Cent. Eur. J. Comp. Sci.
DOI: 10.2478/s13537-012-0003-y

VERSITA

Central European Journal of **Computer Science**

Cryptanalysis and security enhancement of Chen *et al.*'s remote user authentication scheme using smart card

Research Article

Saru Kumari[1]*, Mridul K. Gupta[2], Manoj Kumar[3]

1 *Department of Mathematics, Agra College, Agra-282004, Uttar Pradesh, India*

2 *Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh, India*

3 *Department of Mathematics, R. K. College, Shamli (Muza arnagar), Uttar Pradesh, India*

Received 23 June 2011; accepted 27 February 2012

Abstract: Recently (2011) Chen *et al.* found that Wang *et al.*'s scheme (2007) is vulnerable to impersonation attacks and parallel session attacks; and then proposed a security enhancement of Wang *et al.*'s scheme. Chen *et al.* claimed to inherit the merits and eradicate the aws of the original scheme through their improved scheme. Unfortunately, we found that Chen *et al.*'s scheme inherits some aws of the original scheme, like the known-key attack, smart card loss attack and its serious consequences. In addition, Chen *et al.*'s scheme is not easily reparable and is unable to provide forward secrecy. Thus Chen *et al.*'s scheme still has scope for security enhancement. Finally, we propose an improved scheme with better security strength. Moreover, we analyze the performance of our scheme and prove that ours is suitable for applications with high security requirements.

Keywords: remote user mutual authentication cryptanalysis smart card session key

Versita Sp. z o.o.

1. Introduction

In times gone by, authentication was not a complex task. Enter the computer era, where one cannot see the entity on the remote end of a computer network, and indeed the entity could be a friend or an attacker; authentication is quite a complex task. Authentication is the process of verifying that someone or something is who or what it claims to be. Authentication is usually based on one or more of the following factors:

What you know: It means someone mentally possesses something. It only verifies that someone knows something. It is characterized by secrecy. Example: a password or a PIN

* *E-mail: saryusiirohi@gmail.com*

What you have: It is any form of issued or acquired self-identification token. It is characterized by physical possession. Example: a smart card or physical keys.

What you are/do: It is a biometric related characteristic unique to an individual. Example: fingerprint, iris or voiceprint.

A single password is an excellent authenticator. Its secrecy is a good defence against theft. Furthermore, it is convenient and inexpensive. But a memorable password can often be guessed or searched by an attacker; and a long, random, changing password is difficult to remember. A better authentication method is to require that the user provides a physical object as a "smart card" along with a password. If the smart card is lost or stolen, the user is likely to know about it and act accordingly. Moreover it is password protected and hence, cannot be activated without knowing the password. Smart cards are generally tamper resistant hence, are relatively difficult for a hacker to compromise. A card alone cannot be used as an authenticator because it is completely insecure if lost or stolen. Biometric authenticators offer several advantages over other authenticators. They cannot be lost or forgotten; they are difficult to forge or copy. However, if a biometric is compromised, it is not as easily replaceable as passwords or smart cards.
Thus, none of these methods is entirely foolproof. But properly implemented in combination of one or more can provide secure remote user authentication. Use of a single factor, even if multiple pieces of evidence are offered, is considered to be weak authentication. A combination of two or (more) factors, such as a password and a smart card and/or biometric, is called *two-factor* (or *multi-factor*) *authentication*, is considered to be strong authentication.
With the large scale of proliferation of the Internet and network technologies, smart card-based authentication schemes have been widely deployed to verify the legitimacy of remote user's login request. In remote authentication process, a remote server authenticates a registered user based on his secret credentials. In traditional schemes, the server has to store a password table to save passwords of all the registered users. In 1981, Lamport proposed a remote user authentication scheme requiring a password verification table to be stored on the server's side [24]. Lamport claimed that his scheme is secure even if an adversary eavesdrops on the communication between a user and the server. Lamport's scheme has played an important role in the development of remote user authentication schemes. In 1990, Shimizu proposed the so called CINON protocol [36], to solve the drawbacks of high hash overhead and password resetting of Lamport's scheme. Later, in 1998, Shimizu proposed their PERM (Privacy Enhanced Information Reading and Writing Management) protocol [37] to solve the random number memorizing problem of the CINON protocol. In 1995, Horng and later, in 1998, Jan *et al.*, proposed solutions to prevent password tables from being stolen or modified by attackers, where the password table is no longer required to be kept by the server [12, 15]. In 1995, inspired by Lamport's method, Haller developed the famous S/KEY one-time password for the Internet draft RFC 1760 [11]. Later on, S/KEY authentication scheme was found to be flawed for replay, server impersonation and password guessing attacks [32, 43]. In 2000, Hwang and Li [14] identified that Lamport's scheme is susceptible to the risks of hacking and modifying the password table. Thus Hwang and Li proposed a remote user authentication scheme without using the password table, which was based on El Gamal public key encryption method [7]. Until now, there have been ample of remote user authentication schemes published in the literature and each published scheme has its own merits and demerits [2, 4, 5, 8, 10, 13, 16-18, 20-23, 25-27, 29, 30, 33, 35, 38, 39, 41, 42, 44].
In 2000, Sun proposed an efficient password-based remote user authentication scheme using smart cards [38]. Sun's scheme requires only several hash operations instead of the costly modular exponentiation. However, Sun's scheme does not provide mutual authentications. In 2002, Chien *et al.* [4] proposed password-based remote user authentication scheme, and claimed that their scheme provides mutual authentication, free choice of passwords, no verification table, and involves only a few hashing operations. Later, Ku-Chen [22] showed that Chien *et al.*'s scheme is vulnerable to a reflection attack [33] or an insider attack [22] and is not reparable [13]. Ku-Chen also gave an improved scheme [22] to prohibit the weaknesses of Chien *et al.*'s scheme. But Yoon *et al.* [44] showed that the improved scheme [22] is still susceptible to a parallel session attack and is insecure when changing the user's password in the password changing phase. Hence, Yoon *et al.* [44] presented an enhancement to resolve such problems. In 2007, Wang *et al.* [40] showed that both Ku *et al.*'s and Yoon *et al.*'s schemes are still vulnerable to the guessing attack, forgery attack and denial of service (DoS) attack. To remedy these flaws, they proposed an efficient improvement over Ku *et al.*'s and Yoon *et al.*'s schemes with more security. However Chen *et al.* [3] found that Wang *et al.*'s scheme is still vulnerable to the impersonation attack [1] and parallel session attack [22]. To resolve these flaws they proposed an improved approach [3] over Wang *et al.*'s scheme; which they claimed to fulfil the requirements of security, session key agreement, mutual authentication and perfect forward secrecy.
Unfortunately, we find that Chen *et al.*'s scheme is vulnerable to the malicious user attack, known key attack, smart card loss attack and it's serious consequences such as the impersonation attack, password guessing attack, man-in-the

