CRYPTANALYSIS AND ENHANCEMENT OF PASSWORD AUTHENTICATION SCHEME FOR SMART CARD

Raphael Nyirongo[1], Solomon Kuonga[1], Patrick Ali[1], Levis Eneya[1] and Hyunsung Kim[1,2]

[1]Mathematical Sciences Department, University of Malawi, Chancellor College, Zomba, Malawi

[2] (Corresponding Author) Department of Cyber Security, Kyungil University, Kyungbuk, Korea

ABSTRACT

Password authentication with smart card is one of the simplest and efficient authentication mechanisms to ensure secure communication over insecure network environments. Recently, Tsai et al. proposed an improved password authentication scheme for smart card. Their scheme is more secure than the other previous schemes. In this paper, we show Tsai et al.'s scheme is vulnerable to password guessing attack and has computational overhead. Furthermore, we propose an enhanced password authentication scheme to eliminate the security vulnerability and enhance the overhead. By presenting concrete analysis of security and performance, we show that the proposed scheme cannot only resist various well known attacks, but also is more efficient than the other related works, and thus is feasible for practical applications.

KEYWORDS

Information Security, User Authentication, Password Authentication, Smart Card, Timestamp

1. INTRODUCTION

More resources are getting distributed over the network due to the rapid progress in information technology, which is managed by servers in distributed systems [1]. Many systems that control remote access to computer networks use password based authentication and many people researched about how to make secure authentication [2-4]. However, the password is easily exposed by guessing attacks [3]. However, there still exists challenges in both security and performance aspects due to the stringent security requirements and resource strained characteristics of the clients.

Since Chang et al. in [5] introduced the first remote user authentication scheme using smart cards, there has been many of such schemes proposed [6-9]. One prominent issue in this type of schemes is security against offline guessing attack. Traditionally, to prevent an adversary from launching offline guessing attack, one needs to make sure that the scheme is not going to leak any information useful about the client's password to the adversary in the protocol run, even though the password is considered to be weak and low-entropy. By observing this, many schemes assumed that the smart card is tamper-resistant, i.e., the secret parameters stored in the smart card cannot be revealed. However, recent results have demonstrated that the secret data stored in the smart card could be extracted by some means, such as monitoring the power consumption [10] or analyzing the leaked information [11]. Therefore, such schemes [6-8] based on the tamper resistance assumption of the smart card are at least vulnerable to offline password guessing attacks, once an adversary has obtained the secret data stored in a user's smart card [12-14]. Consequently, a stronger notion of security against offline guessing attack is developed to require

DOI:10.5121/ijcis.2017.7301

that compromising a client's smart card should not help the adversary launch offline guessing attack against the client's password.

Recently, Chen et al. proposed a smart card based user authentication scheme [15]. However, Li et al. pointed out some weaknesses in Chen et al.'s scheme and they also proposed an enhanced smart card based on a user authentication scheme to resist the above flaws existing in Chen et al.'s scheme [16]. However, Wei et al. showed that Li et al.'s scheme is powerless against the off-line password guessing attack and they also proposed an efficient and secure smart card based remote user password authentication scheme [17]. Wei et al.'s scheme is more efficient and secure than other schemes. However, Tsai et al. unfortunately presented security weaknesses on password guessing attack, privileged insider attack and denial of service attack against Wei et al.'s scheme [18]. Furthermore, Tsai et al. proposed an improvement authentication scheme in [18] and argued that their scheme is secure against various attacks.

This paper provides security and performance analyses on Tsai et al.'s scheme focused on password guessing attack vulnerability and computational overhead concern after reviewing the scheme briefly. Addition to that, we propose an enhanced password authentication scheme (EPAS) as a remedy scheme, which is based on hash function and using biometric authentication. For the computational efficiency, the enhanced scheme tries to remove the expensive exponentiation operation, which is extremely slower than the symmetric key cryptosystem operation or hash function. We provide the security of the enhanced scheme based on the BAN logic and hash based oracle.

2. REVIEW OF TSAI ET AL.'S AUTHENTICATION SCHEME

In this section, we briefly review Tsai et al.'s improved password authentication scheme with smart card [18]. Tsai et al.'s scheme is composed of three phases, registration, login and authentication.

2.1. REGISTRATION PHASE

In this phase, the server *SV* makes a smart card SC for a new user, U_i. The smart card *SC* contains six parameters, $\{A_i, p, q, h(.), r, W_i\}$, where $A_i = h(x||ID_i)+h(ID_i||h(PW_i||r))$; $W_i = h(PW_i||r)$; $h(.)$ denotes a secure hash function ($h(.)$: $\{0, 1\}^* \rightarrow Z_p^*$); p and q are two large prime numbers such that $p = 2q + 1$; x denotes a master secret key ($x \in Z_q^*$); r is a random number; ID_i and PW_i are user's identity and password, respectively. p, q, x, and $h(.)$ are selected by *SV*. ID_i, PW_i and r are selected by U_i. U_i sends $\{ID_i, h(PW_i||r)\}$ to *SV*. *SV* does not know the random number r.

2.2. LOGIN PHASE

In this phase, U_i wants to login into *SV* for obtaining some services; U_i first attaches his (or her) smart card to a device reader and inputs his (or her) identity ID_i and password PW_i. The login phase is executed as follows:

LP1) U_i sends the login request parameters, his (or her) identity ID_i and password PW_i to the smart card *SC*. *SC* computes $W_i' = h(PW_i||r)$ and checks whether W_i' is equal to W_i. If it holds, *SC* executes the next steps. If U_i fails to verify ID_i and PW_i for 3 times, U_i will lock *SC*.

