Cryptanalysis and Improvement of Sood et al.'s Dynamic ID-Based Authentication Scheme

Chun-Guang Ma[1], Ding Wang[1,2,*], and Qi-Ming Zhang[1]

[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

Abstract. Anonymity is one of the important properties of remote authentication schemes to preserve user privacy. Recently, Sood et al. showed that Wang et al.'s dynamic ID-based remote user authentication scheme fails to preserve user anonymity and is vulnerable to various attacks if the smart card is non-tamper resistant. Consequently, an improved version of dynamic ID-based authentication scheme was proposed and claimed that it is efficient and secure. In this paper, however, we will show that Sood et al.'s scheme still cannot preserve user anonymity under their assumption. In addition, their scheme is also vulnerable to the offline password guessing attack and the stolen verifier attack. To remedy these security flaws, we propose an enhanced authentication scheme, which covers all the identified weaknesses of Sood et al.'s scheme and is more secure and efficient for practical application environment.

Keywords: Dynamic ID, Authentication protocol, Non-tamper resistant, Smart card, Cryptanalysis, Anonymity.

1 Introduction

With the significant advances in communication networks over the last couple of decades, smart cards have been widely used in many ecommerce 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, digital signature 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, session key generation, forward secrecy and user anonymity.

Recently, because of the advantages of smart cards, a number of password-based authentication schemes with smart cards have been proposed [1-6]. Most of the proposed schemes assume that the smart card is tamper-resistant, i.e., the secret information stored in the smart card cannot be revealed. However, recent research

* Corresponding author.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 141–152, 2012.

results have shown that the secret information stored in the smart card could be extracted by some means, such as monitoring the power consumption [7] or analyzing the leaked information [8]. Therefore, such schemes based on the tamper resistance assumption of the smart card are prone 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.

A common feature among most of the published schemes is that the user's identity is static in all the transaction sessions, which may leak the identity of the logging user once the login messages were eavesdropped, hence user anonymity 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. Therefore, assuring anonymity does not only preserve user privacy but also make remote user authentication protocols more secure. One of the solutions to preserve user anonymity is to employ dynamic ID in different login requests. In 2004, Das et al. [9] first introduced the concept of dynamic ID authentication scheme to resist ID-theft and thus to achieve user anonymity. However, in 2005, Chien and Chen [10] 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. [11] 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.

All of the above three dynamic ID authentication schemes are based on the tamper-resistant assumption of the smart card. However, it is a challenge that the smart card is non-tamper resistant while preserving user anonymity. In 2007, Hu et al. [12] showed that Chien-Chen's scheme is vulnerable to the strong masquerading server/user attack, if the smart card is no longer tamper-resistant, and then they proposed an improved scheme. Later on, Horng et al. [13] showed that Hu et al.'s scheme is still vulnerable to the strong masquerading server/user attack and the offline password guessing attack. Therefore, Horng et al. proposed an improvement over Hu et al.'s scheme to remedy their drawbacks. In 2010, Yeh et al. [14] pointed out that Wang et al.'s scheme is insecure against replay attack, impersonation attack, man-in-the-middle attack and password guessing attacks. In 2011, Khan et al. [15] found Wang et al.'s scheme also does not provide user anonymity, session key agreement and revocation of lost smart card.

In 2011, Sood et al. [16] also identified that Wang et al.'s scheme cannot withstand various attacks stated above and further proposed an enhanced remote authentication scheme. They claimed their scheme is efficient and can overcome all the identified security drawbacks of Wang et al.'s scheme even if the smart card is non-tamper resistant. Unfortunately, in this paper, however, we will demonstrate that Sood et al.'s scheme cannot withstand stolen verifier attack and is still vulnerable to offline password guessing attack. And to our surprise, user anonymity, which is the most essential security feature a dynamic ID authentication scheme is designed to support, cannot be preserved. To conquer the aforementioned weaknesses, an enhancement of Sood et al.'s scheme is presented.

The remainder of this paper is organized as follows: in Section 2, we review Sood et al.'s authentication scheme. Section 3 describes the weaknesses of Sood et al.'s scheme. Our improved 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 Sood et al.'s scheme

In this section, we examine the dynamic ID authentication scheme using smart cards proposed by Sood et al. [16] in 2011. Sood et al.'s scheme consists of four phases: the registration phase, the login phase, the verification and session key agreement phase and the password change 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	master secret of remote server S
y_i	a random value corresponding to user U_i
p,q,n	p and p are two large prime numbers, and n=pq
e,d	e is a prime number and d is an integer, where ed=1mod(p-l)(q-1)
$H(\cdot)$	collision free one-way hash function
$\oplus$	the bitwise XOR operation
$\|$	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

The registration phase involves the following operations:

Step R1. Server S authenticates itself to the user U_i using its public key certificate. Then U_i generates and encrypts the session key (SS) with the public key (PK) of the server S as $(SS)_{PK}$.

