PRIVACY PRESERVING USER AUTHENTICATION SCHEME BASED ON SMART CARD

Beaton Kapito\textsuperscript{1}, Patrick Ali\textsuperscript{1}, Levis Eneya\textsuperscript{1} and Hyunsung Kim\textsuperscript{1,2}

\textsuperscript{1}Mathematical Sciences Department, University of Malawi, Chancellor College, Zomba, Malawi
\textsuperscript{2}Department of Cyber Security, Kyungil University, Kyungbuk, Korea

ABSTRACT

One of the most commonly used user authentication mechanisms is two factor authentication based on smart card and password. The core feature of the scheme is to enforce that the user must have the smart card and know the password in order to gain access to server. Recently, Liu et al. proposed a smart card based password authentication scheme and argued that it is secure against insider attack, replay attack and man in the middle attack and provides perfect forward secrecy. In this paper, we show security weaknesses in Liu et al.’s scheme focused on off-line password guessing attack and masquerading attack and it does not provide perfect forward secrecy and anonymity. Accordingly, we propose a privacy preserving user authentication scheme based on smart card, denoted as PUAS, to remedy these security weaknesses and to provide anonymity and perfect forward secrecy. PUAS is more secure with a bit of computational overhead to support several positive properties in security and privacy.

KEYWORDS

User Authentication, Password Authentication, Smart Card, Bilinear Pairing, Privacy

1. INTRODUCTION

The rapid progress of networks facilitates more and more computers connecting together to exchange great information and share system resources. Password authentication with smart card is one of the most convenient and effective authentication mechanisms for remote systems to assure one communicating party of the legitimacy of the corresponding party by acquisition of corroborative evidence. This technique has been widely deployed for various kinds of authentication applications, such as remote login, online banking, e-commerce and e-health [1-3]. Since Lamport proposed the first remote authentication scheme based on the passwords, a series of authentication schemes have been proposed to improve system security and computation efficiency [4-8]. Lamport’s scheme is based on a password table maintained by a server, which suffers not only from password attacks but also suffers from the cost of protecting and maintaining the password table [5]. To enhance the security of the password based authentication scheme, Chang and Wu introduced password and smart card based two factor user authentication scheme [6]. The main drawback of Chang and Wu’s scheme is using static identity that publicly transmitted identity will reveal user privacy. To conquer the issue, Das et al. proposed an authentication scheme using dynamic identity [7]. However, Liao et al. showed that Das et al.’s scheme cannot resist user impersonation attack and also proposed an improved scheme with mutual authentication [8].
In 2009, Xu et al. proposed a novel user authentication and claimed that their scheme is secure against various attacks [9]. However, Song and Sood et al. found that Xu et al.’s scheme has some weaknesses and proposed improved schemes [5, 10]. Subsequently, Chen et al. pointed out that there are vulnerabilities on Song and Sood et al.’s schemes [11]. Then, Chen et al. presented an enhanced version to solve the weaknesses. Recently, Li et al. claimed that Chen et al.’s scheme is still insecure and proposed a modified smart card based remote user password authentication scheme [12]. Unfortunately, Liu et al. showed that there are weaknesses in Li et al.’s scheme, such as from a man-in-the-middle attack and an insider attack and proposed a remedy scheme [13].

Hence, the purpose of this paper is to provide cryptanalysis on Liu et al.’s scheme and proposes a new privacy preserving user authentication scheme based on smart card, denoted PUAS. First of all, we will show Liu et al.’s scheme is weak against off-line password guessing attack and masquerading attack, and does not provide perfect forward secrecy and anonymity. To solve the weaknesses in Liu et al.’s scheme, PUAS adopts dynamic identity and bilinear pairing, which could provide privacy.

This paper is organized as follows. Section 2 reviews Liu et al.’s smart card based password authentication scheme. In Section 3, security weaknesses will be shown against Liu et al.’s authentication scheme. A new privacy preserving user authentication scheme is proposed in Section 4 with the security and performance analyses at Section 5. Section 6 concludes this paper.