LP2) *SC* computes B_i, D_i, F_i, M_i as follows: $B_i = A_i - h(ID_i||h(PW_i||r)) = h(x||ID_i)$; $D_i = h(ID_i)^a \bmod p$; $F_i = D_i+B_i$; $M_i = h(ID_i||F_i||T_1) \oplus B_i$, where T_1 denotes the current timestamp of *SC* and a denotes a random number.

LP3) *SC* sends $\{ID_i, F_i, M_i, T_1\}$ to *SV*.

The adversary only has three times to guess the user's password in Step 2 of the login phase.

2.3. AUTHENTICATION PHASE

Upon receiving the authentication request message $\{ID_i, F_i, M_i, T_1\}$ from U_i, SV executes this authentication phase as follows:

AP1) SV checks whether ID_i format and the timestamp T_1 are in valid time or not. If both of conditions hold, SV continuously authenticates the following steps.

AP2) SV checks whether $M_i' = h(ID_i||F_i||T_1) \oplus h(x||ID_i)$ is equal to M_i or not. If it does not hold, SV rejects the login request. Otherwise, SV computes $V_i = h(ID_i)^b \bmod p$ and $M_S = h(ID_i||D_i'||V_i||Z_i||T_S)$; where b is a random number, $D_i' = F_i - h(x||ID_i) = h(ID_i)^a \bmod p$, $Z_i = (D_i')^b \bmod p$, and T_S is the current time of SV. Finally, SV sends the message $\{ID_i, V_i, M_S, T_S\}$ to U_i.

AP3) After receiving the message, SC checks ID_i and compares T_S with T_S', where T_S' the time that the message is received. If ID_i is valid and $T_S' - T_S \leq T$, SC computes $Z_i' = V_i^a \bmod p$, $M_S' = h(ID_i||D_i||V_i||Z_i'||T_S)$. If $M_S' \neq M_S$, the session is terminated. Otherwise, SV is authenticated by U_i, and the shared session key is set as $sk = h(ID_i||D_i||V_i||Z_i')$. Furthermore, U_i gets the current time T_i^{new}, and generates a response message $R_i = h(ID_i||D_i||V_i||Z_i'||T_i^{new})$, and then sends the message $\{ID_i, R_i, T_i^{new}\}$ to SV.

AP4) Upon receiving the response message, SV checks ID_i and T_i^{new}. If they are valid, SV computes $R_i' = h(ID_i||D_i'||V_i||Z_i||T_i^{new})$. If $R_i' \neq R_i$, SV terminates the session. Otherwise, U_i is authenticated by SV, and the shared session key is set as $sk' = h(ID_i||D_i'||V_i||Z_i)$. Finally, an agreed session key $sk = sk'$ is established between U_i and SV.

3. ANALYSES OF TSAI ET AL.'S AUTHENTICATION SCHEME

In this section, we provide security analysis and computational overhead analysis. First of all, we will show that Tsai et al.'s scheme in [18] is weak against password guessing attack based on two adversary assumptions. Furthermore, it has big computational overhead due to exponentiation operations in the authentication phase.

3.1. PASSWORD GUESSING ATTACK FEASIBILITY

For the security analysis, we will follow Xu et al.'s two assumptions of the adversary's capabilities explicitly made in this kind of authentication scheme [19] :

A1) Adversary has total control over the communication channel between the users and the remote server in the protocol run, which means the adversary can intercept, insert, delete, or modify any message transmitted in the channel.

A2) Adversary may either steal a user's smart card and then extract the information from it by the method introduced by Kocher et al. [20], or obtain a user's password, but not both.

They have been widely accepted as the standard threat model for cryptographic protocols [21].

By A2, an adversary $\mathcal{A}$ can obtain U_i's smart card and extract the data $\{A_i, p, q, h(.), r, W_i\}$. Subsequently, $\mathcal{A}$ can launch off-line password guessing attacks as follows:
(1) $\mathcal{A}$ picks up a password candidate PW_i'.

(2) $\mathcal{A}$ computes $W_i^{'} = h(PW_i^{'}||r)$. Note that if $PW_i^{'} = PW_i$, then it holds that $W_i^{'} = W_i$, which means that $\mathcal{A}$ can verify the validity of $PW_i^{'}$. Otherwise, $\mathcal{A}$ repeats the above procedure until the correct password is found.

In Tsai et al.'s scheme, the password is selected by the user, which indicates that it is value easy to remember and guess, rather than random numbers with high entropy. Thereby, Tsai et al.'s scheme is still weak against password guessing attack.

3.2. COMPUTATIONAL OVERHEAD CONCERN

For the computational overhead analysis, we need to check the following steps of LP2, AP2 and AP3 from Tsai et al.'s scheme.

LP2) SC computes B_i, D_i, F_i, M_i as follows: $B_i = A_i - h(ID_i||h(PW_i||r)) = h(x||ID_i)$; $D_i = h(ID_i)^a \bmod p$; $F_i = D_i + B_i$; $M_i = h(ID_i||F_i||T_1) \oplus B_i$, where T_1 denotes the current timestamp of SC and a denotes a random number.

AP2) SV checks whether $M_i' = h(ID_i||F_i||T_1) \oplus h(x||ID_i)$ is equal to M_i or not. If it does not hold, SV rejects the login request. Otherwise, SV computes $V_i = h(ID_i)^b \bmod p$ and $M_S = h(ID_i||D_i'||V_i||Z_i||T_S)$; where b is a random number, $D_i' = F_i - h(x||ID_i) = h(ID_i)^a \bmod p$, $Z_i = (D_i')^b \bmod p$, and T_S is the current time of SV. Finally, SV sends the message $\{ID_i, V_i, M_S, T_S\}$ to U_i.