middle attack. Further, Chen *et al.*'s scheme lacks perfect forward secrecy, generates unsecure session keys, suffers from poor reparability and has no provision for revocation of lost smart cards. Undoubtedly, Chen *et al.*'s improved scheme rectifies the pitfalls pointed out in Wang *et al.*'s scheme, but it still inherits many flaws of the original scheme. Therefore, we propose another enhanced protocol to overcome all the identified security flaws of Chen et al.'s scheme.
The rest of this paper is organized as follows. In Section 2, a brief review of Chen *et al.*'s scheme is given. Section 3, describes cryptanalysis of Chen *et al.*'s scheme. Our enhanced scheme is proposed in Section 4. The security analysis and usability analysis of the proposed scheme is discussed in Section 5. The comparison of the cost and functionality of the proposed scheme with Wang *et al.*'s scheme and Chen *et al.*'s scheme is shown in Section 6. Finally, some concluding remarks are included in the last section.

2. Review of Chen *et al.*'s scheme

The notations used throughout this paper are summarized as follows:

U: the user	$\Rightarrow$: a secure channel
ID: the identity of U.	$\rightarrow$: a common channel.
PW: the password of U.	$\oplus$: the bitwise XOR operation.
FP: fingerprint of U.	h(.): a cryptographic hash function.
SC: the smart card of U.	h_P (.): a cryptographic keyed hash function which includes secret code p.
U_A: the attacker.	h_K (.): h(.) when operated using secret key K shared between U and S.
S: the remote server.	δt: the maximum time interval for communication delay.
x: permanent secret key of S.	T_U: current timestamp of U.
IDS: the identity of S known only to S.	T_S: current timestamp of S.
ADB: the account-database maintained by S.	T_A: current timestamp of U_A.
CS: a secret code shared between U and S.	$\|\|$: the string concatenation.
CIUS: combined identity of U and S.	

In this section we briefly review Chen *et al.*'s scheme as follows:

2.1. Registration phase

This phase is invoked whenever U initially registers to S. The following steps are involved in this phase.

1. U selects a random number b and computes h(b $\oplus$ PW).
2. U $\Rightarrow$S: ID, h(b $\oplus$ PW).
3. S performs the following computations: P = h(ID $\oplus$ x), R = P $\oplus$ h(b $\oplus$ PW), V = h_P(h(b $\oplus$ PW)).
4. S $\Rightarrow$ U: SC = {V, R, h(.),h_P(.)}.
5. U enters b into his SC, now SC = {V, R, b, h(.), h_P(.)}.

2.2. Login phase

Whenever U wants to login to S, the following operations will perform:

1. U inserts his SC into the smart card reader and then enters ID and PW.
2. SC computes: P = R $\oplus$ h(b $\oplus$ PW) and checks whether h_P(h(b $\oplus$ PW))? = V holds or not. If not, SC terminates this session.
3. SC generates a random number r and computes C_1 = P $\oplus$ h(r $\oplus$ b), C_2 = h_P(h(r $\oplus$ b) || TU).
4. U $\rightarrow$ S: {ID, C_1, C_2, T_U}

2.3. Verification phase

After the login request is received by S, following steps are performed by S and U:

1. If ID format is incorrect or (T_S -T_U) $\geq \delta t$, S drops the login request; otherwise.
2. S computes P = h(ID $\oplus$ x), C_1^* = P $\oplus$ C_1, C_2^* = h_P(C_1^* || T_U).
3. If C_2^* = C_2, S accepts U's login requests and computes C_3 = h_P(C_1^* $\oplus$ TS || P). Otherwise, S rejects U's login requests.
4. U →S: {T_S, C_3}.
5. If T_S is fresh, U computes C_3^* = h_P(h(r $\oplus$ b) $\oplus$ T_S || P) and checks if C_3^* ? = C_3. The equivalence authenticates S; otherwise U terminates this session.
6. U and S agree on common $K_S ESS$ = h(r $\oplus$b) for the subsequent private communication.

2.4. Password change phase

This procedure is invoked whenever U wants to change his password PW with a new one.

1. U inserts his SC into the smart card reader, inserts ID and PW, and requests to change password.
2. SC computes P* = R $\oplus$ h(b $\oplus$ PW) and V* = h_P*(h(b $\oplus$ PW)); checks whether V*? = V.
3. If they are equal, then U selects new password PW_{NEW}. Otherwise S_C rejects the password change request.
4. SC computes: R_{NEW} = P* $\oplus$ h(b $\oplus$ PW_{NEW}), V_{NEW} = h_P*(h(b $\oplus$ PW_{NEW})).
5. SC replaces R, V with R_{NEW} and V_{NEW} respectively. Now the new password is successfully updated.

3. Cryptanalysis of Chen *et al.*'s scheme

3.1. The known-key attack

A good definition of the known-key attack was introduced in [34]. Given two entities A and B, if one of their previous session keys or one of their previous instances of plaintexts is known to U_A, the future session keys and plaintexts cannot be compromised. It is a basic security requirement of a key agreement scheme.
In Step-6 of the verification phase, if a used session key C_1^* = h(r $\oplus$ b) is compromised by U_A, then U_A can easily employ it to derive the secret information P = h(ID $\oplus$ x) by calculating C_1^* $\oplus$ C_1 in Step-3 of the login phase. Note that U_A can monitor all communicated messages between U and S. Then U_A can select two random numbers r_A and b_A and calculate C_{1A} = P $\oplus$ h(r_A $\oplus$ b_A) and C_{2A} = h_P(h(r_A $\oplus$ b_A) || T_A). Then U_A sends a login request {ID, C_{1A}, C_{2A}, T_A} to S. By performing the verification phase, U_A can establish a new session key C_1^* = h(r_A ï>≈ b_A) with S and achieve the purpose of the cheat.