2. REVIEW OF LIU ET AL.’S AUTHENTICATION SCHEME

Liu et al. proposed a smart card based password authentication scheme, which is consisted of four phases: registration phase, login phase, authentication phase and password change phase [13]. Liu et al. argued that their scheme can achieve mutual authentication and users can freely choose and change their passwords. This section reviews Liu et al.’s scheme briefly. Table 1 shows definition of notations used in this paper.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>The server</td>
</tr>
<tr>
<td>$U_i$</td>
<td>The $i^{th}$ user</td>
</tr>
<tr>
<td>$SC$</td>
<td>The smart card</td>
</tr>
<tr>
<td>$x$</td>
<td>The master secret key of $S$</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>The identity of $U_i$</td>
</tr>
<tr>
<td>$PW_i$</td>
<td>The password of $U_i$</td>
</tr>
<tr>
<td>$T_i$</td>
<td>The timestamp $i$</td>
</tr>
<tr>
<td>$sk$</td>
<td>The shared session key</td>
</tr>
<tr>
<td>$h(.)$</td>
<td>A secure hash function</td>
</tr>
<tr>
<td>$E(.)$</td>
<td>A symmetric key encryption based on AES</td>
</tr>
<tr>
<td>$D(.)$</td>
<td>A symmetric key decryption based on AES</td>
</tr>
<tr>
<td>$\delta(.)$</td>
<td>A bilinear map</td>
</tr>
<tr>
<td>$+$</td>
<td>Exclusive-or operation</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
</tbody>
</table>
2.1. Registration Phase

Before starting Liu et al.’s authentication scheme, the server $S$ selects the master secret key $x$ and a one-way hash function $h(\cdot)$. The registration phase is as follows:

Step 1. The user $U_i$ selects his/her identity $ID_i$, password $PW_i$, and a random number $r$, and then computes $h(r||PW_i)$. $U_i$ submits $\{ID_i, h(r||PW_i)\}$ to $S$ for registration over a secure channel.

Step 2. $S$ computes $A_i = h(ID_i \oplus x || h(x))$, $B_i = A_i \oplus h(r||PW_i)$, and $C_i = h(A_i \oplus ID_i || h(r||PW_i))$.

Step 3. $S$ stores the data $\{B_i, C_i, h(\cdot)\}$ on a new smart card (SC) and issues it to $U_i$ over a secure channel.

Step 4. $U_i$ stores the random number $r$ into SC.

2.2. Login Phase

This phase is invoked whenever $U_i$ wants to login to $S$. The steps of this phase are shown as follows:

Step 1. $U_i$ inserts his/her SC into a card reader and inputs $ID_i$ and $PW_i$.

Step 2. SC first computes two parameters $A_i' = B_i \oplus h(r||PW_i)$ and $C_i' = h(A_i \oplus ID_i \oplus h(r||PW_i))$. Then, SC examines whether $C_i'$ is equal to $C_i$. If the equation holds, SC continues to perform Step 3; otherwise, SC terminates this session.

Step 3. SC randomly selects a number $a$ and computes $D_i = h(ID_i \oplus a)$ and $E_i = A_i \oplus a \oplus T_i$, where $T_i$ is the current timestamp of $U_i$.

Step 4. SC sends the login request message $\{ID_i, D_i, E_i, T_i\}$ to $S$.

2.3. Authentication Phase

After completing this phase, $U_i$ and $S$ can mutually authenticate each other and establish a shared session key for the subsequent secret communication.

Step 1. $S$ verifies whether $ID_i$ is valid and $T_i' - T_{\Delta T}$, where $T_i'$ is the time of receiving the login request message and $\Delta T$ is a valid time threshold. If both conditions are true, $S$ continues to execute Step 2; otherwise, $S$ rejects the login request.

Step 2. $S$ computes $A_i = h(ID_i \oplus x || h(x))$, $A_i' = E_i \oplus A_i \oplus T_i$ and $D_i' = h(ID_i \oplus a')$. Then, $S$ compares whether $D_i' equals D_i$. If they are equal, $S$ confirms that $U_i$ is valid and the login request is accepted; otherwise, the login request is rejected.

Step 3. $S$ randomly selects a number $\beta$ and computes $F_i = h(ID_i \oplus \beta)$ and $G_i = A_i \oplus \beta \oplus T_i$.

Step 4. $S$ sends the mutual authentication message $\{F_i, G_i, T_i\}$ to $U_i$.

