Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080305

Modified Dynamic ID-based User Authentication Scheme Resisting Smart-Card-Theft Attack

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

[1] University of Science, VNU-HCM, Ho Chi Minh city, Vietnam
[2] University of Information Technology, VNU-HCM, Ho Chi Minh city, Vietnam

Received: 7 May. 2013, Revised: 11 Sep. 2013, Accepted: 12 Sep. 2013
Published online: 1 May. 2014

Abstract: Wireless environments such as GSM, 3G, and 4G are more and more popular. Consequently, communications in such networks need to be guarded. It is necessary to have a secure mutual authentication scheme to defend transactions between user and service provider against illegitimate adversaries. Especially, users are those vulnerable to attacks and there are many authentication schemes with smart-card proposed to protect them. Recently, Yung-Cheng Lee has suggested a dynamic identity based user authentication scheme to resist smart-card-theft attack. Nevertheless, he assumed that smart-card is tamperproof. In our opinion, this is not appropriate because Kocher and Messerges pointed that smart-card's confidential information could be extracted by physically monitoring its power consumption. Therefore, design of Yung-Cheng Lee cannot withstand this kind of attack. In addition, anyone who is a legal member can masquerade server or other legal users in his scheme. Moreover, legitimacy verification only starting from server side truly makes Lee's scheme be impractical. In this paper, we present an improvement to his scheme to isolate such problems.

Keywords: Authentication, Password, Dynamic ID, Smart card, Impersonation, Session key

1 Introduction

In network environments, remote authentication schemes play an important role in communicating between partners because it keeps faith and security. Schemes not only must prevent legal users and servers from attacks of illegitimate adversaries, but they also defend legal partners against impersonating to cheat each other.

There are many methods of satisfying above requirements. And one of the approaches many schemes have used is password authentication which has many advantages such as simplicity, efficiency, and convenience. Nonetheless, many schemes [1,2,3,4,5] based on password apply static identity, which is easy to leaking information to attackers. One solution to identity theft is making it change for each login. Later, a number of paper [6,7,20,8,16] have put forward many ideas to protect user anonymity by using random value or time-stamp to vary user identity for each session. However, these schemes issue a smart-card for each user and assume that the contents of smart-card cannot be revealed. This is not practical because users can lost or be stolen smart-card. So, when attackers have smart-card, they completely have capability to impersonate users.

In 2004, Das et al proposed a dynamic ID-based remote user authentication scheme using smart cards [10]. 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.

Recently, Yung-Cheng Lee proposed a new dynamic ID-based user authentication scheme to resist smart-card-theft attack [13] and pointed out that scheme of Das et al is vulnerable to guessing and impersonation attacks. He claimed that his scheme enhanced the security because of using dynamic identity feature. Furthermore, he also stated that his scheme can completely resist smart-card-theft attack. In this paper, we prove that his scheme cannot suffer from smart-card-theft attack. Furthermore, it also cannot withstand masquerading attack. Finally, we see that his scheme does not provide mutual authentication and session-key exchange phase. Eventually, we propose an improved version of Lee' scheme in order to recover all problem mentioned.

* Corresponding author e-mail: ttthinh@fit.hcmus.edu.vn

The remainder of this paper is organized as follows: section 2 quickly reviews Yung-Cheng Lee'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 Yung-Cheng Lee's Scheme

In this section, we review Lee's new dynamic ID-based user authentication scheme to resist smart-card-theft attack [9] and show that his scheme is vulnerable to impersonation attack, smart-card-theft attack. Furthermore, it does not provide mutual authentication.

2.1 Review of Yung-Cheng Lee's Scheme

In this subsection, we review Yung-Cheng Lee's scheme. Their scheme includes three phases: registration, authentication and password update phases. Some important notations in this scheme are listed as follow:

–U_i: a qualified user.
–PW_i: Unique password of U_i.
–S: The remote server that users log in.
–x: The secret key of the remote server.
–$h(.)$: A cryptographic one-way hash function.
–T: The timestamp.
–DID_i: user's dynamic identity.
–SC: the smart card.
–$\oplus$: The exclusive-or operation.
–$A \Rightarrow B$: M: A sends M to B via a secure channel.
–$A \rightarrow B$: M: A sends M to B via a public channel.

2.1.1 Registration Phase

When U_i wants to access resource of S, he or she has to submit his or her PW_i to server through a secure channel. Then, S performs the following steps. Figure 1 illustrates the steps of the registration phase.

–S computes $N_i = h(PW_i) \oplus h(x)$.
–S installs $\{h(.), N_i, h(x)\}$ into SC and issues it to U_i through a secure channel.

In registration phase, we see that user freely chooses PW_i. However, his scheme does not apply identity to participate into registration process. Furthermore, sharing a common $h(x)$ for every member makes this phase be weak because they can exploit to impersonate to cheat each other. To overcome these weak points, we use identity, supply a random value e to make different for each time of registering and do not share the secret key $h(x)$ for all users.

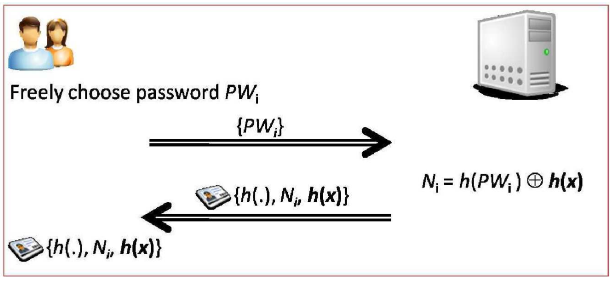

Fig. 1: Yung-Cheng Lee's registration phase

2.1.2 Login Phase

After receiving secret information from S, U_i can use SC when he or she wants to login to S.

