Secure Password-based Remote User Authentication Scheme with Non-tamper Resistant Smart Cards

Ding Wang[1,2⋆], Chun-guang Ma[1,⋆⋆], and Peng Wu[1]

[1] Harbin Engineering University, Harbin City 150001, China
[2] Automobile Management Institute of PLA, Bengbu City 233011, China
wangdingg@mail.nankai.edu.cn, chunguangma@hrbeu.edu.cn

Abstract. It is a challenge for password authentication protocols using non-tamper resistant smart cards to achieve user anonymity, forward secrecy, immunity to various attacks and high performance at the same time. In DBSec'11, Li et al. showed that Kim and Chung's password-based remote user authentication scheme is vulnerable to various attacks if the smart card is non-tamper resistant. Consequently, an improved version was proposed and claimed that it is secure against smart card security breach attacks. In this paper, however, we will show that Li et al.'s scheme still cannot withstand offline password guessing attack under the non-tamper resistance assumption of the smart card. In addition, their scheme is also vulnerable to denial of service attack and fails to provide user anonymity and forward secrecy. As our main contribution, a robust scheme is presented to cope with the aforementioned defects, while keeping the merits of different password authentication schemes using smart cards. The analysis demonstrates that our scheme meets all the proposed criteria and eliminates several hard security threats that are difficult to be tackled at the same time in previous scholarship.

Keywords: Cryptanalysis; Network security; Authentication protocol; Smart card; Non-tamper resistant;User anonymity

1 Introduction

Password-based authentication is widely used for systems that control remote access to computer networks. In order to address some of the security and management problems that occur in traditional password authentication protocols, research in recent decades has focused on smart card based password authentication. Since Chang and Wu [1] introduced the first remote user authentication scheme using smart cards in 1993, there have been many smart card based authentication schemes proposed [2–7]. In most of the previous authentication

⋆ Corresponding author.

⋆⋆ The abridged version of this paper is going to appear in the proceedings of DBSec 2012, LNCS, vol. 7371, pp. 114–121, Springer–Verlag.

schemes, the smart card is assumed to be tamper-resistant, i.e., the secret information stored in the smart card cannot be revealed. However, recent research results have shown that the secret data stored in the smart card could be extracted by some means, such as monitoring the power consumption [8,9] or analyzing the leaked information [10]. Therefore, such schemes based on the tamper resistance assumption of the smart card are vulnerable to some types of attacks, such as user impersonation attacks, server masquerading attacks, and offline password guessing attacks, etc., once an adversary has obtained the secret information stored in a user's smart card and/or just some intermediate computational results in the smart card.

Another common feature of the published schemes is that the user's identity is transmitted in plaintext over insecure networks during the authentication process, which may leak the identity of the logging user once the login messages were eavesdropped, hence user privacy is not preserved. The leakage of the user identity may also cause an unauthorized entity to track the user's login history and current location [5, 7]. In many cases, it is of utmost importance to provide anonymity so that the adversary cannot trace user activity. Therefore, user anonymity is an important feature that a practical authentication scheme should achieve.

As noted by Blake-Wilson et al. [11], forward secrecy is an admired security feature for authentication protocols with session keys establishment. Particularly, forward secrecy is a property concerned with limiting the effects of eventual failure of the entire system. It indicates that, even if the long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities should not be affected and thus the previous sessions shall remain secure [12]. Hence, a sound authentication scheme should achieve this important property.

As mentioned in Refs. [3, 7, 13, 14] and the above description, the following criteria are important for smart card based remote user authentication schemes in terms of friendliness, security and efficiency: (C1) the server needs not to maintain a security-sensitive verification table; (C2) the password is memorable, and can be chosen freely by the user; (C3) the password cannot be derived by the privileged administrator of the server; (C4) the security of the scheme is not based on the tamper resistance assumption of the smart card; (C5) the scheme can resist various kinds of sophisticated attacks, such as offline password guessing attack, replay attack, parallel session attack, denial of service attack, stolen verifier attack, user/server impersonation attack; (C6) the password cannot be broken by guessing attack even if the smart card is lost/stolen and compromised; (C7) the client and the server can establish a common session key during the authentication process; (C8) the scheme is not prone to the problems of clock synchronization and time-delay; (C9) the user can change the password locally without any interaction with the authentication server; (C10) the scheme can achieve mutual authentication; (C11) the scheme preserves user anonymity to avoid partial information leakage; (C12) the scheme provides the property of forward secrecy.

