A New Dynamic ID-Based Remote User Authentication Scheme with Forward Secrecy

Chun-Guang Ma[1], Ding Wang[1,2,★], Ping Zhao[1], and Yu-Heng Wang[3]

[1] College of Computer Science and Technology, Harbin Engineering University,
145 Nantong Street, Harbin City 150001, China
wangdingg@mail.nankai.edu.cn
[2] Automobile Management Institute of PLA, Bengbu City 233011, China
[3] Golisano College of Computing and Information Sciences, Rochester Institute of Technology,
102 Lomb Memorial Dr., Rochester, NY 14623, USA

Abstract. Forward secrecy is one of the important properties of remote user authentication schemes to limit the effects of eventual failure of the entire system when the long-term private keys of one or more parties are compromised. Recently, Tsai et al. showed that Wang et al.'s dynamic ID-based remote user authentication scheme fails to achieve user anonymity and is vulnerable to user impersonation attack, and proposed an enhanced version to overcome all the identified flaws. In this paper, however, we will point out that, Tsai et al.'s scheme still suffers from the denial of service attack and cannot provide forward secrecy. To remedy these security flaws, we propose an enhanced authentication scheme, which covers all the identified weaknesses of Tsai et al.'s scheme and is more suitable for mobile application scenarios where resource constrained and security concerned.

Keywords: Password-based, Authentication protocol, Non-tamper resistant, Smart card, Cryptanalysis, Denial of service attack.

1 Introduction

With the large-scale proliferation of internet and network technologies over the last couple of decades, more and more electronic transactions for mobile devices are implemented on Internet or wireless networks. In electronic transactions, remote user authentication in insecure channel is an important issue. Smart cards have been widely used in many e-commerce applications and network security protocols due to their low cost, portability, efficiency and cryptographic properties. Smart card authentication is based on different techniques such as passwords, digital certificates and biometric technology. Among these techniques, password is the most commonly used authentication technique to authenticate users on the server due to its simplicity and convenience. Except efficiency and convenience, there are also many other desirable properties of a secure remote authentication scheme, such as freedom of choosing passwords, mutual authentication, user anonymity and forward secrecy.

[★] Corresponding author.

H. Wang et al. (Eds.): APWeb 2012 Workshops, LNCS 7234, pp. 199–211, 2012.

Recently, 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 [2-6]. In most of the previous authentication schemes, 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. 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. [7], 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. Hence, a sound authentication scheme should achieve this important property.

In 2004, Das et al. [8] first introduced the concept of dynamic ID-based authentication scheme to resist ID-theft and thus to achieve user anonymity. However, in 2005, Chien and Chen [9] pointed out that Das et al.'s scheme fails to protect the user's anonymity, so they proposed a new one. In 2009, to overcome the security pitfalls of Das et al.'s scheme, Wang et al. [10] also proposed a dynamic ID-based authentication scheme, and claimed that their scheme is more efficient and secure while keeping the merits of Das et al.'s scheme. Later on, Tsai et al. [11] pointed out that Wang et al.'s scheme fails to provide user anonymity as claimed and cannot withstand user impersonation attack, and further proposed an enhanced version to eliminate the identified defects.

In this paper, however, we will demonstrate that Tsai et al.'s scheme fails to provide the property of forward secrecy, and suffers from the denial of service attack and insider attack. In addition, their scheme also has some practical pitfalls. To conquer the identified flaws, 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 Tsai et al.'s authentication scheme. Section 3 describes the weaknesses of Tsai 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 Tsai et al.'s Scheme

For reader's convenience, we first briefly review Tsai et al.'s scheme [11] before demonstrating its weaknesses. Their scheme consists of four phases: the registration phase, the login phase, the verification phase and password update phase. 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 field $GF(n)$
$h(\cdot)$	collision free one-way hash function
$\oplus$	the bitwise XOR operation
$\parallel$	the string concatenation operation
$A \Rightarrow B : M$	message M is transferred through a secure channel from A to B
$A \rightarrow B : M$	message M is transferred through a common channel from A to B

2.1 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 and password P_i.

Step R2. $U_i \Rightarrow S$: $\{ID_i, P_i\}$.

Step R3. On receiving the registration message from U_i, the server S computes $N_i = h(P_i \parallel ID_i) \oplus h(x \parallel ID_i)$.

Step R4. $S \Rightarrow U_i$: A smart card containing security parameters $\{N_i, y, n, g, h(\cdot)\}$.