–U_i inserts SC into card-reader of another terminal. Then he or she keys PW_i.
–SC generates a random value R and computes $DID_i = h(PW_i) \oplus h(N_i \oplus h(x) \oplus R)$, $B_i = h(N_i \oplus h(N_i \oplus h(x) \oplus R))$ and $C_i = h(B_i \oplus h(x) \oplus T)$.
–$U_i \rightarrow S$: DID_i, C_i, T. The U_i sends the login message to S through common channel.

In login phase, we see that user generates a random value R to make login message be renewed for each login. However, this is also the drawback because R does not participate to challenge S. In the next phase, we see that S does not need to know what R is, yet S still authenticates U_i. So, we will fix this weak point of his phase.

2.1.3 Verification Phase

After receiving the login request sent from U_i, S performs the following tasks to authenticate the user's login request. Figure 2 illustrates the steps of login and verification phase.

–On receiving the login request $\{DID_i, C_i, T\}$ from U_i, S checks T to determine its validity. If T is within an expected time interval, S accepts the login request; otherwise, it terminates the request.
–S computes $B_i = h(DID_i \oplus h(x))$.[1]
–S computes $C_i' = h(B_i \oplus h(x) \oplus T)$ and checks if the received C_i is equal to C_i'. If this condition holds, S accepts the login request; otherwise, it terminates the session.

In verification phase, we see that S does not generate any random value to re-challenge U_i. Furthermore, S also does not prove its validity to U_i. So, S and U_i cannot know whether server and user communicating are legal or not. At this point we use user three-way challenge-response handshake technique to recover. With that technique, S can know user's legitimation.

[1] $B_i = h(DID_i \oplus h(x)) = h(h(PW_i) \oplus h(N_i \oplus h(x) \oplus R) \oplus h(x)) = h(N_i \oplus h(N_i \oplus h(x) \oplus R))$

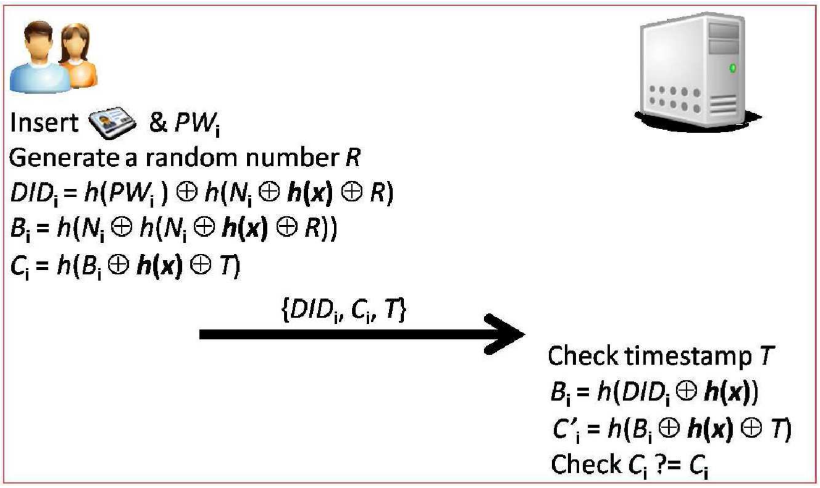

Fig. 2: Yung-Cheng Lee's login and verification phase

2.1.4 Password Update Phase

In this phase, U_i can change his or her password anytime when he or she wants. Figure 3 illustrates the steps of the password change phase.

– U_i inserts SC into card-reader and inputs PW_i.
– U_i chooses a new password PW_{inew}.
– The SC computes $N_{inew} = N_i \oplus h(PW_i) \oplus h(PW_{inew})$. [2]
– SC replaces N_i with N_{inew}. So, user can log into the system by using PW_{inew}

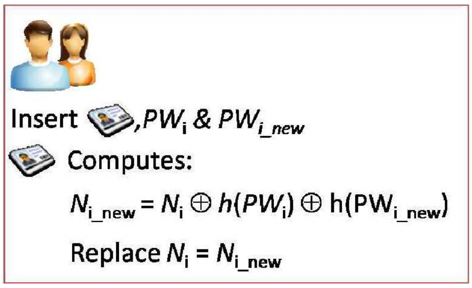

Fig. 3: Yung-Cheng Lee's password update phase

In password update phase, we see that only legal users can change password because this proceduce needs PW_i of users. In our scheme, we also inherit this idea basically.

2.2 Cryptanalysis of Yung-Cheng Lee's Scheme

In this subsection, we present our results on Yung-Cheng Lee's scheme. We will show that his scheme is vulnerable to impersonation and smart-card-theft attacks. Besides, his scheme needs to be supplied mutual authentication and session-key exchange phases.

[2] $N_{inew} = h(PW_{inew}) \oplus h(x)$

2.2.1 Impersonation Attack

In Yung-Cheng Lee's scheme, we see that anyone being a valid member can know $h(x)$. Hence, with $h(x)$, valid users can impersonate other users even the server.

– Firstly, another legal user A can perform following steps to be a valid server.
  – After receiving $\{DID_i, C_i, T\}$ from another user, A computes $B_i = h(DID_i \oplus h(x))$
  – A continues to compute $C_i^{'} = h(B_i \oplus h(x) \oplus T)$
  – Finally, A can check C_i and $C_i^{'}$. Certainly, A does not have to do this. Clearly, A has ability to be a valid server.
– Secondly, another legal user A can perform following steps to be another valid user.
  – A can capture any login message $\{DID_i, C_i, T\}$. Then, A computes $h(x) \oplus DID_i$ to obtain B_i of another user.
  – Next, A computes $C_i^{*} = h(B_i \oplus h(x) \oplus T^{*})$, where T^{*} is the current timestamp.
  – Finally, A sends $\{DID_i, C_i^{*}, T^{*}\}$ to server S. Clearly, this is a completely valid login message.

