Dynamic Identity-Based Authentication Scheme with Perfect Forward Secrecy Session Key

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

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

Abstract. Password is one of the simple and efficient methods to protect the transactions in insecure network environments. There are many authors researching in this area to suggest the protocols preventing illegitimate users from accessing the systems. In 2013, Y-H An proposed the scheme to isolate some problems which exist in Khan et al.'s scheme. In this paper, we demonstrate that Y-H An's scheme is vulnerable to server forgery attack and cannot provide user's anonymity. Furthermore, we also propose the modified scheme to overcome these limitations.

Keywords: Dynamic identity, Mutual authentication, Remote communication, User anonymity, Security.

1 Introduction

Several insecure wireless environments, such as GSM and CDPD, need an authentication scheme to defend against some adversaries. Such schemes play important role in the remote systems because they will protect the transactions from illegal accessing.

There are many approaches to construct a secure authentication. In 1981, Lamport [1] is the pioneer using one-way hash function in authentication scheme. In his scheme, there is the password table for checking user's legality at login phase. Therefore, if this table is revealed, service provider and registered users will be damaged by adversaries. Recently, Das et al. proposed the scheme [2] with some improved ideas. Their scheme do not maintain password table for verification and use dynamic identity to resist ID-theft and forgery attack. In 2005, I-En Liao [3] proved Das's scheme is not able to withstand password-guessing attack and cannot provide mutual authentication. Additionally, password in Das's scheme is submitted in raw form at registration phase. In 2006, E-J Yoon discovered that I-En Liao's scheme is still vulnerable to password guessing attack, and he proposed the scheme [4] to improve I-En Liao's scheme. Yoon utilizes random values with password when using one-way hash function. Hence, this makes the adversaries not to exactly guess true user's password. In 2010, Khan et al. suggested the scheme [5]. In this scheme, he distributes the shared secret information to all users and use time-stamp to confront impersonation and replay

I. Zelinka et al. (eds.), *AETA 2013: Recent Advances in Electrical Engineering and Related Sciences*, Lecture Notes in Electrical Engineering 282,
DOI: 10.1007/978-3-642-41968-3_35,

attack. However, in 2013, Y-H An showed that Khan's scheme cannot protect the users from password guessing and forgery attacks. Moreover, it cannot provide user's anonymity. He also proposed an improved scheme [6] to isolate the weaknesses of Khan et al.'s. Nevertheless, in this paper, we prove that Y-H An's scheme has inability to provide user's anonymity. Furthermore, it also cannot prevent adversaries from impersonating the server. Ultimately, we propose the modified version to recover the problems mentioned.

Rest of this paper is organized as follows: section 2 quickly reviews Y-H An's scheme and discusses its limitations. 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 Y-H An's Scheme

In this section, we review Y-H An's scheme and show his scheme cannot resist server impersonation attack and cannot provide user's anonymity.

2.1 Review of Y-H An's Scheme

His scheme includes three phases: registration phase, login phase, and authentication phase. Some important notations are listed as follow:

- U_i: User i^{th}
- S: Remote server
- pw_i: Password of U_i $h(.)$: The one-way hash function
- x: The secret key kept by the remote server
- y: The common secret number kept by the users
- T: The timestamp
- N: The number of times a user registers
- SC: The smart card
- SK: The common session key
- $\oplus$: The exclusive-or operation
- $\|$: Concatenation operation

Registration Phase. First of all, S chooses a large prime p and finds a primitive element in $GF(p)$. Afterwards, U_i starts to register to S (Illustrated in Figure 1).

1. U_i freely chooses ID_i and pw_i. Then, U_i sends masked password $RPW = h(r \| pw_i)$ and ID_i to S over secure channel, where r is the random value chosen by U_i.
2. After receiving registration message from U_i, S computes $ID_U = (ID_i \| N)$, $J = h(x \| ID_U)$, $m = J \oplus h(x)$, and $L = m \oplus RPW$.
3. S issues the SC containing $\{L, J, y, h(.)\}$ to U_i over secure channel.
4. U_i enters random number r in the SC.