Step 5. Upon receiving the message $\{F_i, G_i, T_i\}$, SC checks the validity of $T_i$. If $T_i' - T_i \leq \Delta T$, where $T_i'$ is time of receiving the mutual authentication message, $S$ continues to perform Step 6; otherwise, SC terminates this connection.

Step 6. SC computes $\beta' = G_i \oplus A_i \oplus T_i$, and $F_i' = h(ID_i \oplus \beta')$, and then checks whether $F_i'$ equals $F_i$. If they are equal, the validity of $S$ is authenticated; otherwise, the session is terminated.

Step 7. $U_i$ and $S$ construct a shared session key $sk = h(a||f'||h(A_i \oplus ID_i)) = h(a'\|h(A_i \oplus ID_i))$. 


2.4. Password Change Phase

Liu et al.‘s protocol allows users to freely update their passwords. The password change phase works as follows:

Step 1. \( U_i \) inserts his/her SC into a card reader, enters his/her old identity \( ID_i \) and password \( PW_i \), and requests to change the password.

Step 2. \( S \) computes \( A_i = B_i \oplus h(\|ID_i\|\|PW_i\|) \) and \( C_i = h(A_i'\|ID_i\|\|h(\|PW_i\|)) \), and then checks whether \( C_i \) equals \( C_i^r \) that is stored in \( SC \). If the equation holds, \( U_i \) submits the new password \( PW_i^{new} \). Otherwise, \( S \) rejects the password change request.

Step 3. \( S \) computes \( B_i^{new} = A_i' \oplus h(\|PW_i^{new}\|) \) and \( C_i^{new} = h(A_i'\|ID_i\|\|h(\|PW_i^{new}\|)) \). Then, \( S \) replaces \( B_i \) and \( C_i \) with \( B_i^{new} \) and \( C_i^{new} \), respectively.

3. Security Weakness of Liu et al.’s Authentication Scheme

This section provides security weaknesses in Liu et al.’s authentication scheme, which are focused on off-line password guessing attack (OPGA), masquerading attack (MA) with smart card loss attack, no perfect forward secrecy (PFS) and no anonymity.

3.1 Off-line Password Guessing Attack

Kocher et al. explained that various information stored in SCs could be extracted by physically monitoring its power consumption [14]. So it is possible to say that if a user loses his/her SC, all information in SC may be revealed to the attacker. In Liu et al.’s authentication scheme, SC stores important information for user login and authentication phases. Furthermore, for the proper attack, it is assumed that the attacker could listen and get the messages from the communication between \( U_i \) and \( S \).

Using SC information of \( U_i \), which are \{\( B_i, C_i, r, h(\cdot) \)\} and \( ID_i \) from the communication messages \{\( ID_i, D_i, E_i, T_i \)\} and \{\( F_i, G_i, T_i \)\}, attacker can perform OPGA to find \( PW \) as follows. (1) The attacker guesses a password candidate \( PW_i' \) and computes \( A_i' = B_i \oplus h(\|PW_i'\|) \) and \( C_i' = h(A_i'\|ID_i\|\|h(\|PW_i'\|)) \). (2) The attacker checks whether \( C_i' \) is equal to \( C_i \). If they are the same, the password guessing is successful. Otherwise, the attacker repeats Steps (1) and (2) until the correct password is withdrawn.

3.2 Masquerading Attack

When an attacker gets or steals the user’s SC in OPGA, he/she can login and authenticate to \( S \), and compute the session key \( sk \) between \( U_i \) and \( S \). So the attacker can impersonate the legitimate user \( U_i \). It is critical problem that the attacker can be authenticated to \( S \) using user’s SC information. The attacker can illegally extract the secret values in the user’s SC and get some important information.