3.2. Smart card loss attack

Suppose SC of U is stolen/found by U_A and the stored secret values V, R, b, h(.) and h_P(.) were extracted through the studies [19,31]. With this information, U_A can guess a password PW* and calculates h(b $\oplus$ PW*). Then U_A can calculate P* = R $\oplus$ h(b $\oplus$ PW*) and V* = h_{P*}(h(b $\oplus$ PW*)). If V* = V, it indicates that U_A did guess the correct password and correct secret information, i.e. PW* = PW and P* = P. Thus, with smart card loss attack U_A is able to mount:

- offline password guessing attack, or
- offline secret information P = (ID $\oplus$ x) guessing attack.

3.3. Tracing ID corresponding to the stolen/lost SC

Suppose an intelligent attacker U_A has maintained the record of various login requests. And as explained in section 3.2., from U's stolen/lost SC, U_A possess the correctly guessed password PW and secret information $P = (ID \oplus x)$. Now U_A randomly picks a login request $\{ID, C_1, C_2, T_U\}$ from the record; computes $C_1^* = C_1 \oplus P$ and $C_2^* = h_P(C_1^* \,||\, T_U)$. If $C_2^* = C_2$, it implies that the stolen/found SC and this login request belong to the same user. Otherwise U_A repeats the same procedure with another login request from the record. In this way, if U_A is successful then he now possesses ID, PW and P corresponding to U.

3.4. Denial of service attack

Once U_A actually guesses the correct password PW* and successfully traces the corresponding ID of U as described in Sections 3.2 and 3.3 respectively, he can change the password of U by applying the following steps:

1. U_A inserts the stolen/found SC into the card reader, keys ID and PW of U, and requests to change password.
2. SC computes $P^* = R \oplus h(b \oplus PW^*)$, $V^* = h_P^*(h(b \oplus PW^*))$ and checks whether $V^* ?= V$, obviously, both of them are the same, and this smart card will accept the password change request.
3. U_A inputs a new password PW_{NEW}. Then the SC computes: $R_{NEW} = P^* \oplus h(b \oplus PW_{NEW})$, $V_{NEW} = h_P^*(h(b \oplus PW_{NEW}))$.
4. SC replaces R, V with R_{NEW} and V_{NEW} respectively. Now the new password is successfully updated. Now U_A replaces the SC of U. After that, the registered legal user U cannot make any valid login requests since his old password will not work anymore. Consequently U will have to face the denial of service.

3.5. User impersonation attack

As explained above, if U_A is successful in obtaining ID, PW and P corresponding to U and possesses the SC of U, then obviously he can impersonate U and S to make fool of both to access the services provided by S.
But even if U_A replaces the stolen smart card of U, he can impersonate U to fool S as he knows ID, PW, b, $P = h(ID \oplus x)$, h(.) and $h_P(.)$; in the following manner:

1. U_A selects a random number r_A and computes $C_{1A} = P \oplus h(r_A \oplus b)$ and $C_{2A} = h_P(h(r_A \oplus b) \,||\, T_A)$ and sends ID, C_{1A}, C_{2A}, T_A to S.
2. Since ID is valid and TA is fresh, S will proceed to compute $P = h(ID \oplus x)$, $C_{1A}^* = P \oplus C_{1A}$, $C_{2A}^* = h_P(C_{1A}^* \,||\, T_A)$. Since the computed result C_{2A}^* equals the received C_{2A}, S will accept U_A's login request treating it to be U's login request.

Thus Chen *et al.*'s scheme does not resist the user impersonation attack.

3.6. Server impersonation attack

In case U_A replaces the stolen SC of U, we continue the above discussion for server impersonation to fool U in the following manner:

1. When U sends the login request $\{ID, C_1, C_2, T_U\}$ to S, U_A intercepts it and computes $C_{1A}^* = P \oplus C_{1A}$, $C_{3A} = h_P(C_{1A}^* \oplus T_A \,||\, P)$ and sends $\{T_A, C_{3A}\}$ to U.
2. Since TA is fresh, so U will proceed to compute $C_{3A}^* = h_P(h(r \oplus b) \oplus T_A \,||\, P) = h_P((C_1 \oplus P) \oplus T_A \,||\, P)$. Since the computed result C_{3A}^* equals the received C_{3A}, so U will be fooled to believe the message to come from legitimate S.

Thus Chen *et al.*'s scheme does not resist the server impersonation attack.

3.7. Lack of perfect forward secrecy/unsecure session key establishment

Suppose that U's long term secret P = h(ID ⊕ x) has been compromised by U_A in some situations. As previously described, U_A can impersonate U to login S or impersonate S to fool U. We have also demonstrated that U_A can search the identity corresponding to a stolen/lost SC. With this identity the login request ID, C_1, C_2, T_U (travelling over an insecure network) of the corresponding user can be identified by U_A. Further, U_A can obtain the session key by computing h(r ⊕ b) = C_1 ⊕ P. As h(r ⊕ b) is used as the session key for securing the communications between U and S, U_A can decrypt/manipulate all the messages of the session. Thus Chen *et al.*'s scheme fails to provide perfect forward secrecy [6]. In other words we can say that Chen *et al.*'s scheme fails to establish a secure session key.

3.8. Man-in-the-middle attack

It is apparent from the above discussion that under above assumed situation, U_A can impersonate U, can spoof S and can access the secret session key. Therefore U_A can perform a man-in-the-middle attack by establishing a parallel session between U and S. Consequently Chen *et al.*'s scheme fails to provide the claimed mutual authentication.

3.9. Attack on server's secret key

Once the secret information P = h(ID ⊕ x) is revealed, then S's secret key x may also be broken. Once it is accomplished then U_A has full control over the scheme. Two different ways to attack the scheme in this scenario are as shown below:

3.9.1. U_A intercepts an arbitrary login request ID*, C_1, C_2, T_U, then he can easily impersonate U* and S

U_A selects two random numbers r_A and b_A; computes C_{1A} = h(ID* ⊕ x) ⊕ h(r_A ⊕ b_A), C_{1A} = h_P(h(r_A ⊕ b_A) —— TA). U_A sends login request ID*, C_{1A}, C_{2A}, T_A to S, which obviously will pass the authentication test. Further S computes C3A = h_P(C_{1A}* ⊕ T_S || P) = h_P(h(r_A ⊕ b_A) ⊕ TS || h(ID* ⊕ x)) and sends TS, C3A. U_A checks the validity of S by verifying the equivalence C3A* = C3A and uses h(r_A ⊕ b_A) as the session key.