In 2009, Kim and Chung [15] showed that Yoon and Yoo's scheme [16] easily reveals a user's password and is prone to stolen verifier attack, user impersonation attack and server masquerading attack, and then they proposed an improved scheme. Later on, Kim et al. [17] showed that Kim and Chung's scheme is vulnerable to offline password guessing attack, unlimited online password guessing attack and server masquerading attack if the smart card is non-tamper resistant. In DBSec'11, Li et al. [18] also identified that Kim and Chung's scheme cannot withstand various attacks stated above and further proposed an enhanced remote authentication scheme. They claimed their scheme is secure and can overcome all the identified security flaws of Kim and Chung's scheme even if the smart card is non-tamper resistant. In this work, however, we will demonstrate that Li et al.'s scheme cannot withstand denial of service attack, and it is still vulnerable to offline password guessing attack under their assumption. In addition, their scheme does not provide the feature of forward secrecy and user anonymity. To conquer the identified weaknesses, a robust authentication scheme based on the secure one-way hash function and the well-known discrete logarithm problem is presented.

The remainder of this paper is organized as follows: in Section 2, we review Li et al.'s authentication scheme. Section 3 describes the weaknesses of Li et al.'s scheme. Our proposed scheme is presented in Section 4, and its security analysis is given in Section 5. The comparison of the performance of our scheme with the other related schemes is shown in Section 6. Section 7 concludes the paper.

2 Review of Li et al.'s scheme

In this section, we briefly illustrate the remote user authentication scheme proposed by Li et al. [18] in DBSec 2011. Their scheme consists of four phases: registration, login, verification and password update. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 1.

Table 1. Notations

Symbol	Description
U_i	i^{th} user
S	remote server
ID_i	identity of user U_i
P_i	password of user U_i
x	the secret key of remote server S
n	a large prime number
g	a primitive element in Galois $GF(n)$
$h(\cdot)$	collision free one-way hash function
$\oplus$	the bitwise XOR operation
$\|$	the string concatenation operation
$A \rightarrow B : C$	message C is transferred through a common channel from A to B
$A \Rightarrow B : C$	message C is transferred through a secure channel from A to B

2.1 Registration phase

The registration phase involves the following operations:

1) User U_i chooses his/her identity ID_i, password P_i, and then generates a random number RN_1.

2) $U_i \Rightarrow S : \{ID_i, h(h(P_i \oplus RN_1))\}$.

3) On receiving the registration message from U_i, the server S creates an entry $\{ID_i, N, h(h(P_i \oplus RN_1))\}$ in the verification table,where $N = 0$ if it is U_i's initial registration, otherwise S set $N = N + 1$. Then, server S computes $C_1 = h(ID_i \parallel x \parallel N) \oplus h(h(P_i \oplus RN_1))$

4) $S \Rightarrow U_i$: A smart card containing security parameters $\{ID_i, C_1, h(\cdot)\}$.

5) Upon receiving the smart card,user U_i stores RN_1 into his/her smart card.

2.2 Login phase

When U_i wants to login to S, the following operations will be performed:

1) U_i inserts his/her smart card into the card reader, and inputs ID_i,P_i and a random number RN_2.

2) The smart card generates a random number RC and then computes $C_2 = h(P_i \oplus RN_1)$,$C_3 = C_1 \oplus h(C_2)$,$C_4 = C_3 \oplus C_2$,$C_5 = h(h(P_i \oplus RN_2))$ and $C_6 = E_{KU_i}(C_5, RC)$,where $K_{U_i} = h(C_2 \parallel C_3)$.

3) $U_i \rightarrow S : \{ID_i, C_4, C_6\}$.

2.3 Verification phase

After receiving the login request from user U_i, S performs the following operations:

1) The server S checks the validity of identity ID_i by checking whether ID_i is already stored in its verification table. If not, the request is rejected. Otherwise, the S computes $C_7 = h(ID_i \parallel x \parallel N)$, $C_8 = C_4 \oplus C_7$, $C_9 = h(C_8)$, and compares C_9 with the third field of the entry corresponding to ID_i in its verification table. If it equals, S successfully authenticates U_i and computes symmetric key $K'_{U_i} = h(C_8 \parallel C_7)$, and obtains ($C_5$,$RC$) by decrypting C_6. Then, S replaces the third field $h(h(P_i \oplus RN_i))$ of the entry corresponding to ID_i with $C_5 = h((P_i \oplus RN_2))$, generates a random RS and computes $K_5 = h(C_7 \parallel C_8)$.

2) $S \rightarrow U_i : \{E_{K_5}(RC, RS, C_5)\}$.

3) On receiving the response from server S, the smart card computes the symmetric key $K'_s = h(C_3 \parallel C_2)$ and obtains (RC', C'_5) by decrypting the received message using K_s. Then, the smart card checks whether (RC', C'_5) equals to (RC, C_5) generated in the login phase. This equivalency authenticates the legitimacy of the server S and replaces original RN_1 and C_1 with new RN_2 and $C_3 \oplus C_5$, respectively.

4) $U_i \rightarrow S : \{h(RS)\}$

5)On receiving $h(RS)'$, the serve S compares the computed $h(RS)$ with the received value of $h(RS)'$. If they are not equal, the connection is terminated.

6) The user U_i and the server S agree on the session key $SK = h(RC \oplus RS)$ for securing future data communications.

2.4 Password change phase

The password change phase is provided to allow users to change their passwords freely. Since the password change phase has little to do with our discussion, we omit it here and detailed information is referred to Ref. [18].