2.2 Login Phase

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

Step L1. U_i inserts his/her smart card into the card reader, and inputs ID_i and P_i.

Step L2. The smart card computes $h(x \parallel ID_i) = N_i \oplus h(P_i \parallel ID_i)$, $C = g^k \bmod n$, $CID_i = ID_i \oplus h(g^k \parallel T_1) \bmod n$, and $B_i = h(CID_i \parallel C \parallel h(x \parallel ID_i) \parallel y \parallel T_1)$, where T_1 is the current timestamp and k is a random number.

Step L3. $U_i \rightarrow S$: $\{CID_i, B, C, T_1\}$.

2.3 Verification Phase

After receiving the login request message from user U_i at time T_2, the server S performs the following operations:

Step V1. S verifies whether $(T_2 - T_1) \leq \Delta T$. If the verification fails, S rejects the login request.

Step V2. S computes $ID_i = CID_i \oplus h(C^x \parallel T_1) \bmod n$ and $B' = h(CID_i \parallel C \parallel h(x \parallel ID_i) \parallel y \parallel T_1)$, and then compares the computed B' with the received B. If they are not equal, S rejects the request.

Step V3. S computes $SK = h(h(x \parallel ID_i) \parallel T_1 \parallel B \parallel CID_i \parallel T_2)$ and $D = h(SK \parallel h(x \parallel ID_i) \parallel T_1 \parallel T_2)$.

Step V4. $S \rightarrow U_i$: $\{D, T_2\}$.

Step V5. Upon receiving the reply message $\{D, T_2\}$ at the time T_3, U_i verifies whether $(T_2 - T_1) \leq \Delta T$. If the verification fails, U_i terminates the session.

Step V6. U_i computes $SK = h(h(x \parallel ID_i) \parallel T_1 \parallel B \parallel CID_i \parallel T_2)$ and $D' = h(SK \parallel h(x \parallel ID_i) \parallel T_1 \parallel T_2)$, and then compares the computed D' with the received D. If they are not equal, U_i terminates the session.

Step V7. After authenticating each other, U_i and S use the same session key SK to secure ensuing data communications.

2.4 Password Change Phase

When U_i wants to change the old password P_i to the new password P_i^{new}, U_i insert his/her own smart card into card reader. The smart card computes $N_i^{new} = N_i \oplus h(P_i) \oplus h(P_i^{new})$ and updates N_i with the new N_i^{new}.

3 Cryptanalysis of Tsai et al.'s Scheme

In this section, we will first show that Tsai et al. have made a mistake when designing the login phase, and then we will demonstrate that their scheme fails to achieve forward secrecy, and suffers from denial of service attack and insider attack. Moreover, several practical pitfalls in their scheme are also pointed out.

3.1 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 A. Once the value of x is obtained, with previously intercepted messages $\{CID_i^j, B^j, C^j, T_1^j, T_2^j\}$ ~~transmitted during~~ U_i's jth authentication process, A can derive the session key SK^j of S and U_i's jth encrypted communication through the following method:

Step 1. Computes $L_1 = (C^j)^x \bmod n$, as C^j and x are known.

Step 2. Computes $ID_i = CID_i^j \oplus h(L_1 \parallel T_1^j)$, as CID_i^j and T_1^j are previously intercepted.

Step 3. Computes $SK^j = h(h(x \parallel ID_i) \parallel T_1^j \parallel B^j \parallel CID_i^j \parallel T_2^j)$.

Once the session key SK^j is obtained, the whole jth session will become completely insecure. Therefore, the property of forward secrecy is not provided.

3.2 Denial of Service Attack

The password change phase of Tsai et al.'s scheme is insecure like that of Wang et al.'s scheme. If an attacker manages to obtain the smart card of legitimate user U_i for a very short time, he can change the password of user U_i as follows:

Step 1. The attacker inserts U_i's smart card into a card reader and initiates a password change request.

Step 2. The attacker submits a random string R as U_i's original password and a new string P_i^{new} as the targeting new password.

Step 3. The smart card computes $N_i^{new}=N_i \oplus h(R) \oplus h(P_i^{new})$ and updates N_i with N_i^{new}.

Once the value of N_i is updated, legitimate user U_i cannot login successfully even after getting his/her smart card back because the value of $h(x \| ID_i)$ cannot be valid and thus U_i's login request will be denied by the server S during the verification phase. Hence, denial of service attack can be launched on the user U_i's smart card.