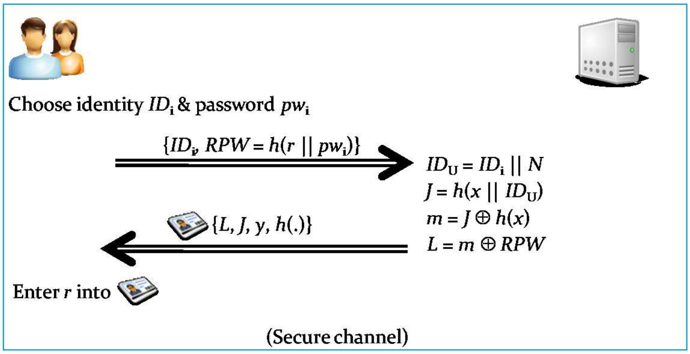

Fig. 1. Y-H An's registration phase

Login Phase. This phase is performed when U_i logins to S (Illustrated in Figure 2).

1. U_i inserts SC into another card-reader. Then he enters pw_i & ID_i.
2. SC generates a random number r_C and computes $R_C = g^{r_C} \bmod p$.
3. Then, SC computes $m = L \oplus RPW$, $\mathbf{C}_1 = J \oplus m \oplus R_C$ and $AID_i = ID_i \oplus h(y \| T_i \| R_C)$, where T_i is the current timestamp.
4. Finally, U_i sends a login request message $\{AID_i, C_1, J, T_i\}$ to S.

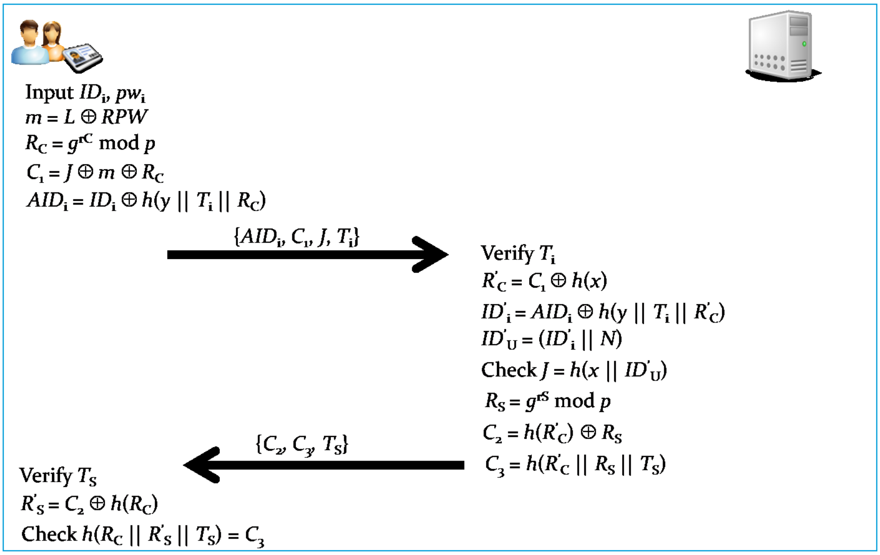

Fig. 2. Y-H An's login and authentication phases

Authentication Phase. This phase is performed when S receives the user's login request message.