3 Cryptanalysis of Li et al.'s scheme

In this section we will show that Li et al.'s scheme is vulnerable to offline password guessing attack and denial of service attack. In addition, their scheme fails to preserve user anonymity and forward secrecy. Although tamper resistant smart card is widely assumed in most of the published authentication schemes, such an assumption is difficult in practice. Many researchers have shown that the secret information stored in a smartcard can be breached by analyzing the leaked information or by monitoring the power consumption [8–10]. Be aware of this threat, Li et al. intentionally based their scheme on the assumption of non-tamper resistance of the smart card. However, Li et al.'s scheme fails to serve its purposes.

3.1 Offline password guessing attack

In Li et al.'s scheme, a user is allowed to choose his/her own password at will during the registration and password change phases. The user usually tends to select a password, i.e., his phone number, which is easily remembered for his/her convenience. Hence, these easy-to-remember passwords, called weak passwords [19], have low entropy and thus are potentially vulnerable to password guessing attack. Therefore, one of the most important security requirements for sound password-based authentication protocols is to resist against this threat. Li et al. showed that Kim and Chung's scheme is vulnerable to offline password guessing attack once the adversary has obtained the secret information stored in the stolen smart card. However, we will show that Li et al.'s scheme still suffers from this threat as follows.

Let us consider the following scenarios. In case a legitimate user U_i's smart card is stolen by an adversary $\mathcal{A}$ just before $U_i^{'}$s jth login, and the stored secret values such as C_1 and RN_j can be revealed. Then, $\mathcal{A}$ returns the smart card to U_i and eavesdrops on the insecure channel. Because U_i's identity is transmitted in plaintext within the login request, it is not difficult for $\mathcal{A}$ to identify the login request message from U_i. Once the jth login request message $\{ID_i, C_4^j = h(ID_i \parallel x \parallel N) \oplus h(P_i \oplus RN_i), C_6^j\}$ is intercepted by $\mathcal{A}$, an offline password guessing attack can be launched in the following steps:

Step 1. Guesses the value of P_i to be P_i^* from a uniformly distributed dictionary.

Step 2. Computes $T = h(h(P_i^* \oplus RN_j)) \oplus h(P_i^* \oplus RN_j)$, as RN_j is known. .

Step 3. Computes $T' = C_1 \oplus C_4^j$, as C_1 has been extracted and C_4^j has been intercepted, where $C_1 = h(ID_i \parallel x \parallel N) \oplus h(h(P_i \oplus RN_j)), C_4^j = h(ID_i \parallel x \parallel N) \oplus h(P_i \oplus RN_j)$.

Step 4. Verifies the correctness of P_i^* by checking if T is equal to T'.

Step 5. Repeats Steps 1, 2, 3, and 4 of this phase until the correct value of P_i is found.

After guessing the correct value of P_i, the adversary $\mathcal{A}$can compute $C_3^j = C_1 \oplus h(h(P_i \oplus RN_j)), C_2^j = h(P_i \oplus RN_j)$ and $K_{U_i}^j = h(C_2^j \parallel C_3^j)$. Then the adversary can obtain RC_j by decrypting C_6^j using $K_{U_i}^j$, and gets RS_j in a similar way. Hence the malicious user can successfully compute the session key $SK_j = h(RC_j \oplus RS_j)$ and renders the jth session between U_i and S completely insecure.

Moreover, once the jth login request message is intercepted, $\mathcal{A}$ may block the communication channel between U_i and S completely until the session key $SK_j = h(RC_j \oplus RS_j)$ was obtained as stated above. Thereafter, $\mathcal{A}$ can fabricate and send a valid login request to the server S and masquerade as a legitimate user U_i, or $\mathcal{A}$ can fabricate and send a valid password change request to update the entry corresponding to U_i in the verification table on S. In either case, from then on U_i will not be able to login to the server S. This leads to a strong denial of service attack.

3.2 Denial of service attack

A denial of service attack is an offensive action whereby the adversary could use some methods to work upon the server so that the login requests issued by the legitimate user will be denied by the server. In Li et at.' scheme, an adversary can easily launch a denial of service attack in the following steps:

Step 1. Eavesdrops over the channel, intercepts a login request $\{ID_i, C_4^j, C_6^j\}$ from U_i and blocks it, supposing it is U_i's jth login.

Step 2. Replaces C_6^j with an equal-sized random number R, while ID_i and C_4^j are left unchanged.

Step 3. Sends $\{ID_i, C_4^j, R\}$ instead of $\{ID_i, C_4^j, C_6^j\}$ to the remote server S.