It should be noted that, although this vulnerability seems too basic to merit discussion, it cannot be well remedied just with minor revisions. To conquer this vulnerability, a verification of the authenticity of the original password before updating the value of N_i in the memory of smart card is essential. And thus, besides N_i, some additional verifier(s) or parameter(s) should be stored in the smart card, which may introduce new vulnerabilities, such as offline password guessing attack and user impersonation attack. Therefore, only radical changes can eliminate this vulnerability.

3.3 Insider Attack

In many scenarios, the user uses a common password to access several systems for his convenience. If the user registers to the server with plaintext password, an insider of server can impersonate user's login by abusing the legitimate user's password and can get access to the other systems [2].

In the registration phase, U_i's password P_i is submitted in plaintext to S, and thus it can be easily learned by the insider of S. If U_i uses this P_i to access several servers for his/her convenience, the insider of S can impersonate U_i to access other servers. Hence, it's an insecure factor to commit plain password to the server, and Tsai et al.'s scheme is susceptible to insider attack.

3.4 Some Practical Pitfalls

In practice, the length of ID_i and $h(y^k \| T_1)$ are much smaller than that of n, performing a modular operation on the value of $ID_i \oplus h(y^k \| T_1)$ as done in Step L2 of the login phase is meaningless but only to increase the computation overhead. The right way is to first compute $Y=y^k \bmod n$ and then derive $CID_i=ID_i \oplus h(Y \| T_1)$, and the same is the case with the derivation of ID_i in Step V2 of the verification phase.

Since clock synchronization is difficult and expensive in existing network environment, especially in wide area networks, these schemes employing timestamp mechanism to resist replay attacks is not suitable for use in distributed networks or large-scale application environments. What's more, these schemes employing timestamp may still suffer from replay attacks as the transmission delay is unpredictable in real networks [12].

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. In other words, without employing any effort an adversary can identify the particular transaction being performed by the user U_i. 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. 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 overcome the security flaws described in previous section.

4.1 Registration Phase

The server S generates two large primes p and q and computes $n = pq$, then chooses a prime number e and an integer d, such that ed=1 mod (p-1)(q-1). Finally, the server S finds an integer g, which is a primitive element in both Galois field $GF(p)$ and $GF(q)$, and make the values of n, e and g public, while p, q and d are only known to server S. The registration phase involves the following operations:

Step R1. The user U_i first chooses his/her identity ID_i, P_i and a random number b, and then computes $PW_i = h(b \| P_i)$.

Step R2. $U_i \Rightarrow S$: ID_i, PW_i.

Step R3. On receiving the registration message from U_i, the server S chooses random value y_i and computes $N_i = h(ID_i \| PW_i) \oplus h(d)$, $A_i = h(PW_i \| ID_i)$, $B_i = y_i \oplus ID_i \oplus PW_i$ and $D_i = h(h(ID_i \| y_i) \oplus d)$. Server S chooses the value of y_i corresponding to U_i to make sure D_i is unique to each user. The server S stores $y_i \oplus h(h(d) \| d)$ and $ID_i \oplus h(d \| y_i)$ corresponding to D_i in its database.

Step R4. $S \Rightarrow U_i$: A smart card containing security parameters $\{N_i, A_i, B_i, n, e, g, h(\cdot)\}$.

4.2 Login Phase

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

Step L1. U_i inserts his/her smart card into the card reader and inputs ID_i^* and P_i^*.

Step L2. The smart card computes $A_i^* = h(PW_i^* \| ID_i^*)$ 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 $P_i^* = P_i$. Then, the smart card chose a random number N_u and computes $y_i = B_i \oplus ID_i \oplus PW_i$, $h(d) = N_i \oplus h(ID_i \| P_i)$, $CID_i = h(ID_i \| y_i) \oplus h(h(d) \| N_u)$, $C_1 = N_u^e \bmod n$, Otherwise, the session is terminated.

Step L3. $U_i \rightarrow S$: CID_i, C_1.

4.3 Verification Phase

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

Step V1. The server S decrypts the random number N_u from C_1 using its private key d, then computes $D_i^* = h(CID_i \oplus h(h(d) \parallel N_u) \oplus d)$ and finds D_i corresponding to D_i^* in its database. If there exists no matched D_i, the request is rejected. Otherwise, server S extracts $y_i \oplus h(d \parallel h(d))$ and $ID_i \oplus h(d \parallel y_i)$ corresponding to D_i^* from its database. Now the server S computes y_i from $y_i \oplus h(h(d) \parallel d)$ and ID_i from $ID_i \oplus h(d \parallel y_i)$ because the server S knows the value of d. Then, the server S generates a random number N_S and computes the session key $SK= h(ID_i \parallel y_i \parallel N_u \parallel N_s \parallel CID_i)$, $C_2 = h(y_i \parallel ID_i \parallel SK)$.