Step R2. $U_i \rightarrow S:(SS)_{PK}, (ID_i)_{SS}, (P_i)_{SS}$.

Step R3. On receiving the registration message from U_i, the server S decrypts the session key (SS) using its private key. Thereafter, the server S decrypt the identity $(ID_i)_{SS}$ and password $(P_i)_{SS}$. Then server S chooses random value y_i and computes $N_i=H(ID_i\|P_i)\oplus H(x)$, $A_i=H(ID_i\|P_i)\oplus P_i\oplus H(y_i)$, $Bi=y_i\oplus ID_i \oplus P_i$ and $D_i=H(H(ID_i\|y_i)\oplus x)$. Server S chooses the value of y_i corresponding to each user to make sure D_i is unique for each user. The server S stores $y_i\oplus x$ and $ID_i\oplus H(x\|y_i)$ corresponding to D_i in its database.

Step R4. $S \Rightarrow U_i$: A smart card containing security parameters $(N_i, A_i, B_i, H(\cdot))$.

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 $y_i^* = B_i \oplus ID_i^* \oplus P_i^*$, $A_i^* = H(ID_i^* \| P_i^*) \oplus P_i^* \oplus H(y_i^*)$. Smart card verifies the validity of A_i^* by checking whether A_i^* equals the stored A_i. If the verification holds, the smart card computes $H(x) = N_i \oplus H(ID_i \| P_i)$, $CID_i = H(ID_i \| y_i) \oplus H(H(x) \| T)$ and $M_i = H(ID_i \| H(x) \| y_i \| T)$, where T is current date and time. Otherwise, the session is terminated.

Step L3. $U_i \rightarrow S$: CID_i, M_i, T.

2.3 Verification and Session Key Agreement Phase

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

Step A1. The server S checks the validity of timestamp T by checking $(T' - T) <= \delta T$, where T' is current date and time of the server S and δT is permissible time interval for a transmission delay. The server S computes $D_i^* = H(CID_i \oplus H(H(x) \| T) \oplus x)$ and finds D_i corresponding to D_i^* in its database and then extracts $y_i \oplus x$ and $ID_i \oplus H(x \| y_i)$ corresponding to D_i^* from its database. Now the server S computes y_i from $y_i \oplus x$ and ID_i from $ID_i \oplus H(x \| y_i)$ because the server S knows the value of x.

Step A2. The server S computes $M_i^* = H(ID_i \| H(x) \| y_i \| T)$ and compares M_i^* with the received value of M_i. This equivalency authenticates the legitimacy of the user U_i and the login request is accepted else the connection is terminated.

Step A3. The user U_i and the server S agree on the common session key $SK = H(H(x) \| ID_i \| T \| y_i)$ for securing future data communications.

2.4 Password Change Phase

When U_i wants to change the password, the following operations will be performed:

Step P1. U_i inserts his/her smart card into the card reader and inputs ID_i^* and P_i^*.

Step P2. The smart card computes $y_i^* = B_i \oplus ID_i^* \oplus P_i^*$, $A_i^* = H(ID_i^* \| P_i^*) \oplus P_i^* \oplus H(y_i^*)$. Smart card verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If it holds, ID_i^* will be equal to ID_i and P_i^* will be equal to P_i, otherwise, the smart card rejects the password change request.

Step P3. The smart card asks the cardholder to resubmit a new password P_i^{new} and computes $N_i^{new} = N_i \oplus H(ID_i \| P_i) \oplus H(ID_i \| P_i^{new})$, $A_i^{new} = H(ID_i \| P_i^{new}) \oplus P_i^{new} \oplus H(y_i)$ and $B_i^{new} = y_i \oplus ID_i \oplus P_i^{new}$. 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} respectively.