2.2.2 Smart-card-theft Attack

In Yung-Cheng Lee's scheme, we see that losing smart-card is very dangerous because it contains $\{h(.), N_i, h(x)\}$. If anyone being a valid member picks smart-card, attacker A easily extracts $h(PW_i)$ by performing $N_i \oplus h(x)$. Next, A can perform some steps to impersonate victim.

– A generates a random value R^* and computes the dynamic DID_i by: $DID_i = h(PW_i) \oplus h(N_i \oplus h(x) \oplus R)$, where PW_i and N_i belongs to victim.
– A computes $B_i = h(N_i \oplus h(N_i \oplus h(x) \oplus R^*))$.
– Next, A computes $C_i = h(B_i \oplus h(x) \oplus T^*)$, where T^* is the current timestamp.
– Finally, A sends $\{DID_i, C_i, T^*\}$ to server S. Obviously, this is the valid login message.

2.2.3 Mutual Authentication & Session Key Agreement Phases

In Yung-Cheng Lee's scheme, we see that only server can verify user's validity. This is not fair because user can communicate with another illegal server. So, we need server proves its validity to user. Furthermore, after successfully authenticating, transmitting data between server and user is necessary. Therefore, we need to supply a sub-step of sharing a common session-key.

3 Proposed Scheme

In this section, we propose an improved version of Yung-Cheng Lee's scheme. Our scheme removes the security problems depicted in the previous sections. Our scheme not only inherits the advantages of his scheme, it also enhances the security.

Before coming to each phase, we will present general ideas in our scheme more detailed. In registration phase, our main goal is obtaining $h(ID_i \oplus h(x \parallel e))$. Random value e assists to withstand re-registration of attackers, with the same identity but various authentication keys at different time. In login and authentication phases, we use two random values R_U and R_S for user and server for challenging each other. Besides, we employ three-way challenge-response handshake technique to resist replay or impersonation attacks instead of using timestamp. And it is very important to have the same session-key for user and server after verification step.

Our scheme is also divided into the four phases of registration, login, mutual authentication and password change phases. Some important notations in our scheme are listed as follow:

– U_i: a qualified user.
– ID_i: Unique identity of U_i.
– PW_i: Unique password of U_i.
– N: The nonce chosen by user in registration phase.
– S: The remote server that users log in.
– x: The secret key of the remote server.
– e: The nonce chosen by server in registration phase.
– $h(.)$: A cryptographic one-way hash function.
– R_U: The nonce chosen by user.
– R_S: The nonce chosen by server.
– CID_i: user's dynamic identity.
– SK: session-key of user and server.
– SC: the smart card.
– $\oplus$: The exclusive-or operation.
– $\parallel$: The concatenation operation.
– $A \Rightarrow B$: M: A sends M to B via a secure channel.
– $A \rightarrow B$: M: A sends M to B via a public channel.

3.1 Registration Phase

Before we present this phase, we enumerate three requirements for a registration phase: secrecy for information transmitted between user and server, the true password of user must not be leaked to anyone even the server, and difference between secret keys provided for each time of registration by server. Easily, we see that Yung-Cheng Lee's scheme achieved the first requirement but not the last. So, we cover these points to have a good registration phase.

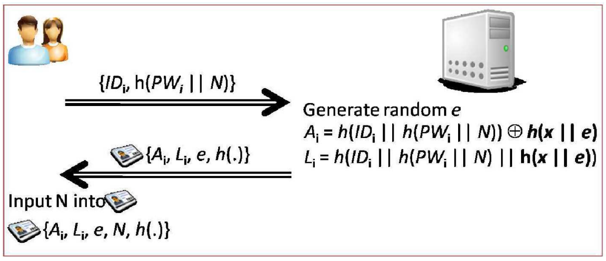

Fig. 4: Proposed registration phase

When U_i wants to register to S, he or she has to submit his or her ID_i, $h(PW_i \parallel N)$. After receiving $\{ID_i, h(PW_i \parallel N)\}$ from user via a secure channel, S performs following steps. Figure 4 illustrates the steps of the registration phase.

1. Generating a random value e.
2. Computing $A_i = h(ID_i \parallel h(PW_i \parallel N)) \oplus h(x \parallel e)$.
3. Computing $L_i = h(ID_i \parallel h(PW_i \parallel N) \parallel h(x \parallel e))$.
4. S sends SC containing $\{A_i, L_i, e, h(.)\}$ to U_i via a secure channel.
5. U_i receives SC and inputs N into SC.

3.2 Login Phase

U_i inserts SC into card-reader and ID_i and PW_i to login to S, and then the SC performs the following steps:

1. Computing $h(x \parallel e) = A_i \oplus h(ID_i \parallel h(PW_i \parallel N))$ and cheking if L_i is equal to $h(ID_i \parallel h(PW_i \parallel N) \parallel h(x \parallel e))$. If this condition holds, SC continues to go next step; otherwise, it terminates the session.
2. Generating R_U and computing $CID_i = ID_i \oplus R_U$.
3. Computing $B_i = h(x \parallel e) \oplus R_U$ and $C_i = h(ID_i \parallel R_U \parallel h(x \parallel e))$.
4. Finally, U_i sends $\{CID_i, B_i, C_i, e\}$ to S.