So, the attacker impersonate \( U_i \) after the success of OPGA as follows. (1) The attacker computes \( A_i, C_i, D_i \) and \( E_i \), using \( ID_i, PW_i, B_i, C_i \) and \( r \), generates a random number \( a_{old} \) and sends \{\( ID_i, D_i, E_i, T_i \)\} to \( S \). (2) \( S \) cannot figure out that the message is from the attacker. So, \( S \) authenticates the attacker’s message, computes \( A_i, a_{old}, D_i', F_i \) and \( G_i \), generates \( \beta \), and then sends \{\( F_i, G_i, T_i \)\} to the attacker. Therefore, the attacker can login and be authenticated to \( S \) with forming the session key \( sk = h(a_{old}\|\|\beta\|\|h(A_i'\oplus ID_i)) \), which is the same to \( S \)‘s session key \( sk \).
3.3 No Perfect Forward Secrecy

PFS is a feature of specific key agreement schemes that gives assurances the session key will not be compromised even if the private key of the server is compromised. But Liu et al.’s scheme does not achieve PFS.

In Liu et al.’s scheme, the attacker can compute the all session keys between $U_i$ and $S$ if the attacker knows one of long-term keys as follows. (1) The attacker gets $\{ID, D_i, E_i, T_i\}$ and $\{F_i, G_i, T_s\}$ in the previous communication between $U_i$ and $S$. (2) The attacker knows one of long-term secret $x$ of $S$ and could derive $A_i'=h(ID_i\oplus x)||h(x)$. So the attacker can compute $\alpha'=E_i \oplus A_i' \oplus T_i$ and $\beta'=G_i \oplus A_i' \oplus T_s$. After that, the attacker could compute $sk'=h(\alpha'||\beta'||h(A_i\oplus ID_i))$. Therefore, Liu et al.’s scheme does not provide PFS.

3.4 No Anonymity

Liu et al.’s authentication scheme does not provide the anonymity. In this scheme, $U_i$ sends his/her own identifier $ID$ to $S$ over public communication without any protection. Therefore, an attacker can easily get $ID$ from public communications. This results in the identity exposure problem. Therefore, the lack of anonymity in Liu et al.’s scheme raises privacy related problems that need to be addressed to Internet of things. To solve this problem, it is necessary to use anonymity mechanism in the communication.

4. PRIVACY PRESERVING USER AUTHENTICATION SCHEME

In this section, we propose a new privacy preserving user authentication scheme (PUAS) based on smartcard and bilinear pairing. PUAS could solve all the security problems and privacy issue depicted in Liu et al.’s authentication scheme. In PUAS, there are also two participants, $U_i$ and $S$, which is consisted of four phases, registration phase, login phase, authentication phase and password changing phase.

4.1 Bilinear Pairings

Let $G_1$ be an additive cyclic group generated by $P$ whose order is a prime $q$, and $G_2$ is a multiplicative cyclic group of the same order. A map $\hat{\epsilon}(\cdot): G_1 \times G_1 \rightarrow G_2$ is called a bilinear mapping if it satisfies the following properties:

- **Bilinear property**: $\hat{\epsilon}(aP, bQ) = \hat{\epsilon}(P, Q)^{ab}$, for all $P$ and $Q$ in $G_1$ and all $a$ and $b$ in $\mathbb{Z}_q^*$.
- **Non-degenerate**: There exists $P$ and $Q$ in $G_1$ such that $\hat{\epsilon}(P, Q) = 1$.
- **Computable**: There is an efficient algorithm to compute $\hat{\epsilon}(P, Q)$ for all $P$ and $Q$ in $G_1$.

We note that $G_1$ is the group of points on an elliptic curve and $G_2$ is a multiplicative subgroup of a finite field. Typically, the mapping $\hat{\epsilon}$ will be derived from either the Weil or the Tate pairing on an elliptic curve over a finite field.

4.2 PUAS

This section proposes a new privacy preserving user authentication scheme to solve the security and privacy problems in Liu et al.’s scheme, which provides mutual authentication between the user and the server and does not require time synchronization. In order to prevent the problems of clock synchronization or a delay-time limitations, PUAS adopts the challenge-response mechanism. The security of PUAS is based on bilinear Diffie-Hellman problem (BDHP) and one-wayness of the hash function.
For the system setup, system administrator sets a Bilinear map \( e(\cdot) \) and \( h(\cdot) : \{0,1\}^* \rightarrow G_1 \), which is a cryptographic hash function with an output size of 512 bits. Furthermore, system administrator selects encryption and decryption function, \( E(\cdot) \) and \( D(\cdot) \), based on Advanced Encryption Standard (AES). Then, the system administrator publishes the system parameters \( e \in G_1, G_2, e(\cdot), q, P, h(\cdot), E(\cdot), D(\cdot) \). PUAS consists of four phases namely; registration, login, authentication and password change phases as shown in Figures 1 and 2.