3 Cryptanalysis of Sood et al.'s Scheme

In this section we will show that Sood et al.'s scheme fails to preserve user anonymity and is vulnerable to offline password guessing attack and stolen verifier attack. Although tamper resistant smart card is widely assumed in most of the authentication schemes, but such an assumption is difficult in practice. Many researchers have

shown that the secret stored in a smartcard can be breached by analyzing the leaked information or by monitoring the power consumption [7,8]. Be aware of this threat, Sood et al. intentionally based their scheme on the assumption of non-tamper resistance of the smart card. However, Sood et al.'s scheme fails to serve its purposes.

3.1 Failure of Protecting the User's Anonymity

Let us consider the following scenarios. A malicious privileged user U_i having his own smart card can gather information $N_i = H(P_i||ID_i) \oplus H(x)$ from his own smart card. Then he can find out the value of H(x) as $H(x)=N_i \oplus H(P_i||ID_i)$ because the malicious user U_i knows his own identity ID_i and password P_i corresponding to his smart card. Then the attacker can successfully learn some sensitive user-specific information about any legitimate client through the following steps:

Step 1. Eavesdrops and intercepts a login request message (CID_k, M_k, T) of user U_k from the public communication channel.
Step 2. Computes $L_1= H (H (x)||T)$, where H (x) and T are known.
Step 3. Computes $L_2= CID_k \oplus L_1$.

Its obvious that L_2 is unconditionally equal to $H(ID_k||y_k)$, while the value of $H(ID_k||y_k)$ is kept the same for all the login requests of user U_k and is specific to user U_k. This value $H(ID_k||y_k)$ can be seen as user U_k's identification, and an attacker can, therefore, use this information to trace and identify the user U_k's requests. From the above attack, any legal user who logins to the remote server would be exposed to attacker U_i, and thus the scheme fails to achieve user anonymity, which is the most essential security feature a dynamic ID authentication scheme is designed to support.

3.2 Offline Password Guessing Attack

In Sood 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 convenience. Hence, these easy-to-remember passwords, called weak passwords, have low entropy and thus are potentially vulnerable to offline password guessing attack. Inevitably, user's ID, chose by the user at will as described in the scheme, suffers from the same threat. Thus, the result of $ID \oplus P$ shall not be of high entropy if both user's ID and password P are human memorable and of low entropy. Therefore, the result of $ID \oplus P$ also is exposed to the same threat.

Let us consider the following scenarios. A malicious privileged user U_i having his own smart card can gather information $N_i = H(P_i||ID_i) \oplus H(x)$ from his own smart card. Then he can find out the value of H(x) as $H(x)=N_i \oplus H(P_i||ID_i)$ because the malicious user U_i knows his own identity ID_i and password P_i corresponding to his smart card. In case another user U_k's smart card is stolen by this malicious user, he can perform offline password guessing attack in the following steps:

Step 1. Extracts the information N_k, A_k and B_k in U_k's smart card.
Step 2. Computes $T_1=P_k \oplus H(y_k)=N_k \oplus A_k \oplus H(x)$, as N_k, A_k and H(x) are known.
Step 3. Computes $T_2= N_k \oplus H(x)$, as N_k and H(x) are known.
Step 4. Assumes $R= P_k \oplus ID_k$.

Step 5. Guesses the value of R to be R^* from a uniformly distributed dictionary.
Step 6. Computes $T_3=H((R^* \oplus T_1 \oplus H(B_k \oplus R^*)) \| (T_1 \oplus H(B_k \oplus R^*)))$.
Step 7. Verifies the correctness of R^* by checking if T_3 is equal to T_2.
Step 8. Repeats steps 4, 5, and 6 of this phase until the correct value of R is found.
Step 9. Computes $P_k= T_1 \oplus H(B_k \oplus R)$, $ID_k=R \oplus P_k$.

Because $y_k= B_k \oplus R$, it is obvious that the following relationships hold true: $P_k = T_1 \oplus H(y_k) = T_1 \oplus H(B_k \oplus R)$, $ID_k= R \oplus P_k= R \oplus T_1 \oplus H(B_k \oplus R)$ and $T_3 =H(ID_k^* \| P_k^*)$. As $N_k= H(ID_k \| P_k) \oplus H(x)$ is predefined by the authentication system, the equality of $H(ID_k \| P_k) = N_k \oplus H(x)$ will always hold. Therefore, the attacker U_i can confirm the correctness of the guessed R^* by the verification in Step 7. Once the correct value of R is obtained, the correct value of password P_k and identity ID_k can be computed in step 9. Thus, Sood et al.'s scheme cannot withstand offline password guessing attack.