After receiving this modified message, S will perform Step $V1$ and $V2$ of the verification phase without observing any abnormality, as a result, the verifier corresponding to ID_i in the verification table will be updated and the response $E_{K_S}(RC_j^*, RS_j, C_5^{j*})$ will be sent to U_i. On receiving the response from S, U_i decrypts $E_{K_S}(RC_j^*, RS_j, C_5^{j*})$ and will find (RC_j^*, C_5^{j*}) unequal to (RC, C_5) , thus the session will be terminated. Thereafter, U_i's succeeding login requests will be denied unless he/she re-registers to S again. That is, the adversary can easily lock the account of any legitimate user without using any cryptographic techniques. Thus, Li et al.'s protocol is vulnerable to denial of service attack.

3.3 Failure to achieve forward secrecy

Let us consider the following scenarios. Supposing the server S's long time private key x is leaked out by accident or intentionally stolen by an adversary $\mathcal{A}$. Once the value of x is obtained, with previously intercepted C_4^j, C_6^j and $E_{K_S}(RC, RS, C_5)$ transmitted in the legitimate user U_i's jth authentication process, $\mathcal{A}$ can compute the session key of S and U_i's jth encrypted communication through the following method:

Step 1. Assumes $N = 0$.
Step 2. Computes $C_7^* = h(ID_i \parallel x \parallel N)$ and $C_8^* = C_7^* \oplus C_4^j$, where ID_i is previously obtained by eavesdropping on the insecure channel.
Step 3. Computes $K_{U_i}^* = h(C_8^* \parallel C_7^*)$ and $K_S^* = h(C_7^* \parallel C_8^*)$.
Step 4. Decrypts C_6^j with $K_{U_i}^*$ to obtain RC_i^*.
Step 5. Decrypts $E_{K_S}(RC, RS, C_5)$ with K_S^* to obtain RC_i^{**}.
Step 6. Verifies the correctness of N by checking if RC_i^* is equal to RC_i^{**}. If they are unequal, sets $N = N + 1$ and goes back to Setp2.
Step 7. Decrypts $E_{K_S}(RC, RS, C_5)$ to obtain RS_i using K_S^*.
Step 8. Computes $SK_i = h(RC_i \oplus RS_i)$.

Note that the value of N should not be very big, since the re-registration phase is not performed frequently in practice, and thus the above procedure can be completed in polynomial time. Therefore, Li et al.'s scheme fails to provide forward secrecy.

3.4 Failure to preserve user anonymity

In many e-commerce applications, the violation of user anonymity may leak some personal secret information (e.g., secret online-order placement, transaction records, etc.) about the logging user to the adversary, and thus the provision of user anonymity is very important. What's more, the leakage of the user identity may cause an unauthorized entity to track the user's login history and current location [5]. Therefore, assuring anonymity does not only preserve user privacy but also make remote user authentication protocols more secure.

In Li et al.'s scheme, user's identity ID is static and in plaintext form in all the transaction sessions, an adversary can easily obtain the plaintext identity of this communicating client once the login messages were eavesdropped, and hence, different login request messages belonging to the same user can be traced out and may be interlinked to derive some secret information related to the user. Hence, user anonymity is not preserved.

4 Our proposed scheme

According to our analysis, three principles for designing a sound password-based remote user authentication scheme are presented. First, user anonymity, especially in some application scenarios, (e.g., e-commence), should be preserved,

because from the identity ID_i, some personal secret information may be leaked about the user. Second, a nonce based mechanism is often a better choice than the timestamp based design to resist replay attacks, since clock synchronization is difficult and expensive in existing network environment, especially in wide area networks, and these schemes employing timestamp may still suffer from replay attacks as the transmission delay is unpredictable in real networks [20]. Finally, the password change process should be performed locally without the hassle of interaction with the remote authentication server for the sake of security, user friendliness and efficiency [3]. In this section, we present a new remote user authentication scheme to satisfy all the twelve criteria listed in section 1.

4.1 Registration phase

Let $(x, y = g^x \bmod n)$ denote the server S's private key and its corresponding public key, where x is kept secret by the server and y is stored inside each user's smart card. The registration phase involves the following operations:

Step R1. U_i chooses his/her identity ID_i, password P_i and a random number b.
Step R2. $U_i \Rightarrow S : \{ID_i, h(b \parallel P_i)\}$.
Step R3. On receiving the registration message from U_i, the server S computes $N_i = h(b \parallel P_i) \oplus h(x \parallel ID_i)$ and $A_i = h(ID_i \parallel h(b \parallel P_i))$.
Step R4. $S \Rightarrow U_i : A$ smart card containing security parameters $\{N_i, A_i, n, g, y, h(\cdot)\}$.
Step R5. Upon receiving the smart card, U_i enters b into his smart card.

4.2 Login phase

When U_i wants to login the system, the following operations will be performed:

Step L1. U_i inserts his/her smart card into the card reader and inputs ID_i^*, P_i^*.
Step L2. The smart card computes $A_i^* = h(ID_i^* \parallel h(b \parallel P_i^*))$ and verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, it implies $ID_i^* = ID_i$ and $P_i^* = P_i$. Otherwise, the session is terminated.
Step L3. The smart card chose a random number u and computes $C_1 = g^u \bmod n$, $Y_1 = y^u \bmod n$, $h(x \parallel ID_i) = N_i \oplus h(b \parallel P_i)$, $CID_i = ID_i \oplus h(C_1 \parallel Y_1)$ and $M_i = h(CID_i \parallel Y_1 \parallel h(x \parallel ID_i))$.[0]
Step L4. $U_i \rightarrow S : \{C_1, CID_i, M_i\}$.