[Registration Phase] This phase is executed by the following steps when a new user \( U_i \) wants to be registered to the server \( S \).

RP1. When \( U_i \) wants to register with \( S \), \( U_i \) selects his/her identity \( ID_i \) and password \( PW_i \), generates a random number \( r \) and computes \( RPW_i = h(PW_i || r) \). \( U_i \) submits \( ID_i \) and \( RPW_i \) to \( S \).

RP2. On receiving the registration request, \( S \) generates a random number \( w \) and computes \( A_i = h(ID_i \oplus x), B_i = h(A_i) \oplus \| RPW_i, C_i = E_i(ID_i || w) \) and \( D_i = (h(A_i) || ID_i || RPW_i) \). \( S \) personalizes \( A \) with the parameters \( \{ B_i, C_i, D_i, e(\cdot), h(\cdot), P \} \) and sends it to \( U_i \) over a secure channel.

RP3. \( U_i \) computes \( P_i = h(ID_i || PW_i) \oplus r \) and writes \( P_i \) into his/her \( SC \).

[Login Phase] If \( U_i \) wants to login, \( U_i \) inserts \( SC \) in a card reader and inputs \( ID_i' \) and \( PW_i' \). Then, \( SC \) performs the following operations:

LP1. \( SC \) extracts \( r' = P_i \oplus h(ID_i || PW_i') \) and computes \( RPW_i' = h(PW_i' || r'), h(A_i)' = B_i \oplus RPW_i' \) and \( D_i' = (h(A_i)' || ID_i || RPW_i) \). \( SC \) verifies \( D_i' \) with stored \( D_i \). If it does not hold, \( SC \) rejects \( U_i \)’s login request. Otherwise, \( SC \) generates a fresh random number \( \alpha \) computes \( E_i = aP \) and \( F_i = h(C_i || h(A_i)' || E_i) \) and sends a login request message \( \{ C_i, E_i, F_i \} \) to \( S \).

[Authentication Phase] Upon receiving the message \( \{ C_i, E_i, F_i \} \), \( S \) and \( SC \) execute the following steps for mutual authentication and session key agreement as follows:

AP1. \( S \) computes \( ID_i || w = D_i (C_i), A_i' = h(ID_i \oplus x) \) and \( F_i' = h(C_i || h(A_i') || E_i) \) and verifies \( F_i' \) with the received \( F_i \). If it does not hold, \( S \) rejects the request. Otherwise, \( S \) generates a fresh random number \( \beta \) computes \( L_i = \beta P, SK = \beta E_i, C_i'' = E_i(ID_i || \beta) \oplus SK \) and \( M_i = h(ID_i || C_i'' || h(A_i') || L_i || SK) \) and sends back a message \( \{ L_i, C_i'', M_i \} \) to \( U_i \). \( SC \) computes \( SK' = \alpha L_i, C_i''' = C_i' \oplus SK' \) and \( M_i' = h(ID_i || C_i'|| h(A_i') || L_i || SK) \) and verifies \( M_i' \) with the received \( M_i \). If it does not hold, \( U_i \) rejects the request. Otherwise, \( SC \) updates \( C_i \) with \( C_i''' \).

After the mutual authentication and session key agreement between \( U_i \) and \( S \), they could securely communicate with each other based on the established \( SK \).

[Password Change Phase] This phase is invoked whenever \( U_i \) wants to change his/her password. By invoking this phase, \( U_i \) can easily change his/her password without taking any assistance from \( S \). If \( U_i \) wants to change his/her password, \( U_i \) inserts \( SC \) in a card reader and inputs \( ID_i' \) and \( PW_i' \). Then, \( SC \) performs the following operations:

PC1. \( SC \) extracts \( r'' = P_i \oplus h(ID_i || PW_i') \) and computes \( RPW_i'' = h(PW_i'' || r''), h(A_i)' = B_i \oplus RPW_i' \) and \( D_i'' = (h(A_i)' || ID_i || RPW_i) \). \( SC \) verifies \( D_i'' \) with stored \( D_i \). If it does not hold, \( SC \) rejects \( U_i \)’s request. Otherwise, \( SC \) asks \( U_i \) to input a new password \( PW_i^{new} \) and computes \( RPW_i^{new} = h(PW_i^{new} || r''), B_i'' = h(A_i)' \oplus RPW_i^{new} \) and \( D_i'' = (h(A_i)' || ID_i || RPW_i^{new}) \). \( SC \) replaces \( B_i'' \) and \( D_i'' \) by \( B_i \) and \( D_i \), respectively.
$U_i$

Selects $ID_i$ and $PW_i$
Generates $r$
Computes $RPW_i = h(PW_i \| r)$

$S$

Verifies $ID_i$
Generates $w$
Computes $A_i = h(ID_i \oplus x)$
$B_i = h(A_i \oplus RPW_i)$
$C_i = E_x(ID_i \| w)$
$D_i = h(h(A_i) \| ID_i \| RPW_i)$
Stores $\{B_i, C_i, D_i, \ell(.), h(.), P\}$ in $SC$

Stores $P = h(ID_i \| PW_i \oplus r)$ into $SC$

Figure 1. Registration phase of PUAS

$U_i$

Inputs $ID_i', PW_i'$
Computes $r' = P \oplus h(ID_i' \| PW_i')$
$RPW_i' = h(PW_i' \| r')$
$h(A_i') = B_i \oplus RPW_i'$
$D_i' = h(h(A_i') \| ID_i' \| RPW_i')$
Verifies $D_i' = D_i$
Generates a random $\alpha$
Computes $E_i = \ell(\alpha, P)$
$F_i = h(C \| h(A_i') \| E_i)$

$S$

Computes $ID_i \| w = D_i(C_i)$
$A_i' = h(ID_i' \oplus x)$
$F_i' = h(C \| h(A_i') \| E_i)$
Verifies $F_i' = F_i$
Generates $\beta$
Computes $L_i = \ell(\beta, P)$
$SK = \ell(\beta, E_i)$
$C_i' = E_x(ID_i' \| \beta) \oplus SK$

$M_i = h(ID_i' \| C_i' \| h(A_i') \| L_i \| SK)$

$C_i' = C_i' \oplus SK'$
$M_i' = h(ID_i' \| C_i' \| h(A_i') \| L_i \| SK')$
Verifies $M_i' = M_i$
Updates $C_i$ with $C_i'$

Figure 2. Login and authentication phases of PUAS
5. ANALYSIS

This section provides security and performance analyses of PUAS. For the security analysis, three security properties of PUAS are considered mainly to show security against OPGA and MA with smart card loss attack, and provide PFS. Furthermore, PUAS provides anonymity. After that, we provide performance analysis of PUAS and comparison with the related schemes in [12-13].

5.1 Security Analysis

For the proper security analysis, we follow the Dolev-Yao threat model [15] and consider the risk of side-channel attack [14] to construct the threat assumptions which are described as follows:

- An adversary can be either \( U \) or \( S \). Any registered user can act as an adversary.
- An adversary can eavesdrop every communication in public channels. He/she can capture any message exchanged between \( U_i \) and \( S \).
- An adversary has the ability to alter, delete or reroute the captured message.
- Information can be extracted from the smart card by examining the power consumption of the card.

**[Off-line Password Guessing Attack]** Suppose an user loses his/her smart card and an adversary gets it, and extracts all of the information stored on the smart card \( \{ B_i, C_i, D_i, P_i, e(\cdot), h(\cdot), P \} \) by power consumption analysis. However, he/she cannot obtain any password related information. There are three values \( B_i, D_i \) and \( P_i \) related to the password. Each value is related with two unknown values to the attacker. Therefore, PUAS is strong against OPGA because the attacker could not do any guessing attack due to two unknown values in each value.