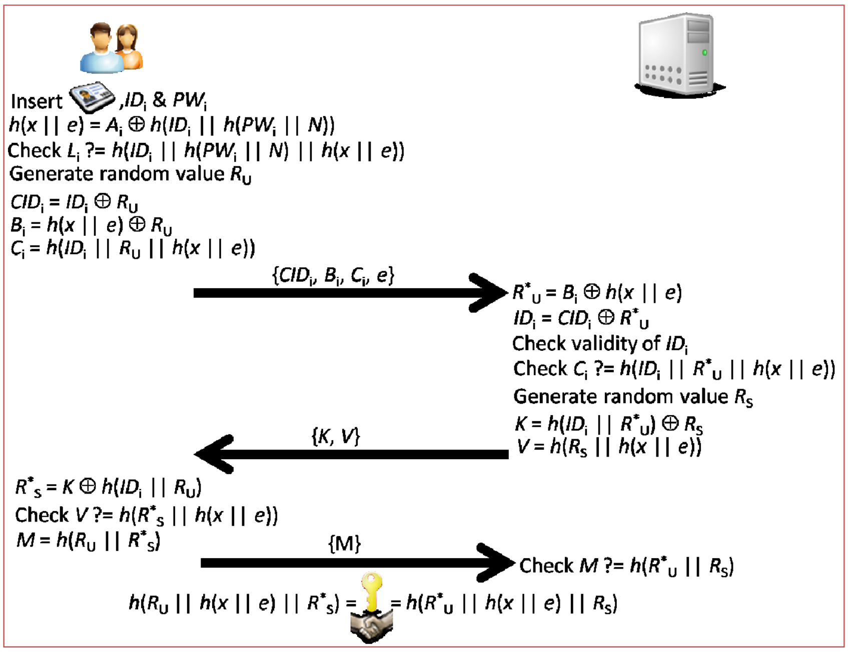

Fig. 5: Proposed login, mutual authentication and session key agreement phase

3.3 Mutual Authentication And Session Key Agreement Phase

Likewise, we also list three requirements helping authentication be more security: User must employ a random value to challenge server. Server must use a random value to re-challenge user. User and server share a secret session-key. In Yung-Cheng Lee's scheme, only user use a random value to make login-message be dynamic but not to challenge server and server also do not re-challenge user. Besides, no session-key is generated after authenticating successfully. Our phase will fix these weak points.

In this section, S receives the login request message (CID_i, B_i, C_i, e) from U_i in the login phase and performs some following steps. Figure 5 illustrates the steps that S authenticates U_i.

1. Computing $R^*_U = B_i \oplus h(x \,\|\, e)$.
2. Extracting $ID_i = CID_i \oplus R^*_U$. Then, S checks ID_i's validity. If this is a valid identity, S continues going to next step. Otherwise, S rejects the login message.
3. S checks whether C_i is equal to $h(ID_i \,\|\, R^*_U \,\|\, h(x \,\|\, e))$. If this condition is true, S goes to next step. Otherwise, S terminates the session.
4. Generating R_S and computing $K = h(ID_i \,\|\, R^*_U) \oplus R_S$, $V = h(R_S \,\|\, h(x \,\|\, e))$.
5. Sending $\{K, V\}$ to U_i via a common channel.
6. After receiving $\{K, V\}$ from S, U_i computes $R^*_S = K \oplus h(ID_i \,\|\, R_U)$.
7. U_i checks whether V is equal to $h(R^*_S \,\|\, h(x \,\|\, e))$. If this condition holds, U_i authenticates S successfully. Otherwise, U_i terminates the session.
8. U_i computes $M = h(R_U \,\|\, R^*_S)$ and sends M to S via a common channel.
9. S checks whether M is equal to $h(R^*_U \,\|\, R_S)$. If this condition is true, S authenticates U_i successfully. Otherwise, S terminates the session.
10. After authenticating successfully, S computes shared $SK = h(R^*_U \,\|\, h(x \,\|\, e) \,\|\, R_S)$ and U_i computes shared $SK = h(R_U \,\|\, h(x \,\|\, e) \,\|\, R^*_S)$.

3.4 Password Update Phase

When U_i wants to change PW_i. He or she can perform following steps:

- Insert SC into card-reader, inputs ID_i, PW_i and choose a new password PW_{inew}.
- SC computes $h(x \,\|\, e) = h(ID_i \,\|\, h(PW_i \,\|\, N)) \oplus A_i$ and $L^*_i = h(ID_i \,\|\, h(PW_i \,\|\, N) \,\|\, h(x \,\|\, e))$.
- SC checks whether L_i is equal to L^*_i. If this condition is false, SC terminates this phase. Otherwise, it goes to next step.
- SC computes $A_{inew} = h(x \,\|\, e) \oplus h(ID_i \,\|\, h(PW_{inew} \,\|\, N))$ and $L_{inew} = h(ID_i \,\|\, h(PW_{inew} \,\|\, N) \,\|\, h(x \,\|\, e))$.
- Finally, SC replaces L_i with L_{inew}, A_i with A_{inew}.

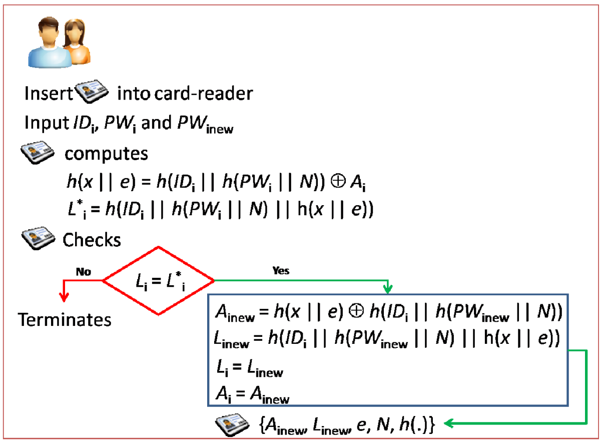

Fig. 6: Proposed password update phase

4 Security and Efficiency Analysis

In this section, we review our scheme and analyze it on two aspects: security and efficiency. Our scheme includes four phases, registration, login, authentication and session-key agreement, and password change phases. Firstly, we summarize all phases of our scheme.