1. S picks the current $T^{'}$ and verifies the user's T_i. If $T^{'}$ - $T_i \leq \Delta T$, S accepts the login request, where ΔT denotes the expected valid time interval for transmission delay.
2. S computes $R^{'}{}_C = C_1 \oplus h(x)$, $ID^{'} = AID_i \oplus h(y \parallel T_i \parallel R^{'}{}_C)$ and $ID^{'}{}_U = ID^{'} \parallel N$.
3. Then, S checks if $J = h(x \parallel ID^{'}{}_U)$ or not. If they are equal, S successfully authenticates U_i.
4. S generates a random value r_S and computes $R_S = g^{r_S} \bmod p$.
5. Continually, S computes $C_2 = h(R^{'}{}_C) \oplus R_S$ and $C_3 = \mathrm{h}(R^{'}{}_C \oplus R_S \oplus T_S)$ to U_i.
6. Once receiving the message from S, SC verifies T_S by picking current $T^{''}$. If $T^{''}$ - $T_S \leq \Delta T$, SC accepts the message.
7. SC computes $R^{'}{}_S = C_2 \oplus h(R_C)$ and checks if $C_3 = h(R_C \oplus R^{'}{}_S \oplus T_S)$ or not. If they are the same, U_i successfully authenticates S.
8. Finally, after obtaining mutual authentication, S and U_i can share a common $SK = (R^{'}{}_S)^{r_C} = (R^{'}{}_C)^{r_S} = g^{r_S r_C} \bmod p$ for secrecy communication.

2.2 Cryptanalysis of Y-H An's Scheme

In this subsection, we prove that Y-H An's scheme fails to provide user's anonymity and withstand server impersonation attack.

Inability to Protect User's Anonymity. In Y-H An's scheme, any user can know the identities of other users. From the SC, he or she can compute the server's secret information $h(x)$. With this key, legal user can capture any login message of other users and find their identities. Following are some steps which another user can perform.

1. He computes masked password $RPW = h(r \parallel pw_i)$. Then he computes $m = L \oplus RPW$ and finds server's secret information $h(x) = m \oplus J$.
2. With this $h(x)$, he will capture any login message $\{AID_i, C_1, J, T_i\}$. Then, he computes $m = h(x) \oplus J$, $R_C = C_1 \oplus m \oplus J$.
3. Finally, with R_C of other users, he can detect identity $ID_i = AID_i \oplus h(y \parallel T_i \parallel R_C)$.

The main reason why Y-H An's scheme cannot provide user's anonymity is that this scheme distributes the same $h(x)$ to all users. Therefore, if another user know this information, he or she will detect identity of other users with above steps.

Server Impersonation. In Y-H An's scheme, any user can impersonate remote server to cheat other users. Similarly, another user also computes $h(x)$. Next, the user captures any login message from another user $\{AID_i, C_1, J, T_i\}$ and finds R_C. All steps is the same as the steps in finding identity except some next steps below:

1. He can generate a random value r_A and compute $R_A = g^{r_A} \bmod p$.
2. Next, he computes $C^*{}_2 = h(R_C) \oplus R_A$ and $C^*{}_3 = h(R_C \oplus R_A \oplus T_A)$, where T_A is the current timestamp.
3. He sends $\{C^*{}_2,\ C^*{}_3,\ T_S\}$ to waiting user.
4. Waiting user will re-compute $R^{'}{}_A = C^*{}_2 \oplus h(R_C)$, then check if $C^*{}_3 = h(R_C \oplus R^{'}{}_A \oplus T_S)$ or not.
5. Finally, waiting user computes a common $SK = (R_A)^{r_C} \bmod p$ and malicious user also computes $SK = (R_C)^{r_A} \bmod p$

The main reason why Y-H An's scheme cannot resist server impersonation is that his scheme distributes the same $h(x)$ to all users. Furthermore, the authentication key J is directly transmitted through a common channel. Consequently, with that supportive message-package, malicious user easily computes secret information of other users.

3 Proposed Scheme

Our scheme includes four phases: registration phase, login phase, authentication phase, and password-update phase. Similarly to Y-H An's scheme, remote system needs to choose large prime p and primitive element in $GF(p)$. Then, U_i starts to register to S.

3.1 Registration Phase

This phase is performed when U_i registers to S (Illustrated in Figure 3).