After guessing the correct value of P_k and ID_k, an attacker can compute $y_k= B_k \oplus ID_k \oplus P_k$. Then the attacker can fabricate and send a valid login request message (CID_k^*, M_k^*, T_u) to the service provider server S, where T_u is the current timestamp of the attacker U_i. Hence the malicious user can successfully make a valid login request to masquerade as a legitimate user U_k.

Moreover, once the adversary obtains the correct value of P_k and ID_k, he/she can easily change the password to a new one, this causes the password change phase becoming insecure. Even if the adversary returns the changed smart card to the original user U_k, U_k will not be able to login to the remote server S. This leads to a denial of service attack.

3.3 Stolen Verifier Attack

Let us consider the following scenarios. A malicious privileged user U_i having his own smart card can gather information $B_i = y_i \oplus ID_i \oplus P_i$ from his own smart card. Then he can find out the value of y_i as $y_i = B_i \oplus ID_i \oplus P_i$ because the malicious user U_i knows his own identity ID_i and password P_i corresponding to his smart card. In case the verifier table in the database of the server S is leaked out or stolen by this malicious user, he can compute the private key x of the server S as $x= (y_i \oplus x) \oplus y_i$ because the value of y_i is known. With this x, the malicious user can compute any y_k corresponding to user U_k from the item $y_k \oplus x$ stored in the verifier table, then the malicious user can launch user/server impersonation attacks successfully. As a result, the entire authentication scheme will be compromised.

4 Our Proposed Scheme

The abbreviations and notations used in the following sections are listed in Table 1.

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-l)(q-1). Finally, the server S makes the values of n and e public, while p, q and d are only known to server S. The registration phase involves the following operations:

Step R1. Server S authenticates itself to the user U_i using its public key certificate. Then U_i generates and encrypts the session key (SS) with the public key (PK) of the server S as $(SS)_{PK}$.

Step R2. $U_i \rightarrow S$: $(SS)_{PK}$, $(ID_i)_{SS}$, $(P_i)_{SS}$.

Step R3. On receiving the registration message from U_i, the server S decrypts the session key (SS) using its private key. Thereafter, the server S decrypts the identity $(ID_i)_{SS}$ and password $(P_i)_{SS}$. Then server S chooses random value y_i and computes $N_i=H(ID_i||P_i) \oplus H(d)$, $A_i=H(P_i||ID_i) \oplus H(y_i)$, $B_i=y_i \oplus ID_i \oplus P_i$ and $D_i=H(H(ID_i||y_i) \oplus d)$. Server S chooses the value of y_i corresponding to each user to make sure D_i is unique for 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, $H(\cdot)$).

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 $y_i^* = B_i \oplus ID_i^* \oplus P_i^*$, $A_i^* = H(P_i^* \,||\, ID_i^*) \oplus H(y_i^*)$. Smart card verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, the smart card chose a random number N_u and computes $H(d)=N_i \oplus H(ID_i||P_i)$, $CID_i=H(ID_i||y_i) \oplus H(H(d)||N_u||T)$, $C_i=N_u^e$ mod n, and $M_i=H(ID_i||H(d)||y_i||T||N_u)$, where T is current date and time. Otherwise, the session is terminated.

Step L3. $U_i \rightarrow S$: CID_i, C_i, M_i, T.

4.3 Verification and Session Key Agreement Phase

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

Step A1. The server S checks the validity of timestamp T by checking $(T' - T) <= \delta T$, where T' is current date and time of the server S and δT is permissible time interval for a transmission delay. The server S decrypts the random number N_u from C_i using its private key d, then computes $D_i^* = H(CID_i \oplus H(H(d)||N_u||T) \oplus d)$ and finds D_i corresponding to D_i^* in its database, then extracts $y_i \oplus H(H(d)||d)$ and $ID_i \oplus H(d||y_i)$ corresponding to D_i^* from its database. Now the server S computes y_i from $y_i \oplus H(H(d)||d)$ and ID_i from $ID_i \oplus H(d \,||\, y_i)$ because the server S knows the value of d.

Step A2. The server S computes $M_i^* = H(ID_i||H(d)||y_i||T)$ and compares M_i^* with the received value of M_i. This equivalency authenticates the legitimacy of the user U_i and the login request is accepted else the connection is terminated.

Step A3. The user U_i and the server S agree on the common session key $SK=H(H(d)||ID_i||T||y_i)$ 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. The other parts of this phase are the same with that of Sood et al.'s scheme.