4.3 Verification phase

After receiving the login request, the server S performs the following operations:

[0] In the abridged version [22], there is a mistake (or typo) in the formation of M_i as $h(CID_i \parallel C_1 \parallel h(x \parallel ID_i))$, and we correct it here. June 3, 2012.

Step V1. The server S computes $Y_2 = (C_1)^x \bmod n$ using its private key x, and derives $ID_i = CID_i \oplus h(C_1 \parallel Y_2)$ and $M_i^* = h(CID_i \parallel Y_2 \parallel h(x \parallel ID_i))$. S compares M_i^* with the received value of M_i. If they are not equal, the request is rejected. Otherwise, server S generates a random number v and computes the session key $SK = (C_1)^v \bmod n$, $C_2 = g^v \bmod n$ and $C_3 = h(SK \parallel C_2 \parallel h(x \parallel ID_i))$.

Step V2. $S \rightarrow U_i : \{C_2, C_3\}$.

Step V3. On receiving the reply message from the server S, U_i computes $SK = (C_2)^u \bmod n$, $C_3^* = h(SK \parallel C_2 \parallel h(x \parallel ID_i))$, and compares C_3^* with the received C_3. This equivalency authenticates the legitimacy of the server S, and U_i goes on to compute $C_4 = h(C_3 \parallel h(x \parallel ID_i) \parallel SK)$.

Step V4. $U_i \rightarrow S : \{C_4\}$

Step V5. Upon receiving $\{C_4\}$ from U_i, the server S first computes $C_4^* = h(C_3 \parallel h(x \parallel ID_i) \parallel SK)$ and then checks if C_4^* is equal to the received value of C_4 . If this verification holds, the server S authenticates the user U_i and the login request is accepted else the connection is terminated.

Step V6. The user U_i and the server S agree on the common session key SK for securing future data communications.

4.4 Password change phase

In this phase, we argue that the user's smart card must have the ability to detect the failure times. Once the number of login failure exceeds a predefined system value, the smart card must be locked immediately to prevent the exhaustive password guessing behavior. This phase involves the following steps.

Step P1. U_i inserts his/her smart card into the card reader and inputs the identity ID_i and the original password P_i.

Step P2. The smart card computes $A_i^* = h(ID_i \parallel h(b \parallel P_i))$ and verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, it implies the input ID_i and P_i are valid. Otherwise, the smart card rejects.

Step P3. The smart card asks the cardholder to resubmit a new password P_i^{new} and computes $N_i^{new} = N_i \oplus h(b \parallel P_i) \oplus h(b \parallel P_i^{new})$, $A_i^{new} = h(ID_i \parallel h(b \parallel P_i^{new}))$. There- after, smart card updates the values of N_i and A_i stored in its memory with N_i^{new} and A_i^{new}.

5 Security analysis

Although it is important to provide a formal security proof on any cryptographic protocols, the formal security proof of user authentication protocols with smart cards remains one of the most challenging issues for cryptography research [21]. Until now, a simple, efficient and convincing formal methodology for correctness

analysis of security protocols is still an important subject of research and an open problem. Few schemes [6, 13] do provide formal security proof, unfortunately they are shortly found contradictory to their security claims because the formal methods employed all fail to capture some realistic attack scenarios [14]. Due to these reasons, most published user authentication schemes using smart cards [1–5, 7, 15–18] have been demonstrated with a simple proof. Therefore, we follow the approaches used in [5, 7] for comparison purpose. This opens a prominent future scope of this work to develop a simple and robust formal method for security analysis of user authentication protocols with smart cards. The security of our proposed authentication scheme is based on the secure hash function and the discrete logarithm problem. In the following, we will analyze the security of the proposed scheme to verify whether the security requirements mentioned in Section 1 have been satisfied under the assumption that the secret information stored in the smart card can be revealed, i.e., the security parameters N_i, A_i and y can be obtained by a malicious privileged user.

(1) User anonymity: Suppose that the attacker has intercepted U_i's authentication messages $\{CID_i, M_i, C_1, C_2, C_3, C_4\}$. Then, the adversary may try to retrieve any static parameter from these messages, but these messages are all session-variant and indeed random strings due to the randomness of u and/or v. Accordingly, without knowing the random number u, the adversary will face to solve the discrete logarithm problem to retrieve the correct value of ID_i from CID_i,, while ID_i is the only static element corresponding to U_i in the transmitted messages. Hence, the proposed scheme can preserve user anonymity.