Step V2. $S \rightarrow U_i$: N_s, C_2.

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

Step V4. $U_i \rightarrow S$: C_3.

Step V5. Upon receiving C_3 from U_i, the server S first computes $C_3^* = h(N_u \parallel N_s \parallel y_i \parallel ID_i \parallel SK)$ and then checks if C_3^* is equal to the received value of C_2 . 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 password change phase should be performed locally without interaction with the authentication server for the sake of security, user friendliness and efficiency. In addition, 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 ID_i, the original password P_i , the new password P_i^{new}.

Step P2. The smart card computes $A_i^*=h(PW_i^* \parallel ID_i^*)$ 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 smart card rejects.

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

5 Security Analysis

Recent research results have shown that the secret data stored in the common smart card could be extracted by some means, such as monitoring the power consumption [13] or analyzing the leaked information [14]. 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 [5]. Hence, a desirable scheme should put aside any special security features that could be supported by a smart-card, and simply assume that once a smart-card is stolen by an adversary, all the information stored in it are known to the adversary. In the following, we will analyze the security of the proposed scheme under the assumption that the secret information stored in the smart card can be revealed, i.e., the secret information b, N_i, A_i and B_i can be revealed. Consequently, $h(d)$ can also be obtained by a malicious privileged user U_k, as $h(d)=N_k \oplus h(h(b||P_k)||ID_k)$, where N_k and b is revealed, and the malicious user U_k knows his own identity ID_k and password P_k corresponding to his smart card.

The security of our proposed authentication scheme is based on the secure hash function and the difficulty of the large integer factorization problem. As summarized in Refs. [15] and discussed in Section 1, the following criteria are important for evaluating smart card based remote user authentication schemes in terms of security.

(1) User Anonymity: Suppose that the attacker has intercepted U_i's authentication messages (CID_i, C_1, $C_{2,}$ C_3). Then, the adversary may try to retrieve any static parameter from these messages, but CID_i and C_1, C_2, C_3 are all session-variant and indeed random strings due to the randomness of N_u. Accordingly, Without knowing the random number N_u, the adversary will face to solve the large integer factorization problem to retrieve the correct value of $h(ID_i||y_i)$ from $CID_{i,}$, while $h(ID_i||y_i)$ is the only static element in the transmitted messages. Hence, the proposed scheme can overcome the security flaw of user anonymity breach.

(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 B_k can also be revealed under our assumption of the non-tamper resistant smart card. Even after gathering this information and obtaining $h(d)=N_k \oplus h(h(\mathrm{b}||P_i)||ID_k)$, the attacker has to at least guess both ID_k and P_k correctly at the same time, because it has been demonstrated that our scheme can provide user anonymity. It is impossible to guess these two parameters correctly at the same time in real polynomial time, and thus the proposed scheme can resist offline password guessing attack with smart card security breach.

(3) Stolen Verifier Attack: In the proposed protocol, only the server S knows private secret d and stores $y_i \oplus h(h(d)||d)$ and $ID_i \oplus h(d||y_i)$ corresponding to D_i in its database. Although a malicious privileged user can compute $h(d)$ in the way described above, he/she does not have any technique to find out the value of d, nor can he/she calculates y_i corresponding to other legitimate user. Therefore, the proposed protocol is secure against stolen verifier attack.

(4) **User Impersonation Attack:** As CID_i , C_1 and C_3 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^* and C_3^*. Because the attacker has no way of obtaining the values of ID_i, P_i and y_i corresponding to user U_i, he/she cannot fabricate the valid CID_i^* and C_3^*. Therefore, the proposed protocol is secure against user impersonation attack.