5 Security Analysis

The security of our proposed authentication scheme is based on the secure hash function and the difficulty of the large integer factorization problem. In this section, we analyze the security features provided by our scheme under the assumption that the secret information stored in the smart card can be revealed, i.e., H(d) can be obtained by a malicious privileged user.

(1) User anonymity: Suppose that the attacker has intercepted U_i's login request message (CID_i, C_i, M_i, T). Then, the adversary may try to retrieve any static parameter from the login message, but CID_i, C_i and M_i 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 login request. Hence, the proposed scheme can overcome the security flaw of user anonymity breach which is inherent in Sood et al.'s scheme.

(2) Offline password guessing attack: Suppose that a malicious privileged user U_i has intercepted U_k's login request message (CID_k, C_k, M_k, T) and also has got U_k's smart card. With these harsh terms and under our assumption of non-tamper resistant smart card, the secret information N_i, A_i and B_i can also be revealed. Even after gathering this information and obtaining $H(d)=N_k \oplus H(P_k||ID_k)$, the attacker has to at least guess both ID_i and P_i correctly at the same time. It impossible to guess these two parameters correctly at the same time in real polynomial time.

(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 in Section 3.1, 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 both CID_i and M_i are protected by secure one-way hash function, any modification to these two parameters of the legitimate user U_i's login request message will be detected by the server S if the attacker cannot fabricate the valid CID_i^* and M_i^*. 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 can not fabricate the valid CID_i^* and M_i^*. 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 $SK = H(H(d)||ID_i||T'||y_i||N_u)$ because the malicious

server does not know the values of N_u, ID_i and y_i corresponding to user U_i. Moreover, the session key is session-variant for the same user U_i. 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 login request message (CID_i, C_i, M_i, T) is verified by checking the freshness of timestamp T. 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 login request message within the valid time frame window. The attacker cannot compute the agreed session key SK between user U_i and server S 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 M_i in the login 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 implicitly by checking whether the other party communicating with can obtain the correct session key $SK = H(H(d)||ID_i||T||y_i||N_u)$ and decrypt the encrypted messages successfully 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. 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, smart card checks the validity of user identity ID_i and password P_i before the password update procedure. Since the smart card computes $A_i^* = H(P_i^*||ID_i^*)) \oplus H(y_i^*)$ 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. Moreover, once the number of login failure exceeds the 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 is secure against online password guessing attack.

(10) Forward secrecy: An authentication scheme with forward secrecy assures that even if the server S's long time private key d is leaked out by accident or is stolen by an adversary, it is still impossible for an adversary to obtain the session keys generated before, nor can the adversary launch user/server impersonation attack successfully. In our scheme, the session key SK and login message M_i are generated with the contribution of y_i, which can not be computed without knowing the correct value of the identity ID_i and password P_i corresponding to user U_i, even the attacker knows the server S's long time private key d and has got user U_i's smart card. As a result, our scheme provides the property of forward secrecy.

6 Performance Analysis

We compare the performance and security features among the relevant dynamic ID-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 dynamic ID-based schemes

	Our scheme	Sood et al. [16]	Khan et al. [15]	Hu et al. [12]	Horng et al. [13]
Total computation cost	$2T_E+12T_H$	$12T_H$	$10T_H$	$4T_E+4T_S+6T_H$	$7T_E+4T_S+8T_H$
Communication overhead	1408 bits	384 bits	768 bits	3456 bits	2432 bits
Storage cost	2432 bits	384 bits	384 bits *	2816 bits	3328 bits

* It's likely that a parameter was missed out when Khan et al. design the registration phase [17].

Table 3. Security features comparison among relevant dynamic ID-based schemes

	Our scheme	Sood et al. [16]	Khan et al. [15]	Hu et al. [12]	Horng et al. [13]
Preserving user anonymity	Yes	No	No	No	Yes
Resistance to offline password guessing attack	Yes	No	No	No	Yes
Resistance to stolen verifier attack	Yes	No	Yes	Yes	Yes
Resistance to user impersonation attack	Yes	Yes	Yes	No	Yes
Resistance to server masquerading attack	Yes	Yes	Yes	No	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	Yes	Yes	Yes	No
Resistance to online password guessing attack	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	No	Yes	Yes

An efficient authentication scheme must take computation cost, communication overhead and storage cost into consideration. We mainly focus on the efficiency of login and verification phases since these two phases are the main body of an authentication scheme. Note that the identity ID_i, password P_i, timestamp values and output of secure one-way hash function are all 128-bit long, while n, e and d 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 $T_E > T_S > T_H >> T_X$.