3.9.2. U_A can exhaust the protected resources of S by successfully login into S as an undetectable unregistered user

From an intercepted login request ID, C_1, C_2, T_U, U_A first analyses the format of ID. Then U_A creates a fake identity IDA of same bits and similar format as that of ID and then proceeds in similar way as discussed in the former case. Since S does not maintain any record of its registered users so it will not be able to differentiate between a registered user and an unregistered user if a login request passes the ID format and timestamp freshness test.

3.10. Poor reparability

Unfortunately the attacks described by Sections 3.5., 3.6., 3.7. and 3.8., cannot be restricted and stopped even if U has detected that P = h(ID ⊕ x) has been compromised and then used a new password to re-register with S. As the value of P is determined only by U's identity ID and S's permanent secret key x, S cannot change P for U unless ID or x can be changed. However, since x is commonly used for all users rather than specifically used only for U, it is unreasonable and inefficient if x should be changed to recover the security of U only.
Additionally, it is also impractical to change ID, which should be tied to U in most application systems. Thus Chen *et al.*'s scheme is not easily reparable.

3.11. No provision for revocation of lost smart card

It is one of the requirements of smart card-based authentication schemes that in case of the loss of cards, there should be provision in the system for invalidating the further use of a lost smart card, otherwise U_A can impersonate valid registered user [8]. Through keeping record of valid card identifiers of each registered user, the authentication system can distinguish the valid card from the invalid one. Unfortunately, like the original Wang *et al.*'s scheme [40], Chen *et al.*'s scheme [3] overlooked this feature and there is no prerequisite to revoke the lost smart card. This flaw would become more catastrophic if U_A has got the lost smart card and has revealed password of a valid user by any means to login into the system for performing secure transactions, e.g. online banking and e-commerce, etc. Thus, their scheme

fails in providing the important feature of smart card-based authentication for revoking the lost smart cards without changing the user's identities [21].

3.12. Drawback

During Step-1 in the registration phase, U has to select a random number b, and remember or record it until it receives his SC from S, so that he can enter b into his SC in Step-5. Since b is a random number, it is not easy for U to remember it. Alternatively, if U records it by writing it down on a slip of paper, he has to protect that slip of paper. Therefore the process of registration in Chen *et al.*'s scheme is inconvenient for U.

4. The proposed protocol

In this section, we propose an enhancement to Chen *et al.*'s scheme that can withstand the security flaws described in the previous section. S maintains an ADB which is signed by S's secret key x. In addition, S will routinely and frequently make offsite backup of ADB. And we assume that the offsite backup is well protected. If any unauthorized modification of ADB occurs, S will detect it and then restore ADB by using the offsite backup. Our presented scheme consists of five different phases. Namely: registration phase, login phase, authentication phase, password change phase, and revocation of lost or stolen smart card phase. We illustrate the detailed processes in sequence along with Fig.1 depicting the entire protocol structure of the proposed scheme. These phases work as follows:

4.1. Registration phase

This phase is invoked whenever U initially registers or re-registers to S. The following steps are involved in this phase.

1. U $\Rightarrow$ S: ID.
2. S checks if ID is already in ADB or not. If ID already exists in ADB, S intimates U to choose another ID. In addition S checks the registration record of U and if U is a new user then S sets value of N = 0, otherwise if U is re- registering in the system then S sets N = 1 and stores (ID, N) in its ADB.
3. S combines ID with IDS in a secret way (known only to S) to produce CIUS. Then S computes CS depending on certain fixed number of bits of x, ID and IDS. Selection of bits is not random but specific pertaining to the position of bits. S computes P = h(x || CIUS || N) and a small digest h(CS) of CS.
4. S $\Rightarrow$ U: SC = P, h(.) and h(CS).
5. U generates FP, freely selects PW and a secret key K.
6. U activates the SC. U inserts the SC into the smart card reader and inputs ID, FP, K & PW.
7. SC computes R = P $\oplus$ h(FP || PW), W = P $\oplus$ K and V = h_K(P || ID || FP || PW).

The SC replaces P with R, W and V. Finally, SC = R, W, V, h(.).

4.2. Login phase

Whenever U wants to login to S, the following operations will be performed:

1. U inserts his SC into the smart card reader, generates FP and then enters ID, FP, PW and h(CS).
2. SC extracts P = R $\oplus$ h(FP || PW), K = W $\oplus$ P and checks whether h_K(P || ID || FP || PW)? = V holds or not. If not, SC terminates this session; otherwise
3. SC generates a nonce n; computes C_1 = P $\oplus$ n, M = K $\oplus$ h(P || n) and C_2 = h_K(h(CS) || n || T_U).
4. U $\rightarrow$ S: {ID, M, C_1, C_2, T_U}.

Note: SC halts and displays a need for re-registration if U fails to enter ID, FP, PW and h(CS) correctly over certain number of times.

4.3. Verification phase

After the login request is received by S, the following steps are performed by S and U:

1. If ID format is incorrect, ID is not on ADB or $(T_S - T_U) \geq \delta$ t then S drops the login request; otherwise.
2. S extracts N from ADB and computes CIUS, P = h((x || CIUS || N) and h(CS).
3. S extracts $n = C_1 \oplus P$, $K = M \oplus h(P \| n)$ and computes $C_2^* = h_K(h(CS) \| n \| T_U)$.
4. S checks if $C_2^*? = C_2$. This equivalence authenticates the legality of U and the login request is accepted else the connection is interrupted.
5. U and S agree on common session key $K_{SESS} = h[(n \| T_U) \cdot (P) \cdot (K \| h(CS))]$. Afterwards all the subsequent messages between U and S are XoR-ed with session key K_{SESS}.

4.4. Password change phase

This procedure is invoked whenever U wants to change his password PW with a new one. In this phase we emphasize that U's smart card must have the ability to detect login failure times. Once the number of login failures exceeds a predefined system value, SC must be locked immediately to prevent the exhaustive password guessing behavior. Suppose U want to select a new password PW_{NEW} and/or a new fingerprint FP_{NEW} to replace original password and/or original fingerprint FP. Then U and SC perform the following steps.