AP3) After receiving the message, SC checks ID_i and compares T_S with T_S', where T_S' the time that the message is received. If ID_i is valid and $T_S' - T_S \leq T$, SC computes $Z_i' = V_i^a \bmod p$, $M_S' = h(ID_i||D_i||V_i||Z_i'||T_S)$. If $M_S' \neq M_S$, the session is terminated. Otherwise, SV is authenticated by U_i, and the shared session key is set as $sk = h(ID_i||D_i||V_i||Z_i')$. Furthermore, U_i gets the current time T_i^{new}, and generates a response message $R_i = h(ID_i||D_i||V_i||Z_i'||T_i^{new})$, and then sends the message $\{ID_i, R_i, T_i^{new}\}$ to SV.

The scheme requires modular exponentiation operations to compute D_i, V_i and Z_i, which requires a big overhead than the other operations.

4. ENHANCED PASSWORD AUTHENTICATION SCHEME

In this section, we propose a new enhanced password authentication scheme (EPAS) with smart card, which could solve all the security and overhead problems depicted in the previous section. Especially, EPAS uses biometrics to cope from the attack and removes the expensive operations to be computationally effective. EPAS has three phases, registration, login and authentication. Figure 1 shows the flows of EPAS.

Initially, the server SV initializes system parameters. SV chooses its master secret key $x \in Z_p^*$ and two secure hash functions, $h(.)$: $\{0, 1\}^* \rightarrow Z_p^*$ and $H(.)$: $\{0, 1\}^* \rightarrow Z_p^*$.

4.1. REGISTRATION PHASE

When a user U_i wants to be a member of SV, this phase is performed as follows:

(1) U_i selects his (or her) identity ID_i and password PW_i after generating a random number r. U_i computes $h(PW_i||r)$ and submits $\{ID_i, h(PW_i||r)\}$ to SV as the registration request message via a secure channel.

(2) After receiving the message, SV checks whether ID_i is valid or not. If it is not, SV rejects the request. Otherwise, SV computes $A_i = h(x||ID_i) \oplus h(ID_i||h(PW_i||r))$ and issues a smart card SC to U_i via a secure channel, which stores $\{A_i, p, h(.), H(.)\}$.

(3) After receiving the SC, U_i inputs PW_i and r, imprints his (or her) fingerprint b, computes $W_i = h(PW_i||r||H(b))$ and stores r and W_i into SC.

4.2. LOGIN PHASE

In this phase, U_i logins into SV for some services; U_i first attaches his (or her) smart card to the smart card reader and inputs his (or her) identity ID_i, password PW_i and fingerprint b. The login phase is executed as follows:

(1) U_i inputs his (or her) identity ID_i, password PW_i and fingerprint b to SC. SC computes $W_i' = h(PW_i||r||H(b))$ and checks whether W_i' is equal to W_i. Only if it holds, SC executes the next steps. If U_i fails to verify ID_i, PW_i and b for 3 times, SC will be locked.

(2) SC computes B_i, D_i, F_i and M_i as follows: $B_i = A_i \oplus h(ID_i||h(PW_i||r)) = h(x||ID_i)$; $D_i = h(ID_i) \oplus a$; $F_i = D_i \oplus B_i$; $M_i = h(ID_i||F_i||B_i||T_1)$, where a denotes a random number and T_1 denotes the current timestamp of SC.

(3) SC sends $\{ID_i, F_i, M_i, T_1\}$ to SV.

The adversary only has three times chance to guess the user's password in Step 1 of the login phase.