(2) Offline password guessing attack: Suppose that a malicious privileged user U_i has got U_k's smart card, and the secret information b, N_k, A_k and y can also be revealed under our assumption of the non-tamper resistant smart card. Even after gathering this information, the attacker has to at least guess both ID_i and P_i correctly at the same time, because it has been demonstrated that our scheme can provide identity protection. It is impossible to guess these two parameters correctly at the same time in polynomial time, and thus the proposed scheme can resist offline password guessing attack with smart card security breach.

(3) Stolen verifier attack and password disclosure to server: In the proposed protocol, no sensitive verifiers corresponding to users are maintained by S. Therefore, the proposed protocol is free from stolen verifier attack. With $h(b \parallel P_i)$ instead of plaintext password P_i submitted to server S, it is computationally infeasible to derive P_i from $h(b \parallel P_i)$ without knowing the random number b due to the one-way property of the secure hash function.

(4) User impersonation attack: As CID_i, M_i, C_3 and C_4 are all protected by secure one-way hash function, any modification to these parameters of the legitimate user U_i's authentication messages will be detected by the server S if the attacker cannot fabricate the valid CID_i^*, M_i^*, $C_3^*$and C_4^*. Because the attacker has no way of obtaining the values of ID_i, P_i and

N_i corresponding to user U_i, he/she cannot fabricate the valid CID_i^*, M_i^*, $C_3^*$and C_4^*. Therefore, the proposed protocol is secure against user impersonation attack.

(5) Server masquerading attack: In the proposed protocol, a malicious server MS cannot compute the correct $Y_2 = (C_1)^x mod n$ because he/she does not know the value of S's private key x, and thus MS cannot derive the valid $ID_i = CID_i \oplus h(C_1 \parallel Y_2)$. Without knowing U_i's valid ID_i and S's private key x, MS has to break the secure one-way hash function to retrieve $h(x \parallel ID_i)$. Furthermore, because MS cannot obtain $h(x \parallel ID_i)$, it is impossible to fabricate the proper $C_3 = h(SK \parallel C_2 \parallel h(x \parallel ID_i))$ to pass the verification of U_i in Step $V3$ of the verification phase. Therefore, the proposed protocol is secure against server masquerading attack.

(6) Replay attack and parallel session attack: Our scheme can withstand replay attack because the authenticity of authentication messages $\{M_i, C_3, C_4\}$ is verified by checking the fresh random number u and/or v. On the other hand, the presented scheme resists parallel session attack, in which an adversary may masquerade as legitimate user U_i by replaying a previously intercepted authentication message. The attacker cannot compute valid C_3 because he does not know the values of $h(x \parallel ID_i)$ corresponding to user U_i. Therefore, the resistance to replay attack and parallel session attack can be guaranteed in our protocol.

(7) Mutual authentication: In our dynamic ID-based scheme, the server authenticates the user by checking the validity of C_4 in the access request. We have shown that our scheme can preserve user anonymity, so user ID_i is only known to the server S and the user U_i itself. We have proved that our scheme can resist user impersonation attack. Therefore, it is impossible for an adversary to forge messages to masquerade as U_i in our scheme. To pass the authentication of server S, the smart card first needs U_i's identity ID_i and password P_i to get through the verification in Step L2 of the login phase. In this Section, we have shown that our scheme can resist offline password guessing attack. Therefore, only the legal user U_i who owns correct ID_i and P_i can pass the authentication of server S. On the other hand, the user U_i authenticates server S by explicitly checking whether the other party communicating with can compute the valid C_3 or not. Since the malicious server does not know the values of ID_i corresponding to user U_i and x corresponding to server S, only the legitimate server can compute the correct $C_3 = h(SK \parallel C_2 \parallel h(x \parallel ID_i))$. From the above analysis, we conclude that our scheme can achieve mutual authentication.

(8) Denial of service attack: Assume that an adversary has got a legitimate user U_i's smart card. However, in our scheme, the smart card computes $A_i^* = h(ID_i \parallel h(b \parallel Pi))$ and compares it with the stored value of A_i in its memory to checks the validity of user identity ID_i and password P_i before the password update procedure. It is not possible for the adversary to guess out U_i's identity ID_i and password P_i correctly at the same time in polynomial time. Moreover, once the number of login failure exceeds

a predefined system value, the smart card will be locked immediately. Therefore, the proposed protocol is secure against denial of service attack.

(9) Forward secrecy: Following our scheme, the client and the server can establish the same session key $SK = (C_1)^v = (C_2)^u = g^{uv} \bmod n$. Based on the difficulty of the computational Diffie-Hellman problem, any previously generated session keys cannot be revealed without knowledge of the ephemeral u and v. As a result, our scheme provides the property of forward secrecy.

6 Performance analysis

To evaluate our scheme, we compare the performance and the satisfaction of the criteria among relevant authentication schemes and our proposed scheme in this section. The reason why the schemes presented in [4,5,23], instead of other works mentioned earlier in this paper, are selected to compare with is that, these three schemes are the few ones that can withstand offline password guessing attack under the non-tamper resistance assumption of the smart cards. The criteria of a secure and practical remote user authentication scheme are introduced in Section 1, and the comparison results are depicted in Table 2 and 3, respectively.