(5) **Server Masquerading Attack:** In the proposed protocol, a malicious server cannot compute the session key $SK=h(ID_i||y_i||N_u||N_s||CID_i)$ and $C_2=h(y_i\ ||ID_i||SK)$ because the malicious server does not know the values of ID_i and y_i corresponding to user U_i , and has to solve the large integer factorization problem to retrieve N_u. 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 (CID_i, C_2, C_3) is verified by checking the fresh random number N_u and/or N_s. 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 the agreed session key SK and valid C_3 because he does not know the values of N_u, ID_i and y_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_3 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 obtain the correct session key $SK = h(ID_i||y_i||N_u||N_s||CID_i)$ and compute the valid C_2 or not. Since the malicious server does not know the values of N_u, ID_i and y_i corresponding to user U_i, only the legitimate server can compute the correct session key SK and C_2. 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. The smart card checks the validity of user identity ID_i and password P_i before the password update procedure. Since the smart card computes $A_i^* = h(h(b||P_i^*)||ID_i^*))$ and compares it with the stored value of A_i in its memory to verify the legality of the user before the smart card accepts the password update request, it is not possible for the adversary to guess out identity ID_i and password P_i correctly at the same time in real polynomial time. Accordingly, 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) Online Password Guessing Attack: In this type of attack, the attacker pretends to be a legitimate client and attempts to login to the server by guessing different words as password from a dictionary. In the proposed scheme, the attacker first has to get the valid smart card and then has to guess the identity ID_i and password P_i corresponding to user U_i. It is not possible to guess out identity ID_i and password P_i correctly at the same time in real polynomial time. Therefore, the proposed protocol can thwart online password guessing attack.

(10) Forward Secrecy: In our scheme, the session key SK is generated with the contribution of identity ID_i and security parameter y_i, thus the attacker cannot compute the previously generated session keys without knowing the correct value of ID_i and y_i corresponding to user U_i, even the attacker knows the server S's long time private key d. As a result, our scheme achieves forward secrecy.

6 Performance Analysis

We compare the performance and security features among the relevant password-based authentication schemes and our proposed scheme in this section. The comparison results are depicted in Table 2 and 3, respectively.

Table 2. Performance comparison among relevant authentication schemes

	Our scheme	Tsai et al. [14] (2010)	Chung et al. [4] (2009)	Horng et al. [5] (2010)	Kim et al. [6] (2011)
Total computation cost	$2T_E$+$14T_H$	$3T_E$+$10T_H$	$4T_E$+$12T_H$	$7T_E$+$4T_S$+$8T_H$	$3T_E$+$6T_H$
Communication overhead	1536 bits	2560 bits	2656 bits	2432 bits	1664 bits
Storage cost	3456 bits	2176 bits *	3232 bits	3328 bits	1280 bits

* It's likely that a parameter was missed out when Tsai et al. designed the registration phase.

Table 3. Security features comparison among relevant authentication schemes

	Our scheme	Tsai et al. [14]	Chung et al. [4]	Horng et al. [5]	Kim et al. [6]
Preserving user anonymity	Yes	Yes	Yes	Yes	No
Resistance to offline password guessing attack	Yes	Yes	Yes	Yes	Yes
Resistance to stolen verifier attack	Yes	Yes	Yes	Yes	Yes
Resistance to user impersonation attack	Yes	Yes	Yes	Yes	Yes
Resistance to server masquerading attack	Yes	Yes	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes	Yes
Resistance to parallel session attack	Yes	Yes	Yes	Yes	Yes
Resistance to denial of service attack	Yes	No	Yes	No	No
Resistance to online password guessing attack	Yes	Yes	Yes	Yes	No
Resistance to password disclosure to server	Yes	No	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes
Forward secrecy	Yes	No	Yes	No	Yes

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 and verification phase are taken into consideration. Note that the identity ID_i, password P_i, random numbers, timestamp values and output of secure one-way hash function are all 128-bit long, while n, e, d and g are all 1024-bit long. Let T_H, T_E, T_S and T_X denote the time complexity for hash function, exponential 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 $T_E >> T_S \geq T_H >> T_X$ [16-17].

In our scheme, the parameters N_i, A_i, B_i, n, e and g 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 1536 (= 4 * 128+1024) bits. During the login and verification phase, the total computation cost of the user and server is $2T_E+14T_H$. The proposed scheme is more efficient than Chung et al.'s scheme, Kim et al.'s scheme and Horng et al.'s scheme. As compared to Tsai et al.'s scheme, our scheme requires less computation cost and communication overhead; to conquer all the identified security flaws, the increase of some additional storage is reasonable and unavoidable.