1. U_i chooses ID_i and pw_i. Then, U_i submits ID_i and $RPW = h(r \parallel pw_i)$ to S through a secure channel, where r is a random value chosen by U_i.
2. When receiving $\{ID_i,\ RPW\}$ from U_i, S checks ID_i's existence. If this ID_i is exist, S asks U_i to choose another identity. Otherwise, S continues to go next step.
3. S generates a random value e and computes $J = h(x \parallel e)$, $L = J \oplus h(ID_i \parallel RPW)$ and $V = h(J \parallel ID_i \parallel RPW)$.
4. S sends the SC including $\{L,\ V,\ e,\ h(.)\}$ to U_i through a secure channel.
5. After receiving SC from S, U_i securely enters random value r into SC.

3.2 Login Phase

U_i inserts SC into card reader, and enters ID_i and pw_i to login to S. Next, SC performs:

1. Compute $RPW = h(r \parallel pw_i)$ and $J = L \oplus h(ID_i \parallel RPW)$.
2. Verify if V ?= $h(J \parallel ID_i \parallel RPW)$ or not. If this condition is hold, SC continues to go next step. Otherwise, SC terminates this session.
3. Generate a random value r_C. Then SC computes $R_C = g^{r_C} \bmod p$, $C_1 = J \oplus R_C$, $AID_i = ID_i \oplus h(R_C)$ and $C2 = h(J \parallel ID_i \parallel R_C)$
4. Finally, SC sends $\{AID_i,\ C_1,\ C_2,\ e\}$ to S through a common channel.

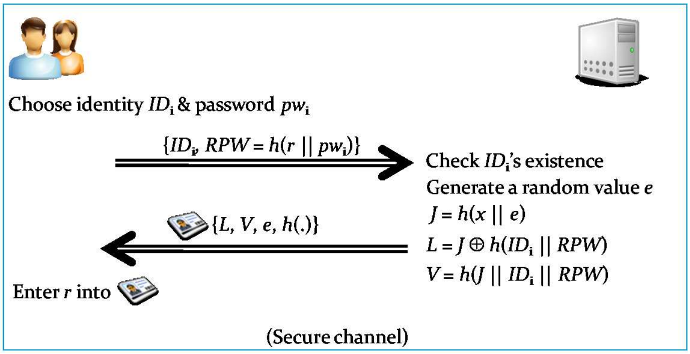

Fig. 3. Proposed registration phase

3.3 Mutual Authentication Phase

S receives U_i's login message $\{AID_i,\ C_1,\ C_2,\ e\}$ and performs following steps (Illustrated in Figure 4).

1. S computes $R_C = h(x \| e) \oplus C_1$ and uncovers $ID_i = AID_i \oplus h(R_C)$. Then, S checks identity's validity. If this identity is not valid, S terminates this session. Otherwise, S goes to next step.
2. S continues to verify if $C_2 \mathrel{?=} h(h(x \| e) \| ID_i \| R_C)$ or not. If this condition does not hold, S terminates this session. Otherwise, S generates a random value r_S and computes $R_S = g^{r_S} \bmod p$, $C_3 = h(R_C) \oplus R_S$, $C_4 = h(R_C \| R_S \| h(x \| e) \| ID_i)$.
3. S sends $\{C_3,\ C_4\}$ to U_i through a common channel.
4. After receiving $\{C_3,\ C_4\}$ from S. U_i computes $R_S = C_3 \oplus h(R_C)$ and check if $C_4 \mathrel{?=} h(R_C \| R_S \| J \| ID_i)$. If this condition does not hold, U_i terminates the session. Otherwise, U_i successfully authenticates S. U_i sends $C_5 = h(R_C \| R_S)$ to S through a common channel and computes $SK = (R_S)^{r_C}$.
5. When receiving $\{C_5\}$ from U_i, S checks if $C_5 \mathrel{?=} h(R_C \| R_S)$. If this condition does not hold, S terminates the session. Otherwise, S successfully authenticates U_i. And S also computes $SK = (R_C)^{r_S}$.