In our scheme, the parameters N_i, A_i, B_i, n and e are stored in the smart card, thus the storage cost is 2432(= 3 * 128+2 * 1024) bits. The communication overhead includes the capacity of transmitting message involved in the authentication scheme, which is 1408 (= 3 * 128+1024) bits. During the login, verification and session key agreement phase, the total computation cost of the user and server is $2T_E+12T_H$. The proposed scheme is more efficient than Hu et al.'s scheme [12] and Horng et al.'s scheme [13], and requires more computation, communication and storage than that of Sood et al.'s scheme [16] and Khan et al.'s scheme [15], but it is highly secure as compared to the related schemes.

Table 3 gives a comparison of the security features of the proposed scheme with the other relevant dynamic ID-based authentication schemes. The proposed scheme provides user anonymity and resists offline password guessing attack, while the latest schemes proposed by [12], [15] and [16] suffer from these attacks. The proposed scheme can withstand denial of service attack, while the scheme presented by [13] is vulnerable to this attack. It is clear that our scheme is more secure as compared to other relevant dynamic ID-based schemes.

7 Conclusion

More recently, Sood et al. showed that Wang et al.'s dynamic ID-based remote user authentication scheme cannot defend against various attacks and then proposed an improved scheme. However, in this paper, we argue that Sood et al.'s scheme fails to preserve user anonymity and is vulnerable to offline password guessing attack and stolen verifier attack under the assumption of non-tamper resistance of the smart card. As to our main contribution, an improved dynamic ID-based authentication scheme was proposed to remedy these security flaws, the security and performance analysis demonstrated that the improved scheme is more secure and practical. In future work, we will give a formal security proof of our proposed scheme.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61170241 and No. 61073042, and the open program of State Key Laboratory of Networking and Switching Technology under Grant No. SKLNST-2009-1-10. The authors are grateful to the four anonymous reviewers for their valuable suggestions and comments that highly improve the readability and completeness of the paper.

Reference

1. 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)
2. Chen, Y.C., Yeh, L.Y.: An efficient nonce-based authentication scheme with key agreement. Applied Mathematics and Computation 169(2), 982–994 (2005)
3. Shieh, W.G., Wang, J.M.: Efficient Remote Mutual Authentication and Key Agreement. Computers and Security 25(1), 72–77 (2006)
4. Hsiang, H.C., Shih, W.K.: Weaknesses and Improvements of the Yoon-Ryu-Yoo Remote User Authentication Scheme using Smart Cards. Computer Communications 32(4), 649–652 (2009)
5. Kumar, M.: A new secure remote user authentication scheme with smart cards. International Journal of Network Security 11, 88–93 (2010)
6. Sood, S.K., Sarje, A.K., Singh, K.: Secure Dynamic Identity-Based Remote User Authentication Scheme. In: Janowski, T., Mohanty, H. (eds.) ICDCIT 2010. LNCS, vol. 5966, pp. 224–235. Springer, Heidelberg (2010)
7. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
8. 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)
9. 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)
10. 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)
11. Wang, Y.Y., Liu, J.Y., Xiao, F.X., Dan, J.: A more efficient and secure dynamic ID-based remote user authentication scheme. Computer Communications 32(4), 583–585 (2009)
12. Hu, L.L., Yang, Y.X., Niu, X.Y.: Improved remote user authentication scheme preserving user anonymity. In: Fifth Annual Conference on Communication Networks and Services Research, pp. 323–328. IEEE Computer Society, Los Alamitos (2007)
13. 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)
14. Yeh, K.H., Su, C.H., Lo, N.W.: Two robust remote user authentication protocols using smart cards. Journal of Systems and Software 83(12), 2556–2565 (2010)
15. Khan, M.K., Kim, S.K., Alghathbar, K.: Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme'. Computer Communications 34(3), 305–309 (2011)
16. Sood, S.K.: Secure Dynamic Identity-Based Authentication Scheme Using Smart Cards. Information Security Journal: A Global Perspective 20(2), 67–77 (2011)
17. He, D.B., Chen, J.H., Zhang, R.: Weaknesses of a dynamic ID-based remote user authentication scheme. International Journal of Electronic Security and Digital Forensics 3(4), 355–362 (2010)