In particular, since smartcards are usually characterized as resource-constrained and low-powered devices, computation cost at the user end is always regarded as a key criterion for smartcard-based schemes. In our scheme, the computation cost at user end is T_E+9T_H during the login and verification phases. Clearly, T_E is the most time-consuming operation and contributes the main overhead at user end. What needs further investigation is that, in practice, the encryption exponent e of this exponential operation is often very limited, such as e=3 and e=7, and a widely accepted encryption exponent is e=2^{16}+1 [18]. As the studies in [17,19] suggest, when the encryption exponent e is much smaller than the modular n, the time taken for pure computation is significantly shorter than that of common exponential operation with big exponents, and thus it is acceptable to conduct one such exponential operation in resource-limited environments, e.g., smart card based applications.

Table 3 gives a comparison of the security features of the proposed scheme with the other password-based authentication schemes. Our improved scheme and the scheme proposed in [4] can provide all eleven security feature, while the schemes presented in [5] and [6] are susceptible to several threats. It is clear that our scheme achieves the highest security strength with negligible decrease of performance as compared to other relevant schemes with non-tamper resistant smart cards.

It should be noted that, in our scheme, server S maintains an account-database, which contains users' security parameters for authentication. If the adversary performs any unauthorized modifications on the account-database, the data will become inconsistent and the system may be crumbled. Thus, special security measures should be taken to eliminate such risks. Fortunately, the countermeasure is not complicated: S can routinely and frequently make offsite backup of the account-database and check the consistency, and restore the account-database by using the offsite backup when necessary. Thus, there is a trade-off between performance and functionality in our scheme, while this trade-off is inevitable for authentication schemes with provision of sound reparability [4].

7 Conclusion

In this paper, we have shown that, besides some practical pitfalls, Tsai et al.'s scheme suffers from denial of service attack and fails to provide forward secrecy. As to our main contribution, a robust dynamic ID-based authentication scheme is proposed to remedy these identified flaws, the security and performance analysis demonstrate that our presented scheme achieves all of the security requirements with high efficiency, and thus our scheme is more secure and efficient for practical application environment. Remarkably, our scheme eliminates several hard security threats that are difficult to be solved in the previous scholarship at the same time. Since our security analysis is still scenario-based, in future work, we will perform a more rigorous security analysis of our scheme by employing some suitable formal methods.

Acknowledgements. 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. 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)
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. Kim, J.Y., Choi, H.K., Copeland, J.A.: Further Improved Remote User Authentication Scheme. IEICE Transactions on Fundamentals 94(6), 1426–1433 (2011)
7. Wilson, S.B., Johnson, D., Menezes, A.: Key Agreement Protocols and Their Security Analysis. In: Darnell, M.J. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 30–45. Springer, Heidelberg (1997)
8. Das, M.L., Saxena, A., Gulati, V.P.: A dynamic ID-based remote user authentication scheme. IEEE Transactions on Consumer Electronics 50(2), 629–631 (2004)
9. Chien, H.Y., Chen, C.H.: A remote authentication scheme preserving user anonymity. In: IEEE AINA 2005, pp. 245–248. IEEE Computer Society, Los Alamitos (2005)
10. Wang, Y.Y., Kiu, J.Y., Xiao, F.X.: A more efficient and secure dynamic ID-based remote user authentication scheme. Computer Communications 32(4), 583–585 (2009)
11. Tsai, J.L., Wu, T.C., Tsai, K.Y.: New dynamic ID authentication scheme using smart cards. International Journal of Communication Systems 23(12), 1449–1462 (2010)
12. Gong, L.: A security risk of depending on synchronized clocks. ACM Operating System Review 26(1), 49–53 (1992)

13. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
14. 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, 541–552 (2002)
15. Tsai, C.S., Lee, C.C., Hwang, M.S.: Password Authentication Schemes: Current Status and Key Issues. International Journal of Network Security 3(2), 101–115 (2006)
16. Schneier, B.: Applied cryptography,protocols, algorithms, and source code in C, 2nd edn. John Wiley and Sons Inc., New York (1996)
17. Wong, D.S., Fuentes, H.H., Chan, A.H.: The Performance Measurement of Cryptographic Primitives on Palm Devices. In: Proceedings of ACSAC 2001, pp. 92–101. IEEE Computer Society, Washington, DC (2001)
18. Mao, M.B.: Modern Cryptography: Theory and Practice. Prentice Hall PTR, New Jersey (2004)
19. Potlapally, N.R., Ravi, S., Raghunathan, A., et al.: A study of the energy consumption characteristics of cryptographic algorithms and security protocols. IEEE Transactions on Mobile Computing 5(2), 128–143 (2006)