1. First SC authenticates U in a similar way as in the login phase up to Step-2. If U is authenticated successfully, then SC asks U to enter PW_{NEW} and/or FP_{NEW}.
2. SC computes $R_{NEW} = P \oplus h(FP_{NEW} \| PW_{NEW})$, $V_{NEW} = h_K(P \| ID \| FP_{NEW} \| PW_{NEW})$
3. SC replaces R, V with R_{NEW} and V_{NEW} respectively.

Note: If U wishes, then he can also update the value of K through this phase.

User		Server
Registration Phase:		
Selects ID	$\{ID\}$ →	Computes CIUS, CS and h(CS) $P = h(x \| CIUS \| N)$
	← $SC = \{P, h(.)\}, h(CS)$	Stores (ID, N) in ADB
Generates FP, selects PW and K Insert ID, FP, K, PW $R = P \oplus h(FP \| PW)$, $W = P \oplus K$ and $V = h_K(P \| ID \| FP \| PW)$ $R, W, V \leftarrow P$ Now $SC = \{R, W, V, h(.)\}$		
Login Phase:		***Verification Phase:***
U: Generates FP; inputs ID, FP, PW, h(CS)		Aborts if ID or T_U is invalid or $(T_S - T_U) \geq \Delta t$
SC: $P \leftarrow R \oplus h(FP \| PW)$, $K \leftarrow W \oplus P$		Computes CIUS, CS and h(CS)
$V^* = h_K(P \| ID \| FP \| PW)$		$P = h(x \| CIUS \| N)$
Aborts if $V^* \neq V$		$n \leftarrow C_1 \oplus P$, $K \leftarrow M \oplus h(P \| n)$
$C_1 = P \oplus n$, $M = K \ h(P \| n)$		$C_2^* = h_K(h(CS) \| n \| T_U)$
$C_2 = h_K(h(CS) \| n \| T_U)$	$\{ID, M, C_1, C_2, T_U\}$ →	
$K_{SESS} = h[(n \| T_U) \bullet (P) \bullet (K \| h(CS))]$		$K_{SESS} = h[(n \| T_U) \bullet (P) \bullet (K \| h(CS))]$

User	Smart Card
Password Change Phase:	
U: Inserts ID, FP, PW	$P \leftarrow R \oplus h(FP \| PW)$, $K \leftarrow W \oplus P$
	$V^* = h_K(P \| ID \| FP \| PW)$, aborts if $V^* \neq V$
	Asks U for PW_{NEW} and/or FP_{NEW}
$\{PW_{NEW}$ and/or $FP_{NEW}\}$ →	
	$R_{NEW} = P \oplus h(FP_{NEW} \| PW_{NEW})$
	$V = h_K(P \| ID \| FP_{NEW} \| PW_{NEW})$
	$R_{NEW} \leftarrow R$, $V_{NEW} \leftarrow V$

User		Server
Revocation Phase:		
Lost smart card	$\{SC$ is lost$\}$ →	$N + 1 \leftarrow N$ in ADB, Revocation accomplished
Sends re-registration request		

Figure 1. Proposed scheme.

4.5. Revocation phase

In case of lost/stolen SC, U, requests S for its revocation for which S increases N by one in the ADB. Later on U can re-register to S without changing his ID. At this stage U may also use a new password and a new fingerprint.

5. Security and usability analysis of the proposed scheme

5.1. Security analysis

A smart card is a memory card that uses an embedded micro-processor from a smart card reader machine to perform required operations specified in the protocol. Kocher *et al.* [19] and Messerges *et al.* [31] pointed out that all existing smart cards cannot prevent the information stored in them from being extracted by techniques such as by monitoring their power consumption. Some other reverse engineering techniques are also available for extracting information from smart cards. It means that once a smart card is stolen by U_A, he can extract the information stored in it. A good password authentication scheme should provide protection from different feasible attacks.

5.1.1. Offline password guessing attack

In case SC of U is stolen/found by U_A, he can extract all the information R, W, V, h(.) stored inside its memory. From R = P $\oplus$ h(FP || PW), U's password cannot be guessed without knowing the secret information P and without being able to generate FP. Secondly, this attack cannot be mounted using V = h_K(P || ID || FP || PW) unless U_A knows about K, and P. In fact it is not possible to guess two values simultaneously in real time polynomial and no one except U can generate FP.
Moreover, in our scheme U's login request message ID, M, C_1, C_2, T_U is well protected and un-involved with U's password. This design eliminates the correlation between U's password and the transmitted message. Thus U_A has no ability to examine his guessed password with previous legitimate requests or reply message in an offline mode. Hence our scheme is secure against the offline password guessing attack.

5.1.2. Offline secret information guessing attack

U_A cannot guess the secret information P from the stolen/found SC. To guess P from R = P $\oplus$ h(FP || PW), he must know PW and must be able to generate FP. To guess P from W = P $\oplus$ K, he must know K. To guess P from V = h_K(P || ID || FP || PW), he must know PW, K and must be able to generate FP. Suppose U_A arbitrarily guesses a value P* and computes K* = W $\oplus$ P*, but he cannot verify his guess using V = h_K(P || ID || FP || PW) without knowing FP and PW. Note that, it is not feasible for U_A to generate FP.
Further, suppose U_A intercepts the transmitted login request ID, M, C_1, C_2, T_U. Then U_A arbitrarily guesses a value P*, computes n* = C_1 $\oplus$ P*, K* = M $\oplus$ h(P* || n*), but he cannot verify his guess using C_2 = h $_K$(h(CS) || n || T_U) because C_2 does not contain the value P.

5.1.3. Tracing ID corresponding to the stolen/lost SC

As we have seen that offline password guessing attack and offline secret information guessing attack are not possible in our scheme, so identity corresponding to the stolen/lost SC cannot be traced using previously intercepted login requests.

5.1.4. Insider attack

In our scheme, U sends only his ID as the registration request. This design rules out the possibility of insider attack.

