme with key agreement. Applied Mathematics and Computation **169** (2005) 982–994

[3] Chien, H., Jan, J.: Robust and simple authentication protocol. The Computer Journal **46** (2003) 193

[4] Chien, H., Jan, J., Tseng, Y.: An efficient and practical solution to remote authentication: smart card. Computers & Security **21** (2002) 372–375

[5] Juang, W.: Efficient password authenticated key agreement using smart cards. Computers & Security **23** (2004) 167–173

[6] Ku, W., Chen, S.: Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. Consumer Electronics, IEEE Transactions on **50** (2004) 204–207

[7] Lee, S., Kim, H., Yoo, K.: Efficient nonce-based remote user authentication scheme using smart cards. Applied mathematics and computation **167** (2005) 355–361

[8] Lee, S., Kim, H., Yoo, K.: Improved efficient remote user authentication scheme using smart cards. Consumer Electronics, IEEE Transactions on **50** (2004) 565–567

[9] Liaw, H., Lin, J., Wu, W.: An efficient and complete remote user authentication scheme using smart cards. Mathematical and Computer Modelling **44** (2006) 223–228

[10] Yoon, E., Ryu, E., Yoo, K.: Further improvement of an efficient password based remote user authentication scheme using smart cards. Consumer Electronics, IEEE Transactions on **50** (2004) 612–614