Modified Efficient and Secure Dynamic ID-Based User Authentication Scheme

Toan-Thinh Truong[1], Minh-Triet Tran[1], and Anh-Duc Duong[2]

[1] University of Science, VNU-HCM
{ttthinh,tmtriet}@fit.hcmus.edu.vn
[2] University of Information Technology, VNU-HCM
daduc@uit.edu.vn

Abstract. Communication is necessary operations in wireless environments. Therefore, we must have a secure remote authentication to defend transactions against illegitimate adversaries in such risky channel. Smart card is one of methods that many schemes used due to its convenience. Recently, Khurram Khan has proposed an enhancement scheme using smart card to recover some pitfalls in Wang et al.'s scheme. They claimed that their scheme remedy those security flaws. Nevertheless, we point out that Khan et al.'s scheme cannot protect user's anonymity. Besides, it does not achieve secret key forward secrecy and cannot resist denial of service attack due to values stored in server's database. Consequently, we present an improvement to their scheme to isolate such problems.

Keywords: Mutual authentication, Dynamic identity, Anonymity.

1 Introduction

In network environment, users can access services via different devices, for example, PC, laptop, mobile phone. Communication between those devices and services can be operated with many network technologies, such as, wireless, 3G.

There are many methods of constructing secure authentication. In 1981, Lamport [1] is the first person applying cryptographic hash function in authentication. Later, many author also use this approach in their protocols. Typically, there are protocols of Cheng-Chi Lee [2] and Jau-Ji Shen [3]. These authors follow Lamport's approach with slight difference that they use identity to authenticate instead of password table. In 2004, Das et al proposed a scheme [4]. Their scheme has three main advantages. Firstly, it allows users to change password freely. Moreover, it does not maintain a verification table which is used to check login message. Finally, the scheme's security is based on secure one-way hash function. In 2005, I-En Liao [5] discovered Das's scheme not only cannot resist some basic kinds of attacks such as password-guessing but also do not provide mutual authentication. Furthermore, in Das's scheme, password is transmitted in plain-text form at registration phase. Therefore, it is easy to be stolen by server. In I-En Liao's scheme, author use hash function with password before transmitting it to server. So, even server do not know actual user's password. Nonetheless,

K. Mustofa et al. (Eds.): ICT-EurAsia 2013, LNCS 7804, pp. 409–415, 2013.
© IFIP International Federation for Information Processing 2013

with hash function, I-En Liao's scheme is also vulnerable to password-guessing attack. Consequently, E-J Yoon proposed an improvement [6] to I-En Liao's. In [6], author utilizes random value with password when hashing. So, this causes attacker not to be able to guess true user's password. Recently, Khuram Khan with E-J Yoon's approach devised a protocol [7]. In [7], authors also distribute common secret information to all users and use timestamp to resist replay attack. They also claimed their scheme can protect user's anonymity. In this paper, we prove that their scheme has inability to defend anonymity. Furthermore, it also cannot achieve secret key forward secrecy and does not withstand denial of service attack due to values stored in database's server. Ultimately, we propose an improved version to recover problems mentioned.

The remainder of this paper is organized as follows: section 2 quickly reviews Khan's scheme and discusses its weaknesses. Then, our proposed scheme is presented in section 3, while section 4 discusses the security and efficiency of the proposed scheme. Our conclusions are presented in section 5.

2 Review and Cryptanalysis of Khurram Khan's Scheme

In this section, we review Khan's scheme [7] and show his scheme cannot obtain secret key forward secrecy and doesn't protect user's anonymity.

2.1 Review of Khurram Khan's Scheme

Their scheme includes five phases. Some important notations are listed as follow:

- U_i, S: Qualified user & remote server.
- pw_i, $h(.)$: Unique password of U_i and one-way hash function.
- x, y: The first & the second secret keys of the remote server.
- T, N: The timestamp & the number of times a user registers.
- SC, SK: The smart card & the session key.
- $\oplus$, $\|$: The exclusive-or operation & concatenation operation.

Registration Phase. U_i submits ID_i & $h(r \parallel pw_i)$ to S via a secure channel, where r is a random value chosen by user. Then, S performs the following steps.