5.1.5. Clock un-synchronization problem and replay attack

As mentioned by Gong [9], the timestamp based authentication scheme may suffer from the replay attack as the transmission delay is unpredictable in a network environment. For this reason, we have also utilized a nonce based design along with timestamp based design. Accordingly, our proposed scheme can prevent serious clock un-synchronization problem.
Furthermore, in the login-verification phase, U_A may intercept U's previous login request, and send it to S as a new login request. However, in our scheme the values M, C_1, or C_2 are not allowed to be used from session to session. U_A

cannot pass the verification process at the server's side with a previously eavesdropped login request. Obviously, the replay attack fails.

5.1.6. User impersonation attack

If U_A intends to impersonate U, he has to make a legal login message. Although U_A can intercept U's previous login request ID, M, C_1, C_2, T_U, but he cannot derive/guess the correct value of P and K from it as we have seen earlier in Section 5.1.2. Moreover, he has no knowledge of CS/h(CS). Therefore, the resistance to the user impersonation attack can be guaranteed in our protocol.

5.1.7. Server impersonation attack

In the server impersonation attack, U_A can manipulate the sensitive data of legitimate users via setting up fake servers. In the proposed protocol, the malicious server cannot compute $K_{SESS} = h[(n \,||\, T_U) \cdot (P) \cdot (K \,||\, h(CS))]$ because the malicious server does not know the values of n, P, K and h(CS). Moreover, the session key is different for the same user in different login sessions. Therefore, the proposed protocol is secure against the server impersonation attack.

5.1.8. Man-in-the-middle attack/parallel session attack

We have seen earlier that U_A cannot masquerade as U either by replaying a valid intercepted login request {ID, M, C_1, C_2, T_U}, or by forging a valid login request because of the following reasons.

- C_2 consists of the fresh timestamp T_U.
- Nonce (n) based design of login request.
- U_A has no way to correctly know n, P, K, CS/h(CS) and PW of U.
- No one other than U can generate the fingerprint FP of U.

Additionally, U_A cannot compute the agreed $K_{SESS} = h[(n \,||\, T_U) \cdot (P) \cdot (K \,||\, h(CS))]$ between U and S because U_A does not know any of the values included in $_{SESS}$. Therefore, the proposed protocol is secure against the man-in-the- middle attack/parallel session attack.

5.1.9. Malicious user attack

During the registration phase, S provides SC = {P = h(x || CIUS || N), h(.)} and h(CS) to a user. A malicious privileged user having his own SC can extract the value P; but he cannot use it to create a valid login request to impersonate another valid user. Further, using his own identity, he cannot guess the secret key x of S from P. The reasons for these are as follows:

- The values CIUS, CS and P vary from one user to the other.
- The values of P and CS depend on SID and x which are known only to S.
- The method/operations employed in computing CIUS and CS is known only to S.

Therefore the proposed protocol is secure against the malicious user attack.

5.1.10. The known-key attack

Unlike Chen *et al.*'s scheme, our scheme does not employ the session key in computing the login request. If a used old session key is compromised by U_A, it is not useful for him to derive any secret information from an intercepted login request {ID, M, C_1, C_2, T_U}. Therefore, our scheme efficiently resists the known-key attack.

5.1.11. Denial of service attack

In this type of attack, an attacker updates password verification information on SC to some arbitrary value and hence a legal user cannot login successfully in a subsequent login request to the server. In our proposed protocol, U's smart card has the ability to detect login failure times. Once the number of login failures exceeds a predefined system value, SC must be locked immediately to prevent the exhaustive password guessing behavior. Besides SC checks the validity

of ID, PW and FP before the password update procedure. In fact the fingerprint FP of U can be generated by no one other than U.
On the server's side there is an ADB in our proposed protocol which stores (ID, N) corresponding to each user. So S is able to differentiate between its registered user and an un-registered user. Therefore, the proposed protocol is secure against the denial of service attack.

5.1.12. Leak of verifier attack/modification of ADB attack

In our scheme, S stores (ID, N) in its ADB. Identity is used in the plaintext form in a login request and N (an integral value which denotes the number of times U re-registers) is involved in P = h(x || CIUS || N). Even after knowing the values (ID, N), U_A cannot compute P without knowing x and IDS. Thus the values stored in ADB do not facilitate in any important guess or computation.
If U_A performs any tampering with the ADB of S, it is detectable because ADB is signed by the private key of S and is regularly verified by S. In addition, S routinely and frequently makes offsite backups of the ADB. If S detects any unauthorized modification within ADB then S can restore ADB using the offsite backup, which is assumed to be well protected. Thus the proposed protocol is secure against the leak of verifier attack/modification of ADB attack.

5.1.13. Server's secret key(x) guessing attack

To guess the secret key x of S, U_A must first know the correct value of P, but there is no way in which U_A can obtain P. Only a malicious privileged user can extract P from his own SC. As discussed in Section 5.1.9., even after knowing P, it is not possible for a malicious privileged user to guess the secret key x of S. The malicious user knows his identity ID, but cannot guess x from P, because it is not possible to guess the two values x and IDS correctly at the same time in real time polynomial.

5.2. Usability analysis

5.2.1. Mutual authentication and session key establishment

S checks the validity of U by checking if $C_2^* = C_2$ (= h_K(h(CS) || n || T_U)). This equivalence confirms the validity of U. After this, both U and S establish and agree upon a session key K_{SESS} = h[(n || T_U) · (P) · (K || h(CS))]. Authentication of S to U is not a problem after the successful establishment of secret K_{SESS}, because then U and S can secretly communicate with each other at their own will, for instance S may authenticate itself to U by sending h(K_{SESS} ⊕ (n || K)). Thus our proposed protocol provides session key establishment in a very simple way and which is also useful in accomplishing the mutual authentication.

5.2.2. Perfect forward secrecy

Suppose the long term secret key x of S is compromised by some means, U_A cannot compute P = h(x || CIUS || N) without knowing IDS and N corresponding to U. Further the mode of computing CIUS is known to S only. Moreover U_A still cannot recover the old session key KSESS = h[(n || T_U) · (P) · (K || h(CS))] unless he knows correct values of n, P, K and h(CS). Unlike Chen *et al.*'s scheme [3], our scheme does not uses K_{SESS} = h[(n || T_U) · (P) · (K || h(CS))] in login request. Further, it is apparent from the security analysis that U_A/malicious user is not able to extract/guess the correct values n, P, K and h(CS). Thus it is computationally infeasible to obtain K_{SESS}. Therefore, our improved scheme provides perfect forward secrecy [6].