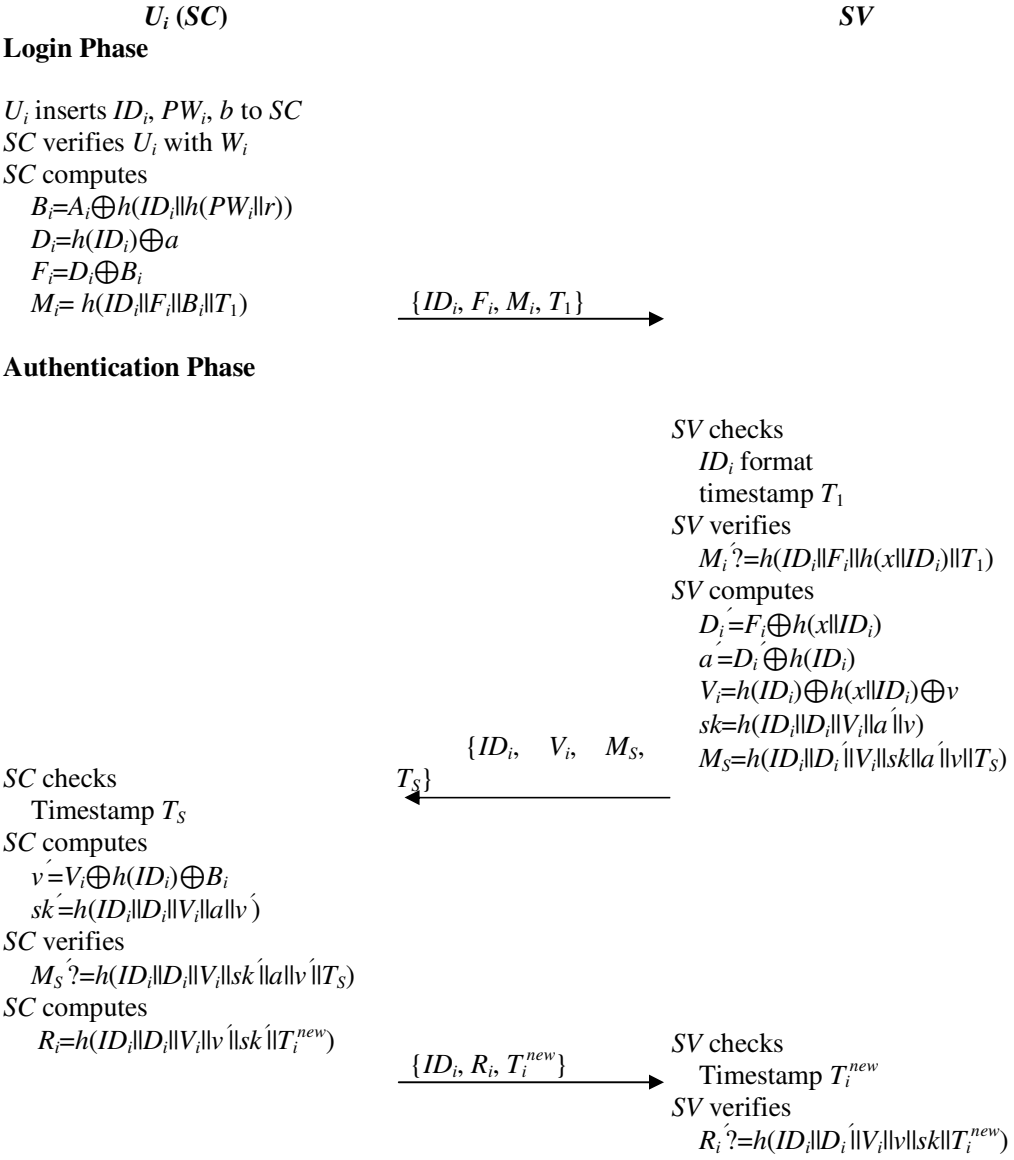

Figure 1. Enhanced password authentication scheme

4.3. AUTHENTICATION PHASE

Upon receiving the authentication request message $\{ID_i, F_i, M_i, T_1\}$ from U_i, SV executes this authentication phase as follows:

(1) SV checks whether ID_i format and the timestamp T_1 are valid or not. If both of conditions hold, SV continuously performs the following steps.

(2) SV checks whether $M_i' = h(ID_i||F_i||h(x||ID_i)||T_1)$ is equal to M_i or not. If it does not hold, SV rejects the login request. Otherwise, SV computes $V_i = h(ID_i) \oplus h(x||ID_i) \oplus v$, $sk = h(ID_i||D_i||V_i||a'||v)$ and $M_S = h(ID_i||D_i'||V_i||sk||a'||v||T_S)$; where v is a random number, $D_i' = F_i \oplus h(x||ID_i) = h(ID_i) \oplus a$, $a' = D_i' \oplus h(ID_i)$ and T_S is the current time stamp of SV. Finally, SV sends the message $\{ID_i, V_i, M_S, T_S\}$ to U_i.

(3) After receiving the message, SC checks ID_i and compares T_S with T_S', where T_S' is the time stamp of SC when the message is received. If ID_i is valid and $T_S' - T_S \leq \Delta T$, SC computes $v' = V_i \oplus h(ID_i) \oplus B_i$, $sk' = h(ID_i||D_i||V_i||a||v')$ and $M_S' = h(ID_i||D_i||V_i||sk'||a||v'||T_S)$. If $M_S' \neq M_S$, SC terminates the session. Otherwise, SV is authenticated by U_i, and the shared session key is set as sk'. Furthermore, U_i gets the current time T_i^{new}, generates a response message $R_i = h(ID_i||D_i||V_i||v'||sk'||T_i^{new})$, and sends the message $\{ID_i, R_i, T_i^{new}\}$ to SV.

(4) Upon receiving the response message, SV checks ID_i and T_i^{new}. If they are valid, SV computes $R_i' = h(ID_i||D_i'||V_i||v||sk'||T_i^{new})$. If $R_i' \neq R_i$, SV terminates the session. Otherwise, U_i is authenticated by SV, and SV believes that an agreed session key $sk = sk'$ is established between U_i and SV.

4.4. PASSWORD CHANGE PHASE

In this phase, U_i changes his (or her) password PW_i into PW_{new} after the success of user authentication from SC. U_i first attaches his (or her) smart card to the smart card reader and inputs his (or her) identity ID_i, password PW_i and fingerprint b. The password change phase is executed as follows:

(1) U_i sends the password change parameters, his (or her) identity ID_i, password PW_i and fingerprint b to SC. SC computes $W_i' = h(PW_i||r||H(b))$ and checks whether W_i' is equal to W_i. If it holds, SC asks an input of a new password PW_{new} to U_i. Otherwise, SC rejects the request.

(2) SC computes $B_i = A_i \oplus h(ID_i||h(PW_i||r))$, $W_{new} = h(PW_{new}||r||H(b))$ and $A_{new} = B_i \oplus h(ID_i||h(PW_{new}||r))$ and updates W_i and A_i with W_{new} and A_{new}, respectively.

5. SECURITY ANALYSES

In this section, we provide security analysis based on BAN logic and formal security analysis. The security analysis of EPAS was conducted under the following assumptions:

1. An adversary $\mathcal{A}$ can be either a user or a server. U_i and as well as SC can act as an adversary.
2. $\mathcal{A}$ can eavesdrop on every communication across public channels. He (or she) can capture any message that is exchanged between U_i and SC.
3. $\mathcal{A}$ has the ability to alter, delete or reroute the captured message.
4. Information can be extracted from the smart card by examining the power consumption of the card.