–Registration phase: U_i sends $\{ID_i, h(PW_i \parallel N)\}$. S returns SC containing $\{A_i, L_i, e, h(.)\}$, where $A_i = h(ID_i \parallel h(PW_i \parallel N)) \oplus h(x \parallel e)$, $L_i = h(ID_i \parallel h(PW_i \parallel N) \parallel h(x \parallel e))$ and e is chosen by S. U_i receives SC and inputs N into it.

–Login phase: U_i inserts SC, ID_i and PW_i. Then, SC extracts $h(x \parallel e)$ by performing $A_i \oplus h(ID_i \parallel h(PW_i \parallel N))$. SC verifies whether L_i is equal to $h(ID_i \parallel h(PW_i \parallel N) \parallel h(x \parallel e))$. If this condition holds, SC goes to next step. Otherwise, it terminates the session. Next, SC generates R_U and computes CID_i, B_i and C_i. Finally, U_i sends $\{CID_i, B_i, C_i, e\}$ to S.

–Authentication and session-key agreement phase: After receiving $\{CID_i, B_i, C_i, e\}$ from U_i. S computes to obtain R^*_U. Afterward, S extracts ID_i and checks its validity. Next, S computes $h(ID_i \parallel R^*_U \parallel h(x \parallel e))$ and compares it with C_i received. If this condition is true, S goes to next step. Otherwise, S terminates the session. Next, S generates R_S and computes K and V. Finally, S sends $\{K, V\}$ back to U_i. When U_i receives this package, U_i re-computes R_S and checks whether V equals to $h(R^*_S \parallel h(x \parallel e))$. If this condition holds, U_i accepts S. Otherwise, U_i rejects S. Finally, U_i sends $M = h(R_U \parallel R^*_S)$ to S. S receives M and checks if M is equal to $h(R^*_U \parallel R_S)$. If this condition holds, S accepts U_i. Otherwise, S rejects U_i. After authenticating successfully, U_i and S shares common $SK = h(R_U \parallel h(x \parallel e) \parallel R_S)$.

–Password change phase: At the beginning of this phase, U_i performs steps similar to login phase's steps. After logining successfully, SC computes A_{inew} and L_{inew}. Finally, SC replaces A_{inew} with A_i, L_{inew} with L_i.

4.1 Security Analysis

In this section, we apply BAN logic, the tool for formally analyzing authentication schemes. BAN-logic uses three objects: principals, encryption keys, and formulas (also called statements for identifying message with statement). Similarly to Burrow [21], we let symbols P and Q be principals, X and Y range over statements, and K represent the cryptographic key. We only use some notations used in BAN-logic for our demonstration.

–$P \mid\equiv X$: P believes X (central construct).
–$P \triangleleft X$: P received a message including X.
–$P \mid\sim X$: P once said X.
–$P \Rightarrow X$: P has jurisdiction over X. (Used when principal has delegated authority over some statement).
–$\#(X)$: X is fresh, that is, no principal sent X in a message before the current run of the protocol.
–$P \overset{K}{\leftrightarrow} Q$: P and Q communicate using shared K. Moreover, K will never be discovered by any principal except P and Q, or a principal trusted by either P or Q.
–X_K: This stands for X encrypted under the K.
–$<X>_Y$: This stands for X combined with Y.
–SK: This session key used in the current round.

Besides, we present some main logical BAN-logics postulates for proving our scheme.

–Message meaning rule: $\frac{P|\equiv P\overset{K}{\leftrightarrow}Q, P\triangleleft\{X\}_K}{P|\equiv Q|\sim X}$
–Nonce verification rule: $\frac{P|\equiv\#(X), P|\equiv(Q|\sim X)}{P|\equiv Q|\equiv X}$
–Jurisdiction rule: $\frac{P|\equiv Q\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$
–Freshness rule: $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$
–Believe rule: $\frac{P|\equiv Q|\equiv(X,Y)}{P|\equiv Q|\equiv X}$, $\frac{P|\equiv X, P|\equiv Y}{P|\equiv(X,Y)}$

All authentication schemes need achieving main eight goals. We use U and S represent for user and server in scheme.

–**G$_1$**: $U \mid\equiv U \overset{ID}{\leftrightarrow} S$
–**G$_2$**: $U \mid\equiv S \mid\equiv U \overset{ID}{\leftrightarrow} S$
–**G$_3$**: $S \mid\equiv U \overset{ID}{\leftrightarrow} S$
–**G$_4$**: $S \mid\equiv U \mid\equiv U \overset{ID}{\leftrightarrow} S$
–**G$_5$**: $U \mid\equiv U \overset{SK}{\leftrightarrow} S$
–**G$_6$**: $U \mid\equiv S \mid\equiv S \overset{SK}{\leftrightarrow} U$
–**G$_7$**: $S \mid\equiv S \overset{SK}{\leftrightarrow} U$
–**G$_8$**: $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$

Now we use the BAN-logic to show proposed scheme can obtain mutually authentication with dynamic identity. Furthermore, our scheme can exchange a common session key SK