5.2.3. Quick wrong password detection mechanism within smart card/efficient password authentication

In the proposed protocol, the ownership check procedure runs whenever SC is inserted in to the card reader. SC drops or rejects the request if the user fails to insert the four correct values ID, FP, PW and h(CS) simultaneously.

5.2.4. Secure password change phase

U can freely choose and change password at his will without the help of S. The phase could also avoid easy modification of the password of U by any unauthorized user if he obtains the SC of U by some means. No one having SC can update the password without the exact values of ID, FP, PW and h(CS) corresponding to the SC.

5.2.5. Revocation facility/easy reparability

The value N is used in constructing P = h(x || CIUS || N), which is stored inside SC embedded within R = P ⊕ h(FP || PW), W = P ⊕ K and V = h_K(P || ID s—— FP || PW) and has a key role throughout the scheme. At the SC loss/theft/when essential, S revokes the SC of U by incrementing N by one. Since the revoked SC contains the previous value of N, therefore SC will not pass the verification phase performed by S. However, U's account is still kept in ADB, U can re-register to S without changing his ID to obtain a new SC. If U finds or suspects that M or N has been compromised, he can change his password or re-registers to S by submitting his identity. Thus the proposed protocol provides the revocation facility and is easily reparable.

6. Performance comparison

In this section, we compare the security features, usability, communication cost and computation cost with Wang *et al.* [40], and Chen *et al.* [3] authentication and key agreement schemes for evaluating our scheme. Comparison of security features according to the criteria mentioned in [40, 28] and usability, is depicted by Table 1 and Table 2, respectively. And comparison of communication cost and computation cost is summarized in Table 3. We assume that each ID, PW, x, n, K and timestamp value is 128-bit long. Moreover, we assume that the output of the secure one-way hash function is 128-bit. Typically, time complexity associated with hash function (h), and XOR operation can be roughly expressed as (h) ≫ (⊕). In our proposed protocol the parameters stored in SC are R, W, V, so the memory needed by SC is 384 (= 3*128) bits. The communication cost of authentication includes the capacity of transmitting messages involved in the authentication scheme. The capacity of transmitting message ID, M, C_1, C_2, T_U is 640 (= 5*128) bits. The computation cost of registration (4h(.)+2⊕) is the total time of all operations executed in the registration phase. The computation cost of the user/smart card (5h(.)+4⊕) and the server (5h(.)+2⊕) is the time spent by the user and the service provider server during the process of authentication and session key establishment.
Table 1 and Table 2 depict that our scheme resists more attacks and achieves more features as compared to the related schemes and are essentially required in implementing a practical remote user authentication scheme using smart cards. It is worth additional three hashing operations and lesser XOR operations to achieve these properties, as shown in Table 3. Thus, our scheme neither overheads SC, nor S with high hash, yet provides proper mutual authentication and session key establishment. Therefore, we have proposed a protocol with better security, more admired usable features and low at computation/communication cost. We envision our proposed protocol to be free from the attacks mentioned in Table 1 and possessing the usable features mentioned in Table 2, within the focus of our discussions. However, there is always scope of improvement in any protocol as new attacks come into existence.

Table 1. Comparison of security features.

↓ Attacks & Schemes →	Wang *et al.*	Chen *et al.*	Our Protocol
Offline PW guessing attack	Yes	Yes	No
Insider attack	No	No	No
Replay attack	No	No	No
User impersonation attack	Yes	Yes	No
Server impersonation attack	Yes	Yes	No
Man-in the middle attack	Yes	Yes	No
Parallel session attack	Yes	No	No
Malicious user attack	Yes	Yes	No
Known-key attack	Yes	Yes	No
Denial of service attack	Yes	Yes	No
Leak of ADB attack	No	No	No
Secret key (x) guessing attack	Yes	Yes	No

Table 2. Comparison of usability.

↓ Usability & Schemes →	Wang *et al.*	Chen *et al.*	Our Protocol
Mutual authentication	No	No	Yes
Secure KSESS establishment	No	No	Yes
Perfect forward secrecy	No	No	Yes
Quick wrong PW detection in SC	Yes	Yes	Yes
Secure PW changing	Yes	Yes	Yes
Revocation facility	No	No	Yes
Easy reparability	No	No	Yes
User chosen PW	Yes	Yes	Yes
Change PW without S	Yes	Yes	Yes

Table 3. Comparison of communication/computational cost and complexity.

↓ Usability & Schemes →	Wang *et al.*	Chen *et al.*	Our Protocol
Memory needed by SC	384 bits	84 bits	384 bits
Comm. cost of authentication	6*128	6*128	5*128
Computation cost of registration	3h(.) + 3⊕	3h(.) + 3⊕	4h(.) + 2⊕
Computation cost of SC	5h(.) + 6⊕	5h(.) + 5⊕	5h(.) + 4⊕
computation cost of S	3h(.) + 4⊕	3h(.) + 3⊕	5h(.) + 2⊕
Sum	**11h(.) + 13⊕**	**11h(.) + 11⊕**	**14h(.) + 8⊕**

7. Conclusion

In this paper we have presented the cryptanalysis of Chen *et al.*'s remote user authentication scheme. We have shown that Chen *et al.*'s scheme suffers from the known-key attack, smart card loss attacks, poor reparability, no revocation of lost smart card, unsecure session key establishment, absence of perfect forward secrecy and malicious user attack. In addition, we have shown the serious consequences of smart card loss attack on Chen *et al.*'s scheme like the user impersonation attack, attack on server's secret key, man-in-the-middle attack/parallel session attack and denial of service attack. To overcome the identified problems, we have developed an improved a smart card based remote user authentication scheme. Our proposed remedy eradicates all the identified weaknesses of Chen *et al.*'s scheme and is more secure and robust for real life use.