3.4 Password Update Phase

When U_i changes password pw_i. He can perform following steps:

1. Insert SC into card reader, enter ID_i, pw_i and choose a new password pw_{inew}.
2. SC computes $RPW = h(r \| pw_i)$ and $J = L \oplus h(ID_i \| RPW)$. Then, SC checks if $V \mathrel{?=} h(J \| ID_i \| RPW)$. If this condition does not hold, SC terminates the session. Otherwise, SC computes $L_{new} = J \oplus h(ID_i \| RPW_{new})$ and $V_{new} = h(J \| ID_i \| RPW_{new})$, where $RPW_{new} = h(r \| pw_{inew})$.
3. Finally, SC replaces L with L_{new}, V with V_{new}.

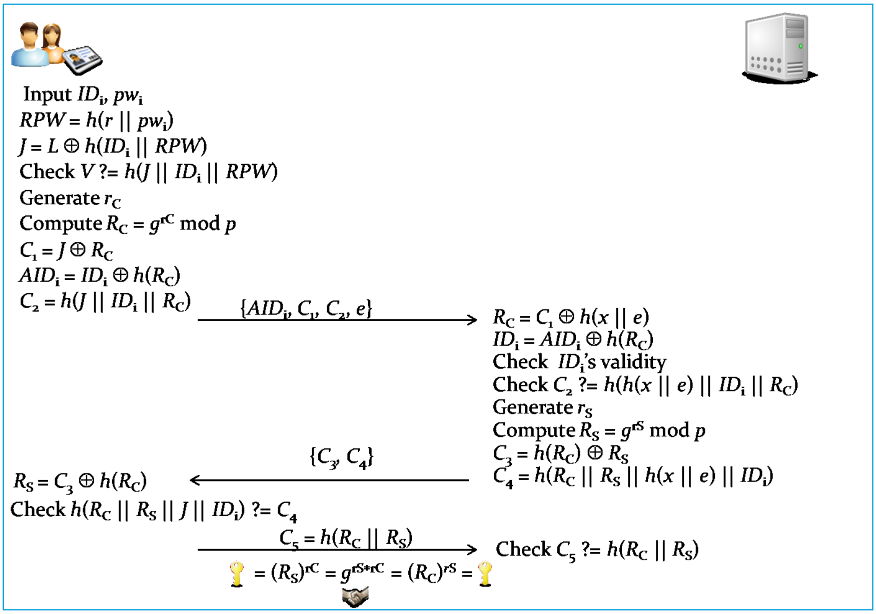

Fig. 4. Proposed login and authentication phases

4 Security and Efficiency Analysis

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

4.1 Security Analysis

In this subsection, we present security analyses and show that proposed scheme can resist many kinds of attacks. Assuming that wireless communication is insecure and that there exists an adversary, he has capability to intercept or hear all messages transmitted between server and user. Besides, we assume that the adversary can steal smart-card's information.

Forgery Attack. In our scheme, attacker cannot impersonate user because he fails to compute authentication key J, random value R_C or identity ID_i. Similarly, attacker cannot impersonate server to cheat other users because he also fails to compute random value R_S. In our design, we use a random value r instead of distributing common secret information to all users. Hence, each user has a different key at different time and other users have no chance to exploit constant clues in their smart cards. Finally, proposed scheme can completely resist forgery attack.

Password Guessing Attack. In our scheme, we also use random value r with user's password in cryptographic hash function. Therefore, attacker is hard to exactly guess user's password from hashing-output. Furthermore, in our design, masked password RPW does not participate in login and authentication phases. Hence, attacker has no chance to computes anything from $\{AID_i, C_1, C_2, e\}$ and $\{C_3, C_4\}$. Ultimately, proposed scheme successfully withstands this kind of attack.

User Anonymity. In our scheme, user's identity is concealed by using user's random value R_C. To compute this random value, attacker needs to have user's authentication key $J = h(x \parallel e)$. Clearly, knowing server's master key x is impossible. Attacker can exploit login-message $\{AID_i, C_1, C_2, e\}$ and response-message $\{C_3, C_4\}$ to figure something about J. We see that there are three clues: AID_i, C_1 and C_3 which can be used to compute. However, attacker cannot find anything from them and our scheme completely protects user's anonymity.