Since the login phase and verification phase are executed much more frequently than the other two phases, only the computation cost, communication overhead and storage cost during the login phase and verification phase are taken into consideration. Without loss of generality, the identity ID_i, password P_i, random numbers, timestamp values and output of secure one-way hash function are all recommended to be 128-bit long, while n, y and g are all 1024-bit long. Let T_H, T_E, T_I, T_S and T_X denote the time complexity for hash function, exponential operation, inverse operation, symmetric cryptographic operation and XOR operation respectively. Since the time complexity of XOR operation is negligible as compared to the other three operations, we do not take T_X into account. Typically, time complexity associated with these operations can be roughly expressed as $T_E \approx T_I > T_S \geq T_H \gg T_X$ [24–26].

In our scheme, the parameters $\{N_i, A_i, y_i, n, g, b\}$are stored in the smart card, thus the storage cost is $3456 * (= 3 * 128 + 3 * 1024)$bits. The communication overhead includes the capacity of transmitting message involved in the authentication scheme, which is $2560 (= *4 * 128 + 2 * 1024)$ bits. During the login and verification phase, the total computation cost of the user and server is $6T_E + 12T_H$. As illustrated in Table 2, the proposed scheme is more efficient than Horng et al.'s scheme, enjoys nearly the same performance with Chen et al.'s scheme and Chung et al.'s scheme.

[1] A denial of service attack on the server is identified in their scheme as any malicious legitimate user can fabricate valid login requests but this malicious action can be only detected in the final exchange.

[2] A reflection attack is identified in their scheme as any adversary can impersonate server S to send $\{M = T, U = V\}$ to U_i on receiving the login request message $\{ID, V, T, N\}$ at any given session.

Table 2. Performance comparison among relevant authentication schemes

	Our scheme	Li et al. [18](2011)	Chung et al.[22] (2009)	Chen et al. [4](2010)	Horng et al.[5] (2010)
Computation cost	$6T_E + 12T_H$	$12T_H$	$4T_E + 2T_I + 12T_H$	$6T_E + 5T_H$	$7T_E + 4T_S + 8T_H$
Communication cost	2560 bits	856 bits	2560 bits	2560 bits	2432 bits
Storage overhead	3456 bits	384 bits	3200 bits	3200 bits	3328 bits

Table 3. Criteria comparison among relevant authentication schemes

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Li et al.[18]	No	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes	No	No
Chung et al.[22]	Yes	Yes	Yes	Yes	No[1]	Yes	Yes	Yes	No	Yes	No	Yes
Chen et al.[4]	Yes	Yes	No	Yes	No[2]	No	Yes	No	No	Yes	No	No
Horng et al.[5]	Yes	Yes	Yes	Yes	No[1]	Yes	Yes	Yes	No	Yes	Yes	Yes

As compared to Li et al.'s scheme, to withstand offline password guessing attack, public-key techniques are employed, which has been proved unavoidable by Halevi and Krawczyk in [27], and thus at least two exponentiations are required; to provide the feature of forward secrecy, the generation of the session key based on the Diffie-Hellman key exchange algorithm is common practice, and hence it needs another four exponentiations; to achieve user anonymity and other functionalities simultaneously, some additional costs are necessary. As a word, to conquer all the identified security flaws, the decrease of some performance is unavoidable and reasonable.

Table 3 gives a comparison of the admired features of our proposed scheme with the other relevant authentication schemes. Our proposed scheme provides forward secrecy (C12) and can change password locally (C6), while the schemes presented by Li et al. and Chen et al. fail to achieve these features; Our proposed scheme preserves user anonymity (C11), while the schemes presented by Li et al., Chung et al. and Chen et al. do not provide this property; our proposed scheme can resist various kinds of known attacks (C5), while the other four latest schemes suffer from several security vulnerabilities. It is clear that our scheme meets more criteria as compared to other relevant authentication schemes using non-tamper resistant smart cards.

7 Conclusion

In this paper, we have demonstrated several attacks on Li et al.'s scheme. As to our main contribution, a robust authentication scheme is proposed to remedy these identified flaws, the security and performance analysis demonstrate that our presented scheme achieves all of the twelve independent requirements with high efficiency and thus our scheme is more secure and efficient for practical use. Remarkably, our scheme eliminates several hard security threats that are

difficult to be solved at the same time in previous scholarship. In future work, we will develop a practical formal method for security analysis of authentication protocols using smart cards.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions. This research was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61170241 and No. 61073042.