References

[1] Chan C.K., Cryptanalysis of a remote user authentication schemes using smart cards, IEEE Trans. Consum. Electron., 46, 992-993, 2000

[2] Chang C.C., Hwang K.F., Some forgery attacks on a remote user authentication scheme using smart cards. Inf., 14, 289-294, 2003

[3] Chen T.H., Hsiang H.C., Shih W.K., Security enhancement on an improvement on two remote user authentication schemes using smart cards, Future Gener. Comp. Sy., 27, 377-380, 2011

[4] Chien H.Y., Jan J.K., Tseng Y.M., An efficient and practical solution to remote authentication, smart card., Comput. Secur., 21, 37-375, 2002

[5] Chien H.Y., Chen C.H., A remote authentication scheme preserving user anonymity (In: Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)), 2, 245-248, 2005

[6] Diffie W., Oorschot P.C. Van, Wiener M.J., Authentication and authenticated key exchanges., Design. Code. Cryptogr., 2, 107-125, 1992

[7] El Gamal T., A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inf. Theory, 31, 469-472, 1985

[8] Fan C.I., Chan Y.C., Zhang Z.K., Robust remote authentication scheme with smart cards., Comput. Secur., 24, 619-628, 2005
[9] Gong L., A security risk of depending on synchronized clocks, SIGOPS, 26, 49-53, 1992
[10] Goripathi T., Das M.L., Saxena A., An improved bilinear pairing based remote user authentication scheme, Comp. Stand. Inter., 2009, 31, 181-185
[11] Haller N.M., The S/KEY one-time password system, RFC1760, February 1995
[12] Horng G., Password authentication without using password table, Inform. Process. Lett., 55, 247-250, 1995
[13] Hsu C.L., Security of Chien et al.'s remote user authentication scheme using smart cards. Comp. Stand. Inter., 26, 167-169, 2004
[14] Hwang M.S., Li L.H., A new remote user authentication scheme using smart card, IEEE Trans. Consum. Electron., 46, 28-30, 2000
[15] Jan J.K., Chen Y.Y., Paramita Wisdom' password authentication scheme without verification tables, J. Syst. Software, 42, 45-57, 1998
[16] Khan M.K., Zhang J., Improving the security of a flexible biometrics remote user authentication scheme, Comp. Stand. Inter., 29, 82-85, 2007
[17] Khan M.K., Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world, IETE Tech. Rev., 26, 191-195, 2009
[18] Kim S.K., Chung M.G., More secure remote user authentication scheme, Comput. Commun., 32, 1018-1021, 2009
[19] Kocher P., Jaffe J., Jun B., Differential power analysis (In: Advances in Cryptology, CRYPTO'99), 388-397, 1999
[20] Ku W.C., Chang S.T., Chiang, M.H., Further cryptanalysis of fingerprint-based remote user authentication scheme using smartcards, IEE Electron. Lett., 41, 240-241, 2005
[21] Ku W.C., Chang S.T., Impersonation attack on a dynamic ID-based remote user authentication scheme using smart cards, IEICE Trans. Commun., E88-B, 2165-2167, 2005
[22] Ku W.C., Chen S.M., Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards, IEEE Trans. Consum. Electron., 50, 204-207, 2004
[23] Kumar M., New remote user authentication scheme using smart cards, IEEE Trans. Consum. Electron., 50, 597-600, 2004
[24] Lamport L., Password authentication with insecure communication, Commun. ACM, 24, 770-771, 1981
[25] Lee C.C., Hwang M.S., Yang, W.P., A flexible remote user authentication scheme using smart cards, ACMOSR, 36, 46-52, 2002
[26] Lee N.Y., Chiu Y.C., Improved remote authentication scheme with smart card, Comp. Stand. Inter., 27, 177-180, 2005
[27] Leung K.C., Cheng L.M., Fong A.S., Chan C.K., Cryptanalysis of a modified remote user authentication scheme using smart cards, IEEE Trans. Consum. Electron., 49 (4), 1243-1245, 2003
[28] Liao I.E., Lee C.C., Hwang, M.S., A password authentication scheme over insecure networks, J. Comput. Syst. Sci. Int., 72, 727-740, 2006
[29] Liao Y.P., Wang S.S., A secure dynamic ID-based remote user authentication scheme for multi-server environment, Comp. Stand. Inter., 31, 24-29, 2009
[30] Liu J.Y., Zhou A.M., Gao M.X., A new mutual authentication scheme based on nonce and smart cards, Comput. Commun., 31 (10), 2205-2209, 2008
[31] Messerges T.S., Dabbish, E.A., Sloan R.H., Examining smart card security under the threat of power analysis attacks, IEEE Trans. Comput., 51 (5), 541-552, 2002
[32] Mitchell C.J., Chen L., Comments on the S/KEY user authentication scheme, ACMOSR, Oct, 30, 12-16, 1996
[33] Mitchell C., Limitation of challenge-response entity authentication, Electr. Lett., 25, 1195-1196, 1989
[34] Peyravian M., Roginsky A., Zunic, N.L., Hash-based enrcyption system, Comput. Secur., 18, 345-350, 1999
[35] Shen J.J., Lin C.W., Hwang M.S., A modified remote user authentication scheme using smart cards, IEEE Trans. Consum. Electron., 49, 414-416, 2003
[36] Shimizu A., A dynamic password authentication method by one-way function, IEICE Trans. Inf. Syst., J73-D-I, 630-636, July 1990
[37] Shimizu A., Horioka T., Inagaki H., A password authentication method for contents communication on the Internet, IEICE Trans. Commun., E81-B, 1666-1763, August 1998
[38] Sun H.M., An efficient remote user authentication scheme using smart cards, IEEE Trans. Consum. Electron., 46, 958-961, 2000

[39] Tsai J.L., Efficient multi-server authentication scheme based on one-way hash function without verification table, Comput. Secur., 27, 115-121, 2008
[40] Wang X.M., Zhang W.F., Jhang J.S., Khan M.K., Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards, Comp. Stand. Inter., 29, 507-512, 2007
[41] Wang Y.Y., Kiu J.Y., Xiao F.X., Dan J., A more efficient and secure dynamic IDbased remote user authentication scheme, Comput. Commun., 32, 583-585, 2009
[42] Yang J.H., Chang C.C., An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem, Comput. Secur., 28, 138-143, 2009
[43] Yen S.M., Liao K.H., Shared authentication token secure against replay and weak key attacks, Inform. Process. Lett., 62, 77-80, 2009
[44] Yoon E.J., Ryu E.K., Yoo K.Y., Further improvement of an efficient password based remote user authentication scheme using smart cards, IEEE Trans. Consum. Electron., 50, 612-614, 2004