Session-Key Agreement. Our proposed scheme provides session key agreement during the authentication phase. Session key $SK = (R_S)^{r_C} \bmod p = (R_C)^{r_S} \bmod p = g^{r_S * r_C} \bmod p$ is shared between U_i and S. Apparently, even R_C and R_S are revealed due to leaking master key x, attacker cannot compute this key because he must face Diffie - Hellman problem. Consequently, our scheme not only provides session key agreement but also satisfies perfect forward secrecy considered in some schemes using elliptic curve [7, 8, 9, 10].

Table 1. A comparison between our scheme & Y-H An's for withstanding various attacks

Kinds of Attacks \ Schemes	Y-H An's	Ours
Forgery attack	No	Yes
Password guessing attack	Yes	Yes
User anonymity	No	Yes
Secret key forward secrecy	Yes	Yes
Session key agreement	Yes	Yes

4.2 Efficiency Analysis

To compare efficiency between our scheme and Y-H An's, we reuse the method which is used in Y-H An's scheme to appraise computational complexity. That is, we ignore cost of concatenation and XOR operations because they are not time or energy-consuming operations. Additionally, we let T_h denote the computing-time of one-way hash function. Y-H An's scheme needs $3 \times T_h$ in registration phase, and $9 \times T_h$ in login and authentication phases. Our scheme needs $4 \times T_h$ in registration phase and $12 \times T_h$ in login and authentication phases.

Table 2. A comparison of computation cost

Schemes \ Phases	Registration	Login & Authentication
Y-H An's	$3 \times T_h$	$9 \times T_h$
Ours	$4 \times T_h$	$12 \times T_h$

Because our scheme and Y-H An's are based on smart-card, we compare the storage capacity of smart-card. To do that, we assume output hash function is 160 bit long, for example SHA-I. Moreover, we also would like to consider communication cost between user and server in term of authentication in two schemes. In Table 3, we see that the bits in smart-card of our schemes and Y-H An's are the same. Besides, in authentication phase, our scheme also only needs the same bits as Y-H An's scheme.

Table 3. A comparison of communication cost and storage capacity

Capacity & Communication \ Schemes	Y-H An's	Ours
Bits in smart-card	800	800
Bits in authentication	1120	1120

5 Conclusions

In this paper, we re-consider Y-H An's scheme. Although his scheme can withstand some attacks, we see that his scheme is still vulnerable to server impersonation attack. Besides, his scheme does not provide user's anonymity. Consequently, we propose an improved scheme to eliminate such problems. With our improvement, the schemes can be applied in many applications, especially in financial area.

In the near future, we intend to employ visual cryptography with elliptic curve to store user's authentication key. By approaching this new method, authentication scheme can easily store authentication key in smart-card and user does not have to remember this long key. Therefore, the scheme has been more secure and convenient than previous schemes.

References

[1] Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24, 770–772 (1981)

[2] 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)

[3] 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)
[4] 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)
[5] Khan, 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)
[6] An, Y.-H.: Security improvements of dynamic id-based remote user authentication scheme with session key agreement. In: 2013 15th International Conference on Advanced Communication Technology (ICACT), pp. 1072–1076 (2013)
[7] Yang, J.-H., Chang, C.-C.: An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Computers and Security 28(3-4), 138–143 (2009)
[8] Yoon, E.-J., Yoo, K.-Y.: Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ecc. In: IEEE International Conference on Computational Science and Engineering, vol. 2, pp. 633–640 (2009)
[9] Islam, S.H., Biswas, G.P.: A more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Journal of Systems and Software 84(11), 1892–1898 (2011)
[10] Debiao, H., Jianhua, C., Jin, H.: An id-based client authentication with key agreement protocol for mobile clientserver environment on ecc with provable security. Information Fusion (2011)