5.1. PROOF USING BAN LOGIC

Formal security analysis of EPAS is verified with the help of Burrows, Abadi and Needham (BAN) logic [22]. The formal analysis of a network security protocol using BAN logic involves

following steps: (1) Converting original scheme statements to their idealized form. (2) Determining the assumptions about the initial state of the system. (3) Representation of the state of the system after executing each statement as logical assertions by attaching logical formulas to each statement. (4) Application of logical postulates to assumptions and assertions.

The following notations are used in formal security analysis using the BAN logic:

- $Q \mid\equiv X$: Principal Q believes the statement X.
- $\#(X)$: Formula X is fresh.
- $Q \mid\Longrightarrow X$: Principal Q has jurisdiction over the statement X.
- $Q \lhd X$: Principal Q sees the statement X.
- $Q \mid\sim X$: Principal Q once said the statement X.
- (X, Y): Formula X or Y is one part of the formula (X, Y).
- $\langle P \rangle_Q$: Formula P combined with the formula Q.
- $Q \overset{sk}{\leftrightarrow} R$: Principal Q and R may use the shared session key, sk to communicate among each other. The session key sk is good, in that it will never be discovered by any principal except Q and R.

In addition, the following four BAN logic rules are used to prove that EPAS provides a secure mutual authentication between U_i and SV:

Rule 1. **Message-meaning rule**: $\dfrac{R \mid\equiv R \overset{Y}{\leftrightarrow} S,\ \ R \lhd \langle X \rangle_Y}{R \mid\equiv S \mid\sim X}$

Rule 2. **Nonce-verification rule**: $\dfrac{R \mid\equiv \#(X),\ \ R \mid\equiv S \mid\sim X}{R \mid\equiv S \mid\equiv X}$

Rule 3. **Jurisdiction rule**: $\dfrac{R \mid\equiv S \mid\Rightarrow X,\ \ R \mid\equiv S \mid\equiv X}{R \mid\equiv X}$

Rule 4. **Freshness-concatenation rule**: $\dfrac{R \mid\equiv \#(X)}{R \mid\equiv \#(X,Y)}$

In order to show that EPAS provides secure mutual authentication between U_i and SV, we need to achieve the following four goals:

Goal 1: $U_i \mid\equiv (U_i \overset{sk}{\leftrightarrow} SV)$

Goal 2: $SV \mid\equiv (SV \overset{sk}{\leftrightarrow} U_i)$

Goal 3: $U_i \mid\equiv SV \mid\equiv (U_i \overset{sk}{\leftrightarrow} SV)$

Goal 4: $SV \mid\equiv U_i \mid\equiv (SV \overset{sk}{\leftrightarrow} U_i)$

Idealized form: The arrangement of the transmitted messages between U_i and SV in EPAS to the idealized forms is as follows:

Message 1. $U_i \rightarrow SV : ID_i, F_i, \langle M_1 \rangle_{h(x||ID_i)}, T_1$

Message 2. $SV \rightarrow U_i : ID_i, \langle V_i \rangle_{h(x||ID_i)}, \langle M_S \rangle_{sk}, T_S$

Message 3. $U_i \rightarrow SV : ID_i, \langle R_i \rangle_{sk}, T_s^{new}$

Assumptions: The following are the initial assumptions of EPAS:

A1: $U_i \mid\equiv \#(T_1, T_s^{new})$
A2: $SV \mid\equiv \#(T_S)$
A3: $U_i \mid\equiv (U_i \overset{h(x||ID_i)}{\longleftrightarrow} SV)$
A4: $SV \mid\equiv (SV \overset{h(x||ID_i)}{\longleftrightarrow} U_i)$
A5: $U_i \mid\equiv SV \mid\Rightarrow U_i \overset{sk}{\leftrightarrow} SV$
A6: $SV \mid\equiv U_i \mid\Rightarrow SV \overset{sk}{\leftrightarrow} U_i$

PROOF:

In the following, we prove the test goals in order to show the secure authentication using the BAN logic rules and the assumptions.

Based on Message 1, we could derive:
Step 1. $SV \lhd ID_i, F_i, <M_i>_{h(x||ID_i)}, T_1$
According to assumption A4 and the message meaning rule, we could get:

Step 2. $SV \models U_i \mid\sim (ID_i, F_i, <M_i>_{h(x||ID_i)}, T_1)$
According to assumption A1 and the freshness concatenation rule, we could get:

Step 3: $SV \models \#(ID_i, F_i, <M_i>_{h(x||ID_i)}, T_1)$
According to Step 2, Step 3 and the nonce verification rule, we could get:

Step 4. $SV \models U_i \models (ID_i, F_i, <M_i>_{h(x||ID_i)}, T_1)$
According to Step 4, assumption A3 and the believe rule, we could get:

Step 5. $SV \models U_i \models (U_i \overset{h(x||ID_i)}{\longleftrightarrow} SV)$
According to the jurisdiction rule, we could get:

Step 6. $SV \models (SV \overset{h(x||ID_i)}{\longleftrightarrow} U_i)$
Based on Message 2, we could derive

Step 7. $U_i \lhd ID_i, <V_i>_{h(x||ID_i)}, <M_S>_{sk}, T_S$
According to assumption A3 and the message meaning rule, we could get:

Step 8. $U_i \models SV \mid\sim (ID_i, <V_i>_{h(x||ID_i)}, <M_S>_{sk}, T_S)$
According to assumption A2 and the freshness concatenation rule, we could get:

Step 9: $U_i \models \#(ID_i, <V_i>_{h(x||ID_i)}, <M_S>_{sk}, T_S)$
According to Step 8, Step 9 and the nonce verification rule, we could get:

Step 10. $U_i \models SV \models (ID_i, <V_i>_{h(x||ID_i)}, <M_S>_{sk}, T_S)$
According to Step 10, assumption A4 and the believe rule, we could get:

Step 11. $U_i \models SV \models (SV \overset{h(x||ID_i)}{\longleftrightarrow} U_i)$
According to the jurisdiction rule, we could get:

Step 12. $U_i \models (U_i \overset{h(x||ID_i)}{\longleftrightarrow} SV)$
According to Step 8, Step 9, Step 10 and the nonce verification rule, we could get:

Step 13. $U_i \models SV \models (SV \overset{sk}{\leftrightarrow} U_i)$ **(Goal 3)**
According to assumption A5 and the jurisdiction rule, we could get:

Step 14. $U_i \models (U_i \overset{sk}{\leftrightarrow} SV)$ **(Goal 1)**
Based on Message 3, we could derive

Step 15. $SV \lhd ID_i, <R_i>_{sk}, T_s^{new}$
According to assumption A4 and the message meaning rule, we could get:

Step 16. $SV \models U_i \mid\sim (ID_i, <R_i>_{sk}, T_s^{new})$
According to assumption A1 and the freshness concatenation rule, we could get:

Step 17: $SV \models \#(ID_i, <R_i>_{sk}, T_s^{new})$
According to Step 16, Step 17 and the nonce verification rule, we could get:

Step 18. $SV \models U_i \models (ID_i, <R_i>_{sk}, T_s^{new})$
According to Step 18, assumption A3 and the believe rule, we could get:

Step 19. $SV \models U_i \models (U_i \overset{sk}{\leftrightarrow} SV)$ **(Goal 4)**
According to assumption A6 and the jurisdiction rule, we could get:

Step 20. $SV \models (SV \overset{sk}{\leftrightarrow} U_i)$ **(Goal 2)**

According to Steps 14 and 20, EPAS successfully achieves both goals (Goals 1 and 2). Both U_i and SV believes that they share a common session key $sk = h(ID_i||D_i||V_i||a||v)$.

5.2. FORMAL SECURITY ANALYSIS

This subsection demonstrates the formal security analysis of EPAS and shows that it is secure. First of all, we define the hash function [23].

Definition 1. A secure one way hash function $h(.)$: X=0, 1}* -> Y={0, 1}^{n}, which takes an input as an arbitrary length binary string $x \in \{0, 1\}^*$ and outputs a binary string $h(x) \in \{0, 1\}^n$, which satisfies the following requirements:

a. Given $y \in Y$, it is computationally infeasible to find an $x \in X$ such that $y=h(x)$.
b. Given $x \in X$, it is computationally infeasible to find another $x' \neq x \in X$ such that $h(x')=h(x)$.
c. It is computationally infeasible to find a pair $(x', x) \in X' \times X$, with $x' \neq x$, such that $h(x')=h(x)$.

Theorem 1. Under the assumption that the one way hash function $h(.)$ closely behaves like an oracle, EPAS is provably secure against an adversary $\mathcal{A}$ for the protection of U_i's identity ID_i, password PW_i and fingerprint $b$and SV's secret value x that is selected by SV.

Proof. The formal security proof of EPAS is based on those in [24-26]. Using the oracle to construct $\mathcal{A}$ who has the ability to derive U_i's identity ID_i, password PW_i and fingerprint b and SV's secret value x.

Reveal : $\mathcal{A}$ will unconditionally output the input x from the given hash result $y= h(x)$.

Now, $\mathcal{A}$ runs an experimental algorithm $EXP_{HASH,A}^{EPAS}$ for EPAS. If the success probability of $EXP_{HASH,A}^{EPAS}$ is defined as $Success_{HASH,A}^{EPAS} = |Pr[EXP_{HASH,A}^{EPAS} = 1] - 1|$, the advantage function for this experiment becomes $ADV_{HASH,A}^{EPAS}(t, q_R)=max_A\, Success_{HASH,A}^{EPAS}$, where the maximum is taken over all of $\mathcal{A}$ with the execution time t and the number of queries q_R that are made to the Reveal oracle. If $\mathcal{A}$ has the ability to solve the hash function problem that is provided in Definition 1, he (or she) can directly derive U_i's identity ID_i, password PW_i and fingerprint b and SV's secret value x. In this case, $\mathcal{A}$ will discover the complete connections between U_i and SV. However, it is a computationally infeasible to invert the input from a given hash value, i.e., $ADV_{HASH,A}^{EPAS}(t) \leq \varepsilon$, $\forall \varepsilon > 0$. Then, we have $ADV_{HASH,A}^{EPAS}(t, q_R) \leq \varepsilon$, since $ADV_{HASH,A}^{EPAS}(t, q_R)$ depends on $ADV_{HASH,A}^{EPAS}(t)$. As

a result, there is no way for $\mathcal{A}$ to discover the complete connections between U_i and SV and by deriving $\{ID_i, PW_i, b, x\}$, EPAS is provably secure against the adversary.

Algorithm $EXP_{HASH,A}^{EPAS}$

1. Eavesdrop the login request message $\{ID_i, F_i, M_i, T_1\}$
2. Call the Reveal oracle. Let $\{ID_i', F_i', B_i', T_1'\} \leftarrow Reveal(M_i)$
3. Eavesdrop the login response message $\{ID_i, V_i, M_S, T_S\}$
4. Call the Reveal oracle. Let $\{ID_i'', D_i', V_i', a', v', T_S''\} \leftarrow Reveal(M_S)$
5. **If** $(ID_i' = ID_i'')$ **then**
6. Call the Reveal oracle. Let $\{PW_i', r', b'\} \leftarrow Reveal(W_i)$
7. Call the Reveal oracle. Let $\{x', ID_i'''\} \leftarrow Reveal(B_i)$
8. Accept ID_i', PW_i', b' and x' as the correct ID_i, PW_i, b and x
9. **return** 1
10. **else**
11. **return** 0
12. **end if**

6. PERFORMANCE ANALYSIS

In this section, we summarize the performance analysis of EPAS in terms of the computation complexities. We thus present a performance evaluation to compare EPAS to the other related schemes [17, 18]. We present a comparison of the computational costs, and measure the execution time. The computational analysis of an authentication scheme is generally conducted by focusing on operations performed by each party within the schemes. Therefore, for analysis of the computational costs, we concentrated on the operations that are conducted by the parties in the network: namely a user and a server. In order to facilitate the analysis of the computational costs, we define the following notation.