1. We idealize our scheme.
   - $\boldsymbol{CID_i} = <U \overset{ID}{\leftrightarrow} S, R_U>$
   - $\boldsymbol{B_i} = <U \overset{h(x\|e)}{\leftrightarrow} S, R_U>$
   - $\boldsymbol{C_i} = <R_U, U \overset{ID}{\leftrightarrow} S, U \overset{h(x\|e)}{\leftrightarrow} S>$
   - $\boldsymbol{K} = <R_S, R_U, U \overset{ID}{\leftrightarrow} S>$
   - $\boldsymbol{V} = <R_S, R_U, U \overset{h(x\|e)}{\leftrightarrow} S>$
   - $\boldsymbol{M} = <R_U, R_S>$
2. We write the assumptions about the initial state.
   - $\mathbf{A_1}$: $U \mid\equiv U \overset{ID}{\leftrightarrow} S$
   - $\mathbf{A_2}$: $U \mid\equiv U \overset{h(x\|e)}{\leftrightarrow} S$
   - $\mathbf{A_3}$: $U \mid\equiv S \Rightarrow U \overset{SK}{\leftrightarrow} S$
   - $\mathbf{A_4}$: $S \mid\equiv U \Rightarrow U \overset{ID}{\leftrightarrow} S$
   - $\mathbf{A_5}$: $S \mid\equiv U \Rightarrow U \overset{SK}{\leftrightarrow} S$
   - $\mathbf{A_6}$: $S \mid\equiv S \overset{h(x\|e)}{\leftrightarrow} U$
   - $\mathbf{A_7}$: $U \mid\equiv \#(R_S)$
   - $\mathbf{A_8}$: $S \mid\equiv \#(R_U)$
3. We analyze our schemes idealized form based on the BANlogic rules and the assumptions.
   - Because U registers ID with S, we have the first goal $U \mid\equiv U \overset{ID}{\leftrightarrow} S$.
   - Using $\mathbf{A_6}$ and the message $\mathbf{C}_i$, we apply the message-meaning rule to derive $S \mid\equiv U \mid\sim <R_U, U \overset{ID}{\leftrightarrow} S, U \overset{h(x\|e)}{\leftrightarrow} S>$ (1)
   - Using $\mathbf{A_8}$, we apply freshness rule to infer $S \mid\equiv \#<R_U, U \overset{ID}{\leftrightarrow} S, U \overset{h(x\|e)}{\leftrightarrow} S>$ (2)
   - Using (1) and (2), we apply nonce - verification rule to derive $S \mid\equiv U \mid\equiv <R_U, U \overset{ID}{\leftrightarrow} S, U \overset{h(x\|e)}{\leftrightarrow} S>$ (3)
   - Using (3), we apply believe rule to derive $S \mid\equiv U \mid\equiv U \overset{ID}{\leftrightarrow} S$ ($\boldsymbol{G_4}$)
   - Using $\mathbf{G_4}$ and $\mathbf{A_4}$, we apply jurisdiction rule to infer $S \mid\equiv U \overset{ID}{\leftrightarrow} S$ ($\boldsymbol{G_3}$)
   - Using $\mathbf{A_2}$ and $\boldsymbol{K}$, we apply the message-meaning rule to derive $U \mid\equiv S \mid\sim <R_S, R_U, U \overset{ID}{\leftrightarrow} S>$ (4)
   - Using (4) and $\mathbf{A_7}$, we apply freshness rule to derive $U \mid\equiv \#<R_S, R_U, U \overset{ID}{\leftrightarrow} S>$ (5)
   - Using (4) and (5), we apply nonce - verification rule to derive $U \mid\equiv S \mid\equiv <R_S, R_U, U \overset{ID}{\leftrightarrow} S>$ (6)
   - Using (6), we apply believe rule to derive $U \mid\equiv S \mid\equiv U \overset{ID}{\leftrightarrow} S$ ($\boldsymbol{G_2}$)
4. With goal 1, 2, 3 and 4, we achieve that both S and U believe the other believes the identity. That is, U and S mutually authenticate with dynamic identity. Now we prove U and S can exchange SK.
   - Using $\boldsymbol{V}$ and $\mathbf{A_2}$, we apply the message-meaning rule to derive $U \mid\equiv S \mid\sim <R_S, U \overset{h(x\|e)}{\leftrightarrow} S>$ (7)
   - Using $\mathbf{A_7}$ and $\boldsymbol{V}$, we apply freshness rule to derive $U \mid\equiv \#<R_S, U \overset{h(x\|e)}{\leftrightarrow} S>$ (8)
   - Using (7) and (8), we apply nonce - verification rule to derive $U \mid\equiv S \mid\equiv <R_S, U \overset{h(x\|e)}{\leftrightarrow} S>$ (9)
   - Using (9), we apply believe rule to derive $U \mid\equiv S \mid\equiv S \overset{SK}{\leftrightarrow} U$ ($\boldsymbol{G_6}$)
   - Using $\mathbf{A_3}$ and $\mathbf{G_6}$, we apply jurisdiction rule to obtain $U \mid\equiv U \overset{SK}{\leftrightarrow} S$ ($\boldsymbol{G_5}$)
   - Using $\boldsymbol{M}$ and $\mathbf{A_6}$, we apply the message-meaning rule to derive $S \mid\equiv U \mid\sim <R_U, R_S>$ (10)
   - Using $\boldsymbol{M}$ and $\mathbf{A_8}$, we apply freshness rule to derive $S \mid\equiv \#<R_U, R_S>$ (11)
   - Using (10) and (11), we apply the nonce - verification rule to derive $S \mid\equiv U \mid\equiv <R_U, R_S>$ (12)
   - Using (12) and $\mathbf{A_6}$, we apply believe rule to infer $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$ ($\boldsymbol{G_8}$)
   - Using (12) and $\mathbf{A_5}$, we apply message-meaning rule to infer $S \mid\equiv <R_U, R_S>$ (13)
   - Using (13), we apply believe rule to derive $S \mid\equiv S \overset{SK}{\leftrightarrow} U$ ($\boldsymbol{G_7}$)
5. With goal 5, 6, 7 and 8, we achieve both S and U believe the other believes SK is shared between them.