**[Masquerading Attack]** Suppose an adversary intercepts all of the message \( \{ C_i, E_i, F_i \} \) and \( \{ L_i, C_i', M_i \} \) transmitted in public channel between \( U_i \) and \( S \), and steals the smart card of \( U_i \) to get \( \{ B_i, C_i, D_i, P_i, e(\cdot), h(\cdot), P \} \). There are two possible attacks to the attacker to masquerade \( U_i \) as or \( S \). For \( U_i \) MA, the attacker need to form a legal login message \( \{ C_i, E_i, F_i \} \). However, the attacker could not form a legal value \( F_i \) due to the lack of knowledge on \( A_i \), which is related with the password guessing attack. For \( S \) MA, the attacker need to form a message \( \{ L_i, C_i', M_i \} \). However, the attacker could not form a legal value \( M_i \) due to the lack of knowledge on \( A_i \) and \( C_i' \), which are related to the long term secret \( x \) of \( S \). Therefore, PUAS is strong against MA.

**[Perfect Forward Secrecy]** The security of PUAS is based on the bilinear pairing. In PUAS, a session key is computed between \( U \) and \( S \) as \( SK=e(\alpha, \beta P)=e(\alpha P, \beta) \). Even if \( S \)'s long term secret key \( x \) is compromised, the adversary cannot retrieve \( \alpha \) nor \( \beta \) from \( E_i \) and \( L_i \) to generate the session key. The session key of PUAS is based on the difficulty of BDHP. Thereby, PUAS provides PFS.

**[Anonymity]** To address anonymity, PUAS uses \( C_i \), which is an amplified encrypted identity and is regularly changed in each session. Only \( S \) could generate and check the identity of \( U \) by using the long term secret key \( x \). Furthermore, the renewal of is not only depending on \( x \) but also depending on the session key \( SK \). Thereby, only legal entity could know the amplified identity \( E_i(ID||\beta) \). Thereby, PUAS provides anonymity.

<table>
<thead>
<tr>
<th>Properties</th>
<th>OPGA</th>
<th>MA</th>
<th>PFS</th>
<th>Anonymity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al. [12]</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Liu et al. [13]</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PUAS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2. Security and privacy comparison of user authentication schemes.
5.2 Performance Analysis

This subsection evaluates the performance of PUAS in terms of computational cost. Table 3 shows a comparison of PUAS and the related schemes [12-13]. From Table 3, we can see that PUAS has a bit overhead than the other schemes to provide security and privacy. The security and privacy are top most important in any cryptographic schemes. For the efficient comparison, we only considered login and authentication phases’ operational requirements with the notations $M$, $E$, $B$, $S$ and $H$ for multiplication/division operation, modulus exponential operation, bilinear pairing operation, symmetric encryption/decryption operation and hash operation, respectively.

Table 3. Performance comparison of user authentication schemes.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>$U_i$</th>
<th>$S$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu et al. [12]</td>
<td>$1M+3E-4H$</td>
<td>$3E-3H$</td>
<td>$1M-6E-7H$</td>
</tr>
<tr>
<td>Liu et al. [13]</td>
<td>$6H$</td>
<td>$6H$</td>
<td>$12H$</td>
</tr>
<tr>
<td>PUAS</td>
<td>$2B-5H$</td>
<td>$2B-2S-4H$</td>
<td>$4B-2S-9H$</td>
</tr>
</tbody>
</table>

6. CONCLUSION

In this paper, we provided analyses on Liu et al.’s smart card based password authentication scheme. Our research showed that Liu et al.’s scheme is vulnerable to the password guessing attack and impersonation attack and furthermore does not provide perfect forward secrecy not anonymity. As a remedy scheme of Liu et al.’s scheme, we proposed a privacy preserving user authentication scheme (PUAS) based on smart card. We demonstrated that PUAS has much better security features and performance when compared to Liu et al.’s scheme and the related other schemes.

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


AUTHORS

Beaton Kapito received the B.E. degree in Mathematics from Chancellor College of University of Malawi and is currently a Masters Degree student with the Department of Mathematics, Chancellor College, University of Malawi. He is also working as a part time lecturer at Chancellor College, University of Malawi from 2017. He is also an adjunct lecturer at Malawi Adventist University an affiliate of The University of Eastern Africa, Baraton. He has been a Mathematics teacher at Chikwawa Secondary School, Soche Adventist Secondary School and Chileka Mission Secondary School. His research interest is in Cryptography; His Masters thesis proposal is “Privacy Preserving Authenticated Key Agreement for Internet of Things.”

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 optimization, 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.