1. S checks U_i's registration credentials & checks ID_i's existence. If it already exists & N is equal to 0, S requests U_i to choose another ID_i. Otherwise, S computes $J = h(x \parallel ID_U)$ where $ID_U = (ID_i \parallel N)$ and $L = J \oplus RPW$.
2. S issues SC with $\{L, y\}$ to U_i over secure channel. Then, U_i stores r in SC.

Login Phase. After receiving SC from S, U_i uses it to login into S.

1. U_i inserts SC into another terminal's card-reader. Then he keys pw_i & ID_i.
2. SC computes $RPW = h(r \parallel pw_i)$ & $J = L \oplus RPW$. Then, SC acquires T_i & computes $C_1 = h(T_i \parallel J)$.
3. SC generates a random value d & computes $AID_i = ID_i \oplus h(y \parallel T_i \parallel d)$. Finally, SC sends login message $m = \{AID_i, T_i, d, C_1\}$ to S.

Authentication Phase. S authenticates the users login request.

1. Verifies the validity of time interval between T_i & $T^{'}$. If ($T^{'}$ - T_i) $\geq \Delta T$, where ΔT denotes the expected valid time interval for transmission delay. If this holds, then S rejects & terminates the session.
2. Computes $ID_i = AID_i \oplus h(y \parallel T_i \parallel d)$ & checks if ID_i is valid, otherwise terminates the operation. Then S checks N in database & computes $ID_U = (ID_i \parallel N)$, $J = h(x \parallel ID_U)$. Next S checks if $h(T_i \parallel J)$?= C_1. If this holds, it means U_i is an authentic user, whereas the login request is rejected.
3. S acquires T_S, computes $C_2 = h(C_1 \oplus J \oplus T_S)$ & sends $\{C_2, T_S\}$ to U_i.
4. When receiving message from S, U_i checks time interval between T_S & $T^{''}$, where $T^{''}$ is the timestamp when mutual authentication message was received. If ($T^{''}$ - T_S) $\geq \Delta T$, then U_i rejects this message & terminates the session. Otherwise, U_i checks if $h(C_1 \oplus J \oplus T_S)$?= C_2. If this doesn't holds, U_i terminates session. Otherwise U_i & S share $SK = h(C_2 \oplus J)$.

Password Change Phase. In this phase, U_i can change his or her password.

1. U_i inserts SC into card-reader & inputs ID_i & pw_i.
2. SC computes $RPW^* = h(r \parallel pw_i)$ & $J^* = L \oplus RPW^*$. If J ?= J^* holds, then U_i is allowed to update password. Otherwise, this phase is terminated.
3. SC computes $L = J \oplus RPW \oplus RPW^* \oplus h(r \parallel pw^{'}_i)$ & replaces the old value of L with the new value. Now, the new password is updated.

Lost Smart Card Revocation Phase. U_i performs some steps to revoke SC.

1. S checks secret credentials's U_i, e.g. date of birth, identity card number.
2. S changes the value of N to revoke SC. In every case of stolen or lost of SC, N is increased by one. Later, U_i can re-register to S without changing ID_i.
3. S requests U_i to return to registration phase. Here, U_i is strongly recommended not to use any previous values for new registration, e.g. password & random value, otherwise anybody can impersonation U_i by using the same credentials previously saved in the lost or stolen SC.

2.2 Cryptanalysis of Muhammad Khurram Khan's Scheme

- Inability To Protect User's Anonymity: Khan et al claimed that only S can recover ID_i of U_i due to y used to hide the user's identity during transmission of login message. Hence, adversaries cannot identify the person trying to login into S. We see this explanation is not approriate because anyone being a valid user can know y. For example, another valid user captures $\{AID_i, T_i, d, C_1\}$. Then, he computes $ID_i = AID_i \oplus h(y \parallel T_i \parallel d)$
- Secret key forward secrecy: Khan et al claimed that even if the server's x & y happens to be compromised, an adversary cannot impersonate legitimate users by using the revealed keys, because he cannot compute AID_i & C_1 in the login message without knowledge of the user's ID_i, pw_i, r & ID_U. In this subsection, we will prove his claim is not true.