4.2 Other Discussions

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

4.2.1 Replay Attack

The replay attack is replaying the same message of the receiver or the sender again. We use nonce and three-way challenge-response handshake technique instead of timestamp to withstand replay attacks. For example, an attacker A re-uses $\{CID_i, B_i, C_i, e\}$ to re-send to S. When S sends $\{K, V\}$ back to A, A is not capable of computing M because A has no information about R_U and R_S. So, A cannot replay with $\{CID_i, B_i, C_i, e\}$. Now, if A re-uses $\{K, V\}$ of server S, user will recognize this is a replay message because verification of V does not hold. It is said that our scheme can resist replay attack.

4.2.2 Impersonation Attack

In our scheme, we use use nonce and three-way challenge-response handshake technique. Therefore, it is difficult for attackers to impersonate user or server. For example, if attacker A wants to fake legal user, A has to compute M to re-send to S. So, it is impossible for A to perform that task because A has no idea about R_U and R_S. Now, if A wants to fake legal S, A must have information about x of server and random value R_U of user. Clearly, this is impossible mission. It is said that our scheme can withstand impersonation completely.

4.2.3 Stolen Verifier Attack

Because S does not store any password verification table, the proposed scheme can withstand stolen-verifier attacks. In our scheme, S generates a random value e for each user. Consequently, when authenticating with S, U_i only needs to send e to S and S uses x to re-construct $h(x \| e)$ of that user. So, S does not need to keep U_i's password in the storage space when a new user is participated into our system.

4.2.4 Stolen Informaton from Smart-card Attack

In our scheme, SC contains $\{A_i, L_i, e, N, h(.)\}$. If anybody picks or steals this information, he or she cannot derive further information because L_i is a hash value. Moreover, if ID_i and PW_i of victim who losts SC are not leaked, attacker cannot compute $h(x \| e)$ of victim. Clearly, our scheme can counteract this kind of attack.

4.2.5 Known-key Attack

The known-key security means that compromise of a past session-key cannot derive any further session-key. In our scheme, SK is associated with two random values R_U, R_S and $h(x \| e)$, which are unknown to the adversary. Even though the past SK is disclosed, the attacker cannot derive R_U, R_S and $h(x \| e)$ based on the security of one-way hash function and random values. Thus, the attacker can not obtain any further session-key.

4.2.6 Mutual Authentication

In our scheme, both user and server generate random values to challenge each other. S only accepts U_i when C_i is equal to $h(ID_i \| R_U^* \| h(x \| e))$ and M is equal to $h(R_U^* \| R_S)$. U_i only accepts S when V is equal to $h(R_S^* \| h(x \| e))$. In other words, user and server must computes random values to prove their validity. Clearly, our scheme provides mutual authentication.

4.2.7 Session-key Agreement

In our scheme, after finishing mutual authentication successfully, both user and server share common SK to encrypt messages later. So, our scheme not only satisfies mutual authentication but also provides session-key to partners.

Our scheme is a revised version of Yung-Cheng Lee's scheme, so it can also resist password guessing attack and provide user anonymity.

4.3 Efficiency Analysis

To compare efficiency between our scheme and the Yung-Cheng Lee's, we reuse approach used in his scheme 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. In addition, similarly to Lee's scheme, we also ignore exclusive-or($\oplus$) and concatenation operations($\|$) because they requires very few computations.

In table 1, there are our scheme and Yung-Cheng Lee's. Lee's scheme needs $1 \times T_h$ in registration phase, and $7 \times T_h$ in login and verification phases. Our scheme needs $4 \times T_h$ in registration phase and $12 \times T_h$ in login and mutual authentication phases.

Table 1: Comparison of computation cost

Schemes \ Phases	Registration	Login & Auth
Lee's	$1 \times T_h$	$7 \times T_h$
Ours	$4 \times T_h$	$12 \times T_h$

Clearly, proposed scheme needs more computational amount than Yung-Cheng Lee's scheme. However, those costs are necessary to protect user's anonymity and provide session-key for partners. In short, Additional computational cost is essential to enhance security.

Due to the fact that our scheme and Lee's are based on smart-card, we compare the storage capacity of smart-card. To do that, we assume that output hash function is 160 bit long, for example SHA-I. Furthermore, we also would like to consider communication cost between user and server in term of authentication in two schemes. In table 2, we see that smart-card of our schemes contains 160 bits longer than one in Lee's scheme. Besides, in authentication phase, our scheme also needs more twice bits than Lee's scheme. However, these costs increase security for scheme.

Table 2: Comparison of communication cost & storage capacity

Capacity & Communication costs \ Schemes	Lee	Our
Bits in smart-card	320	480
Bits in authentication	480	1120

In table 3, we list the comparisons between our improved scheme and Yung-Cheng Lee's scheme for withstanding various attacks. We see that his scheme cannot resist to impersonation and smart-card-theft attacks. In addition, their scheme does not provide mutual authentication and session-key agreement. It can be seen that our proposed scheme is more secure against various attacks.

Table 3: A comparison between our scheme and the Yung-Cheng Lee's for withstanding various attacks

Kinds of Attacks \ Schemes	Lee's	Ours
Impersonation	No	Yes
Smart-card-Theft	No	Yes
Password guessing	Yes	Yes
Stolen verification table	Yes	Yes
Known-key	No existing	Yes
Mutual authentication	No existing	Yes
Session-key exchange	No existing	Yes
Replay	Yes	Yes

5 Conclusions

In this paper, we review a new dynamic ID-based user authentication scheme to resist smart-card-theft attack of Yung-Cheng Lee. Although his scheme can withstand some attacks, such as password guessing. Nevertheless, we see that his scheme is still vulnerable to impersonation and smart-card-theft attacks. Morever, his scheme cannot provide mutual authentication and session-key agreement. Consequently, we propose an improved scheme to eliminate such problems.