References

1. Chang, C.C., Wu, T.C.: Remote password authentication with smart cards. IEE Proceedings-E 138(3), 165–168 (1993)
2. Ku, W.C., Chen, S.M.: Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 50(1), 204–207 (2004)
3. Liao, I.E., Lee, C.C., Hwang, M.S.: A password authentication scheme over insecure networks. Journal of Computer and System Sciences 72(4), 727–740 (2006)
4. Chen, Y., Chou, J.S., Huang, C.H.: Improvements on two password-based authentication protocols. Cryptology ePrint Archive, Report 2009/561 (2009), http://eprint.iacr.org/2009/561.pdf
5. Horng, W.B., Lee, C.P., Peng, J.: A secure remote authentication scheme preserving user anonymity with non-tamper resistant smart cards. WSEAS Transactions on Information Science and Applications 7(5), 619–628 (2010)
6. Xu, J., Zhu, W.T., Feng, D.G.: An improved smart card based password authentication scheme with provable security. Computer Standards & Interfaces 31(4), 723–728 (2009)
7. Sood, S.K.: Secure Dynamic Identity-Based Authentication Scheme Using Smart Cards. Information Security Journal: A Global Perspective 20(2), 67–77 (2011)
8. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 99. LNCS, vol. 1666, pp. 388–397. Springer-Verlag, Berlin (1999)
9. Standaert, F.X., Malkin, T.G., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In: Joux, A.(ed) Advances in Cryptology - EUROCRYPT 2009. LNCS, vol 5479, pp. 443–461. Springer, Heidelberg (2009)
10. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining Smart-Card Security under the Threat of Power Analysis Attacks. IEEE Transactions on Computers 51(5), 541–552 (2002)
11. Wilson, S.B., Johnson, D., Menezes A.: Key agreement protocols and their security analysis. In: Darnell, M.(ed.): 6th IMA International Conference on Cryptography and Coding, Cirencester, LNCS, vol. 1355, pp.30–45. Springer, Heidelberg (1997)
12. Krawczyk, H.: HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: Shoup, V.(ed) CRYPTO 2005. LNCS, vol. 3621, pp. 546–566. Springer, Heidelberg (2005)
13. Wang, R.C., Juang, W.S., Lei, C.L.: Robust authentication and key agreement scheme preserving the privacy of secret key. Computer Communications 34(3), 274–280 (2011)
14. Wu, S.H., Zhu, Y.F., Pu, Q.: Robust smart-cards-based user authentication scheme with user anonymity. Security and Communication Networks 5(2), 236–248 (2012)

15. Kim, S.K., Chung, M.G.: More secure remote user authentication scheme. Computer Communications 32(6), 1018–1021 (2009)
16. Yoon, E., Yoo, K.: More efficient and secure remote user authentication scheme using smart cards. In: Proceedings of 11th International Conference on Parallel and Distributed System, pp. 73–77. IEEE Computer Society, Los Alamitos (2005)
17. Kim, J.Y., Choi, H.K., Copeland, J.A.: Further Improved Remote User Authentication Scheme. IEICE Transactions on Fundamentals E94-A(6), 1426-1433 (2011)
18. Li, C.T., Lee, C.C., Liu, C.J., Lee, C.W.: A Robust Remote User Authentication Scheme against Smart Card Security Breach. In: Li, Y. (ed.) DBSec 2011. LNCS, vol. 6818, pp. 231–238. Springer, Heidelberg (2011)
19. Klein, D.V.: Foiling the Cracker: A Survey of, and Improvements to, Password Security. In: 2nd USENIX Security Workshop, pp.5–14. USENIX Association, Portland (1990)
20. Gong, L.: A security risk of depending on synchronized clocks. ACM Operating System Review 26(1), 49–53 (1992)
21. He, D., Ma, M., Zhang, Y., Chen, C.: A strong user authentication scheme with smart cards for wireless communications. Computer Communications 34(3), 367–374 (2011)
22. Wang, D., Ma C.G., Wu, P.: Secure Password-Based Remote User Authentication Scheme with Non-tamper Resistant Smart Cards. In: N. Cuppens-Boulahia et al. (Eds.): DBSec 2012. LNCS, vol. 7371, pp. 114–121, Springer, Heidelberg.
23. Chung, H.R., Ku, W.C., Tsaur, M.J.: Weaknesses and improvement of Wang et al.'s remote user password authentication scheme for resource-limited environments. Computer Standards & Interfaces 31(4), 863–868 (2009)
24. Mao, W.B.: Modern Cryptography: Theory and Practice. Prentice Hall PTR, New Jersey (2004)
25. Wong, D.S., Fuentes, H.H.,Chan, A.H.: The Performance Measurement of Cryptographic Primitives on Palm Devices. In: Proceedings of ACSAC'01, pp.92–101. IEEE Computer Society, Washington, DC (2001)
26. Potlapally, N.R., Ravi, S., Raghunathan, A., Jha, N.K.: A study of the energy consumption characteristics of cryptographic algorithms and security protocols. IEEE Transactions on Mobile Computing 5(2), 128–143 (2006)
27. Halevi, S., Krawczyk, H.: Public-key cryptography and password protocols. ACM Transactions on Information and System Security 2(3), 230–268 (1999)