- T_h: the time to execute a one-way hashing operation
- T_e: the time to compute a modular exponentiation operation

In addition, in order to achieve accurate measurement, we performed an experiment. This experiment was performed using the Crypto++ Library [27] on a system using the 64-bits Windows 7 operating system, 3.2 GHz processor, 4 GB memory, Visual C++ 2013 Software, the SHA-1 hash function, the AES symmetric encryption/decryption function, and the ECC-160 function. According to our experiment, T_h is nearly 0.0002 seconds on average and T_e is nearly 0.6 seconds on average.

Table 1. Performance comparisons among the related schemes.

Overhead / Scheme	User side	Server side	Total
Wei et al. in [17]	$2T_e+6T_h$	$2T_e+7T_h$	$4T_e+13T_h$
Tsai et al. in [18]	$2T_e+7T_h$	$2T_e+6T_h$	$4T_e+13T_h$
EPAS	$6T_h$	$6T_h$	$12T_h$

Table 1 shows a comparative analysis of the computational cost among the related schemes. In addition, even though EPAS is computationally efficient than the other schemes, EPAS assures higher security, and affords resistance to the most well known attacks, while providing functionality.

7. CONCLUSION

This paper first examined Tsai et al.'s improved password authentication scheme for smart card. Our cryptanalysis showed that the scheme is vulnerable to password guessing attack once the private information stored in the smart card has been disclosed. In addition, we also pointed out that Tsai et al.'s scheme has computational overhead problem. Subsequently, to overcome the defects existing in the scheme, we proposed an enhanced password authentication scheme for smart card. By presenting the concrete analysis of security, we demonstrated that our proposal is not only free from various well known attacks, but also is more efficient than the other previous related works. Thus, our scheme is more feasible for practical applications.

ACKNOWLEDGEMENTS

Corresponding author is Hyunsung Kim. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

REFERENCES

[1] Prabhakar, M.,(2013)"Elliptic Curve Cryptography in Securing Networks by Mobile Authentication", International Journal on Cryptography and Information Security, Vol. 3, No. 3, pp 31-46.

[2] Valluri, M. R.,(2012)"Authentication Schemes using Polynomials over Non-commutative Rings", International Journal on Cryptography and Information Security, Vol. 2, No. 4, pp 51-58.

[3] Lee, C.-C. Liu,C.-H.&Hwang, M.-S.,(2013)"Guessing Attacks on Strong-Password Authentication Protocol", International Journal of Network Security, Vol. 15, No. 1, pp 64-67.

[4] Belgacem, N.,Nait-Ali, A., Fournier, R. & Bereksi-Reguig, F., (2012) "ECG based human Authentication using Wavelets and Random Forests", International Journal on Cryptography and Information Security, Vol. 2, No. 2, pp 1-11.

[5] Chang,C.C.& Wu,T.C.,(1991) "Remote password authentication with smart cards," IEE Proceedings-Computers and Digital Techniques, Vol. 138, No. 3, pp 165-168.

[6] Chen,T.,Hsiang, H. & Shih, W., (2011) "Security enhancement on an improvement on two remote user authentication schemes using smart cards," Future Generation Computer Systems, Vol. 27, No. 4, pp 377–380.

[7] Khan, M., Kim, S. & Alghathbar, K., (2011) "Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme," Computer Communications, Vol. 34, No. 3, pp 305–309.

[8] Wang, Y.,Liu,J., Xiao, F.& Dan,J.,(2009) "A more efficient and secure dynamic id-based remote user authentication scheme," Computer communications, Vol. 32, No. 4, pp 583–585.

[9] Kim, H.S.& Lee,S.W.,(2010) "Robust Remote User Authentication Scheme using Smart Cards", Journal of Security Engineering, Vol. 7, No. 5, pp 495-502.

[10] Bogdanov,A.& Kizhvatov, I., (2012) "Beyond the limits of dpa: Combined side-channel collision attacks," IEEE Transactions on Computers, Vol. 61, No. 8, pp 1153–1164.

[11] Kasper,T.,Oswald,D.& Paar,C.,(2012) "Side-channel analysis of cryptographic RFIDs with analog demodulation," Lecture Notes in Computer Science, Vol. 7055, pp 61–77.

[12] Chen, B., Kuo, W. & Wuu, L., (2012) "Robust smart-card-based remote user password authentication scheme," International Journal of Communication Systems, doi: http://dx.doi.org/10.1002/dac.2368.

[13] Wen,F.& Li,X.,(2012)"An improved dynamic id-based remote user authentication with key agreement scheme," Computers & Electrical Engineering, Vol. 38, No. 2, pp 381–387.

[14] Xiang, T.,Wong,K.& Liao,X.,(2008) "Cryptanalysis of a password authentication scheme over insecure networks," Journal of Computer and System Sciences, Vol. 74, No. 5, pp 657–661.