Acknowledgement

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] Lamport, L.: Password Authentication with Insecure Communication. Communications of the ACM., **24**. 770-772 (1981).
[2] Li, L. H., Lin, I. C., Hwang, M. S.: A Remote Password Authentication Scheme for Multiserver Architecture Using Neural Networks. IEEE Transactions on Neural Network., **12**, 1498-1504 (2001).
[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] Hwang, M. S., Lee, C. C., Tang, Y. L.: A Simple Remote User Authentication Scheme. Mathematical & Computer Modelling, **36**, 103-107 (2002).
[5] Lee, C.C., Hwang, M.S., Yang, W.P.: Flexible Remote User Authentication Scheme Using Smart Cards. ACM Operating Systems Review, **36**, 46-52 (2002).
[6] I-En Liao., Cheng-Chi Lee., Min-Shiang Hwang: Security enhancement for a dynamic ID-based remote user authentication scheme. IEEE Transactions on Consumer Electronics, **50**, 629-631 (2004).
[7] Eun-Jun Yoon, Kee-Young Yoo: Improving the Dynamic ID-Based Remote Mutual Authentication Scheme. OTM Workshops, **1**, 499-507 (2006).
[8] Cheng-Chi Lee, Tsung-Hung Lin, Rui-Xiang Chang: A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. Expert Systems with Applications, **38**, 13863-13870 (2011).
[9] Chin-Ling Chen, Cheng-Chi Lee, and Chao-Yung Hsu: Mobile Device Integration of a Fingerprint Biometric Remote Authentication Scheme, International Journal of Communication Systems (ISSN 1074-5351) (Accepted: Mar. 13, 2011).
[10] Manik Lal Das, Ashutosh Saxena, Ved P. Gulati: A Dynamic ID-based Remote User Authentication Scheme. IEEE Transactions on Consumer Electronics, **50**, 629-631 (2004).
[11] SK Hafizul Islam, G.P. Biswas: A more efficient & secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Journal of Systems & Software, In Press, Corrected Proof, Available online 7 July 2011.
[12] Hsiang, Han-Cheng and Shih, Wei-Kuan: Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standard Interfaces. Elsevier Science Publishers B. V, **31**, 1118-1123 (2009).
[13] Yung-Cheng Lee: A New Dynamic ID-based User Authentication Scheme to Resist Smart-Card-Theft Attack. Applied Mathematics and Information Sciences, **6**, 355-361 (2012).
[14] Liao, Yi-Pin and Wang, Shuenn-Shyang: A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standard Interfaces. Elsevier Science Publishers B. V, **31**, 24-29 (2009).
[15] Zhang, Juan and Deng, Fangmin: The Authentication and Key Agreement Protocol Based on ECC for Wireless Communications. IEEE International Conference on Management and Service Science, 1-4 (2009).
[16] I En Liao and Cheng Chi Lee and Min Shiang Hwang: Security enhancement for a dynamic ID-based remote user authentication scheme. International Conference on Next Generation Web Services Practices. IEEE Computer Society Washington, DC, USA, **6**, 517-522 (2009).
[17] Debiao, He and Jianhua, Chen and Jin, Hu: An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security. Information Fusion. Elsevier B.V.
[18] Jen-Ho Yang and Chin-Chen Chang: An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Computers and Security, **28**, 138-143 (2009).
[19] Eun-Jun Yoon and Kee-Young Yoo: Robust ID-Based Remote Mutual Authentication with Key Agreement Scheme for Mobile Devices on ECC. Computational Science and

Engineering, IEEE International Conference on. IEEE Computer Society, **2**, 633-640 (2009).

[20] Y Y. Wang and J Y. Kiu and F X. Xiao and J. Dan: A more efficient and secure dynamic ID-based remote user authentication scheme. Computer Communications, **32**, 583-585 (2009).

[21] M Burrows and M Abadi and R Needham: A logic of authentication. ACM Transactions on Computer System, **8**, 18-36 (1990).

Toan Thinh Truong obtained his B.Sc., M.Sc. degrees in computer science from University of Science, Ho Chi Minh City, Vietnam in 2008 and 2011. He joined the University of Science, VNU-HCM, in 2012. His research interests include mutual authentication, software engineering and visual cryptography. He is currently lecturer of Software Engineering Department, Faculty of Information Technology, University of Science, VNU-HCM.

Minh Triet Tran obtained his B.Sc., M.Sc., and Ph.D. degrees in computer science from University of Science, Ho Chi Minh City, Vietnam in 2001, 2005, and 2009. He joined the University of Science, VNU-HCM, in 2001. His research interests include cryptography and security, human-computer interaction, and software engineering. He is currently Deputy Head of Software Engineering Department, Faculty of Information Technology, University of Science, VNU-HCM. He is a member of the IEEE.

Anh Duc Duong is a professor at the University of Information Technology, Ho Chi Minh City, Vietnam. He obtained his B.Sc. and M.Sc. in computer science from the University of Ho Chi Minh City in 1990 and 1995, respectively. He received his PhD degree in mathematics from the University of Science, VNU-HCM, Vietnam in 2002. His research interests include cryptography and security, geographic information systems, computer graphics, and image processing. He is currently the President of University of Information Technology and Chair of Program of Information Technology and GIS of Ho Chi Minh City, Vietnam. He is a member of the IEICE, ACM, and IEEE.