1. With y, adversary A can capture any login message & compute ID_i of any user by performing $ID_i = AID_i \oplus h(y \parallel T_i \parallel d)$.
2. A creates a new login message to impersonate U_i. Firstly, A picks T_A, random value d_A. Then, A computes $AID_i = ID_i \oplus h(y \parallel T_A \parallel d_A)$.
3. A assumes $N = 0$ & computes $J = h(x \parallel ID_U)$, where $ID_U = (ID_i \parallel N)$
4. A computes $C_A = h(T_A \parallel J)$ and sends $\{AID_i, T_A, d_A, C_A\}$ to S.
5. If everything is alright, A successfully impersonates U_i. Otherwise, A turns back to step 3 with increasing N by one & continues later steps.

– Denial of service attack: In Khan's scheme, passwords are not stored at the verification server. However, this scheme stores N value of U_i. If these values are modified by attackers, many users cannot login into server. For example, in authentication phase of Khan's scheme, S must compute $J = h(x \parallel ID_U)$, where $ID_U = (ID_i \parallel N)$. Then, S compares C_1 with $h(T_i \parallel J)$. So, if N is modified, C_1 is not equal to $h(T_i \parallel J)$ & S will reject login message.

3 Proposed Scheme

Our scheme is also divided into the five phases

3.1 Registration Phase

U_i submits ID_i, $h(r \parallel pw_i)$, where r is a nonce chosen by U_i. After receiving $\{ID_i, h(r \parallel pw_i)\}$ from user via a secure channel, S performs following steps.

1. Checking ID_i's existence. If it existed, S intimates U_i to choose another ID.
2. Generating a random value e & computing $J = h(x \parallel e)$, $P = h(J)$ & $L = J \oplus RPW$. Then S sends SC containing $\{L, e, P\}$ for U_i via a secure channel.
3. U_i receives SC & inputs r into it.

3.2 Login Phase

U_i inserts SC into card-reader, inputs ID_i & pw_i to login S. Then, SC performs:

1. Computing $RPW = h(r \parallel pw_i)$ & $J^* = L \oplus RPW$.
2. Checking whether $h(J^*)$?= P. If this holds, SC goes to next step. Otherwise, it terminates the session. Then, SC generates a random value R_U & computes $AID_i = ID_i \oplus R_U$, $C_1 = R_U \oplus J^*$ & $M_1 = h(ID_i \parallel J^* \parallel R_U)$.
3. Finally, SC sends $\{AID_i, C_1, M_1, e\}$ to S.

3.3 Mutual Authentication and Session Key Agreement Phase

S receives U_i's login message $\{AID_i, C_1, M_1, e\}$ and performs following steps.

1. S computes $J^{**} = h(x \parallel e)$, $R_U = C_1 \oplus J^{**}$ & $ID^*{}_i = AID_i \oplus R^*{}_U$.

2. S checks identity's validity. If everything isn't alright, S terminates the session, otherwise S checks whether $M_1 \stackrel{?}{=} h(ID^*_i \parallel J^{**} \parallel R^*_U)$. If this doesn't holds, S terminates the session, otherwise S generates a random value R_S & computes $C_2 = R_S \oplus J^{**}$, $M_2 = h(R_S \parallel J^{**} \parallel ID^*_i)$. Then S sends $\{M_2, C_2\}$ to user via a common channel.
3. After receiving $\{M_2, C_2\}$ from S. U_i computes $R^*_S = C_2 \oplus J^*$ & check if $M_2 \stackrel{?}{=} h(R^*_S \parallel J^* \parallel ID_i)$. If this does not hold, U_i terminates the session. Otherwise, U_i authenticates S successfully. U_i sends $M_3 = h(R_U \parallel R^*_S)$ to S & computes a session key $SK = h(R_U \parallel R^*_S \parallel J^* \parallel ID_i)$.
4. When receiving $\{M_3\}$ from U_i, S checks if $M_3 \stackrel{?}{=} h(R^*_U \parallel R_S)$. If this does not hold, S terminates the session. Otherwise, S authenticates U_i successfully. And S also computes $SK = h(R^*_U \parallel R_S \parallel J^{**} \parallel ID^*_i)$.

3.4 Password Update Phase

When U_i wants to change password pw_i. He can perform following steps:

1. Insert SC into card-reader, input ID_i, pw_i & choose a new password pw_{inew}.
2. SC computes $RPW = h(r \parallel pw_i)$ & $J^* = L \oplus RPW$. Then, SC checks if $h(J^*) \stackrel{?}{=} P$. If this doesn't holds, SC terminates the session. Otherwise, SC computes $L_{new} = J^* \oplus RPW_{new}$, where $RPW_{new} = h(r \parallel pw_{inew})$
3. Finally, SC replaces L_i with L_{inew}.