[15] Chen, B.L.,Kuo,W.C.& Wu, L. C.,(2014) "Robust smart-card-based remote user password authentication scheme", International Journal of Communication Systems, Vol. 27, No. 2, pp 377–389.

[16] Li, X.,Niu,J., Khan, M. K. & Liao, J., (2013) "An enhanced smart card based remote user password authentication scheme", Journal of Network and Computer Applications, Vol. 36, No. 5, pp 1365–1371.

[17] Wei, J., Liu, W. & Hu, X., (2016) "Secure and efficient smart card based remote user password authentication scheme", International Journal of Network Security, Vol. 18, No. 4, pp 782–792.

[18] Tsai, C. Y., Pan, C. S. & Hwang, M. S., (2016) "An Improved Password Authentication Scheme for Smart Card", Recent Development in Intelligent Systems and Interactive Applications, Vol. 541, pp 194-199.

[19] Xu, J.,Zhu, W. T. & Feng, D. G., (2009) "An improved smart card based password authentication scheme with provable security", Computer Standards and Interfaces, Vol. 31, No. 4, pp 723-728.

[20] Kocher, P., Jaffe, J. & Jun, B., (1999) "Differential power analysis", Lecture Notes in Computer Science, Vol. pp 388-397.

[21] Dolev,D.& Yao, A. C. (1983) "On the security of public key protocols", IEEE Transactions on Information Theory, Vol. 29, No. 2, pp 198-208.

[22] Burrow, M.,Abadi, M. & Needham, R., (1990) "A Logic of Authentication", ACM Transactions on Computer Systems, Vol. 8, No. 1, pp 18-36.

[23] Jung, J.,Kang, D., Lee, D. & Won, D., (2017) "An Improved and Secure Anonymous Biometric-Based for Authentication with Key Agreement Scheme for the Integrated EPR Information System," PLOS One, DOI:10.1371/journal.pone.0169414.

[24] Lu, Y.,Li, L.,Yang, X. & Yang, Y., (2015) "Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards," PLOS One, 10(5):e0126323. Doi:10.1371/journal.pone.0126323.

[25] Moon, J.,Choi,Y.,Jung,J.& Won, D., (2015) "An Improvement of Robust Biometrics-Based Authentication and Key Agreement Scheme for Multi-Server Environment Using Smart Cards," PLOS One, 10(12):e0145263. Doi:10.1371/journal.pone.0145263.

[26] Das,A.K.,Paul, N. R. & Tripathy, L., (2012) "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," Information Sciences, Vol. 209, No. 20, pp 80-92.

[27] Dai, W., Crypto++ Library 5.6.1, Available online: http://www.cryptopp.com (accessed on 5 Dec. 2016).

AUTHORS

Raphael Nyirongo received the B.E. degree in Mathematics from Domasi College of Education, Malawi and is currently a Master Degree student with the Department of Mathematics, Chancellor College, University of Malawi, Malawi. He is currently a Mathematics teacher at Ndirande Hill Secondary School from 2013, Blantyre, Malawi. His research interests include computational mathematics, information security, cryptography and formal proof.

Solomon Kuonga received the B.E. degree in Mathematics from University of Livingstonia and is currently a Master Degree student with the Department of Mathematics, Chancellor College, University of Malawi, Malawi. He is also working as a part time lecturer at Chancellor College, University of Malawi from 2017. He was the Mathematics teacher at Mvera Girls Private Secondary School, Mvera, Malawi at 2016. His research interests include computational mathematics, information security, cryptography and formal proof.

Patrick Ali received the M.Sc. and the Ph.D. degree from Department of Mathematics, Chancellor College, University of Malawi in 2006 and from the Department of Mathematics, University of KwaZulu-Natal, South Africa in 2011, respectively. He is a senior lecturer at the Department of Mathematical Sciences, Chancellor College, University of Malawi from 2006 and is the current Head of Department. He has been an active researcher in graph theory and combinatorial matrix theory. He achieved the research grant from IMU-Simons African Fellowship Grant at 2016. He also achieved two conference awards of the second best PhD student talk at the 52nd SAMS Annual Congress at 2009 and the best PhD student talk at the Faculty of Science and Agriculture Postgraduate Research Day at 2010.

Levis Eneya received the Ph.D. degree from the Humboldt University of Berlin, Germany in 2010. He is the current Dean of Science and is a Senior Lecturer in the Department of Mathematics, University of Malawi, Malawi. Before becoming dean of faculty in January 2015. He has been an active researcher in optimisation, mathematical modelling, and strengthening mathematics teaching and learning through problem solving. He has worked on developing efficient optimization methods for minimizing energy functionals; infectious diseases modelling; and he is currently working on transport optimization and logistics in value chain analysis, and optimization of transport networks in cities. He is also in a team of five, on a collaborative project "Improving Quality and Capacity of Mathematics Teacher Education in Malawi" between the University of Malawi and University of Stavanger in Norway, funded by the NORAD (2014 - 2018). He also served as president of the Southern Africa Mathematical Sciences Association (SAMSA) from 2012 - 2014.

Hyunsung Kim received the M.Sc. and Ph.D degrees in computer engineering from Kyungpook National University, Korea, in 1998 and 2002. He is a Professor with the Department of Cyber Security, Kyungil University, Korea from 2012. Furthermore, he is currently a visiting professor at the Department of Mathematical Sciences, Chancellor College, University of Malawi, Malawi from 2015. He also was a visiting researcher at Dublin City University for 2009. From 2000 to 2002, he worked as a senior researcher at Ditto Technology. He had been an associate professor from 2002 to 2012 with the Department of Computer Engineering, Kyungil University. His research interests include cryptography, VLSI, authentication technologies, network security, ubiquitous computing security and security protocol.