3.5 Lost Smart Card Revocation Phase

We also recommend user not to use any previous values for new SC, e.g. password and random value. Following are some steps to perform in this phase:

1. User inputs old ID_i, new password pw_i & new random value r. Then U_i sends {credentials, ID_i, RPW} to S via a secure channel.
2. After receiving this package of message, S checks ID_i's validity in database. If it does not exist, S terminates the session. Otherwise, S continues to check U_i's credentials. If everything is alright, S generates a new random value e.
3. S computes $J = h(x \parallel e)$, $L = J \oplus RPW$ & $P = h(J)$. Then S sends new SC containing $\{L, e, P\}$ to U_i. Finally, U_i updates random value r into SC.

4 Security and Efficiency Analysis

In this section, we analyze our scheme on two aspects: security and efficiency.

4.1 Security Analysis

1. Replay Attack: In our scheme's authentication phase, if adversary A captures $\{AID_i, C_1, M_1, e\}$, he still cannot re-send this package again. For example, A replays the package, but A cannot compute random value R_S from S because of lacking value J. So, our scheme resists this kind of attack.

2. User's Anonymity: In our scheme, if adversary A wants to know ID_i of user, A must know random value R_U. However, R_U is enrypted with the value J which is not be leaked. Therefore, our scheme can protect user's anonymity.
3. Stolen Verifier Attack: In our scheme, S does not store any user's information except identity, so our scheme can counteract this kind of attack. In our scheme, S generates a random value e for each user. Hence, when authenticating with S, U_i only needs to send e to S and S uses master key x to re-construct $h(x \parallel e)$ of that user. So, S doesn't need to keep U_i's password.
4. Denial of Service Attack: In Khan's scheme, author stores value N of each user. So, if all N in server's database are modified, all users cannot login to server in login phase. Unlike his scheme, our scheme does not store any user's information. Hence, our scheme is immune from this kind of attack.

Besides above attacks, our scheme also has security features similar to Khan's.

4.2 Efficiency Analysis

We reuse approach used in some previous schemes to analyze computational complexity. That is, we calculate the number of one-way hash function execution. Let T_h be the time to compute one-way hash function. Khan's scheme needs $2 \times T_h$ in registration phase, and $8 \times T_h$ in login and authentication phases. Our scheme needs $3 \times T_h$ in registration phase and $9 \times T_h$ in login and authentication phases. Besides, we see his scheme doesn't resist to denial of service attack and cannot protect user's anonymity. Our proposed scheme recovers two pitfalls successfully. However, we don't solve secret key forward secrecy yet.

Table 1. A comparison between our scheme & Khan's for withstanding various attacks

Kinds of Attacks \ Schemes	Khan's	Ours
Denial of service attack	No	Yes
User's anonymity	No	Yes
Insider attack	Yes	Yes
Stolen verification table attack	Yes	Yes
Secret key forward secrecy	No	No
Mutual authentication & SK exchange	Yes	Yes
Replay attack	Yes	Yes

5 Conclusions

In this paper, we review Khan's scheme. Although his scheme can withstand some attacks, we see that his scheme is still vulnerable to denial of service attack, secret key forward secrecy and cannot protect user's anonymity. Consequently, we propose an improved scheme to eliminate some problems existing.

References

[1] Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24, 770–772 (1981)
[2] Lee, C.-C., Lin, T.-H., Chang, R.-X.: A secure dynamic id based remote user authentication scheme for multi-server environment using smart cards. Expert Syst. Appl. 38(11), 13 863–13 870 (2011)
[3] Shen, J.-J., Lin, C.-W., Hwang, M.-S.: A modified remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 49, 414–416 (2003)
[4] 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)
[5] Liao, I.-E., Lee, C.-C., Hwang, M.-S.: Security enhancement for a dynamic id-based remote user authentication scheme. IEEE Transactions on Consumer Electronics 50, 629–631 (2004)
[6] Yoon, E.-J., Yoo, K.-Y.: Improving the Dynamic ID-Based Remote Mutual Authentication Scheme. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 499–507. Springer, Heidelberg (2006)
[7] Khana, M.K., Kimb, S.-K., Alghathbara, K.: Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme. Computer Communications 34(3), 305–309 (2010)