International Journal of Security and Its Applications
Vol.8, No.4 (2014), pp.307-320
http://dx.doi.org/10.14257/ijsia.2014.8.4.28

Privacy Preserving Three-party Authenticated Key Agreement Protocol using Smart Cards

Suyeon Park* and Hee-Joo Park**

*School of Computer IT Engineering, Daegu University
**(Corresponding Author) Dept. of Cyber Security, Kyungil University
hjpark@kiu.ac.kr

Abstract

How to make people keep both security and privacy in communication networks has been a hot topic in recent years. Researchers proposed three party authenticated key agreement (3PAKA) protocols to answer this question, which allows two parties to agree a new secure session key with the help of a trusted server. Recently, Yang et al. proposed a provably secure 3PAKA protocol. However, this paper finds out Yang et al.'s protocol has a security weakness against password guessing attack and two lack properties in authentication for password updating phase and privacy preserving. Furthermore, we propose anew privacy preserving 3PAKA (P_3PAKA) protocol using smart cards to solve the security problems in Yang et al.'s protocol. It provides user anonymity and un-traceability by adopting dynamic identifier depending on each session's nonce. Comparing with other typical 3PAKA protocols, P_3PAKA protocol is more secure while maintaining efficiency.

Keywords: *Authenticated key exchange, privacy-preserving, three-party protocol, smart card*

1. Introduction

Client's program tries to communicate with the server's program over insecure networks like Internet [1]. In the process, the identity and a secret password of a client are used for mutual authentication and access control. However, password can be compromised during transmission, if an efficient protocol is not followed. Also certain systems in a network needs to change the client's password periodically for the protection of the system resources from adversary, and until a secured password change protocol that allows the client to change the password safely, the systems are not well protected. Since Lamport in 1987 first proposed a remote password authentication protocol for the insecure communication, many researchers have proposed theirprotocols to address password authentication problems [2-8].

After the authentication, two communicating parties usually need to establish session keys to protect the confidentiality, integrity and authenticity of transmitted data over insecure channel. Key agreement protocol is used for the protection and Diffie and Hellman proposed the first key agreement protocol [9], which does not use identity authentication mechanism and suffers from the man in the middle attack. Therefore, authentication and key agreement are combined to yield the concept of authenticated key agreement (AKA). AKA has great interest in network security field and many AKA protocols have been proposed [10-18]. AKA protocols can be divided into two categories, two-party AKA (2PAKA) protocols and three-party AKA (3PAKA) protocols, depending on whether an online trusted server is required or not. 2PAKA

ISSN: 1738-9976 IJSIA

protocols have an advantage of no online trusted server requirement but have lack of network expandability, which need to share a secret beforehand between each communicating pair and requiremany pre-shared secrets especially in a network with large number of users. However, 3PAKA protocols require each user share a pre-shared secret with a trusted server and thereby they could support the network expendability.

Some 3PAKA protocols using smart cards have been proposed [14-18]. Juang proposed a 3PAKA protocol using smart cards. However, Juang's protocol requires heavy communication overhead, which has five rounds of messages, and hence cannot support rapid response [14]. Chang, *et al.,* proposed a 3PAKA protocol using smart cards without any modular exponentiation operation; however, it needs five rounds of messages [15]. Kwon, *et al.,* proposed a smart card-based 3PAKA protocol which requires only three rounds of messages [16]. Unfortunately, as pointed out by Yoon and Yoo, Kwon, *et al.,* protocol is vulnerable to impersonation attacks [17]. Yoon and Yoo also proposed an improved 3PAKA protocol with three rounds of messages. However, like Kwon, *et al.,* protocol, Yoon and Yoo's protocol is based on timestamp technique and is not suitable for use in a network with large numbers of users. Recently, Yang, *et al.,* proposed a provably secure 3PAKA protocol using smart cards to reduce the communication overhead and to remove time synchronization problem in timestamp based protocol [18]. Furthermore, they argued that their protocolis secure against various attacks.

There are two purposes of this paper: one is to show security weaknesses in Yang, *et al.,* protocol and the other is to propose a new 3PAKA protocol to solve the problems in Yang, *et al.,* 3PAKA protocol. First of all, this paper shows a security weakness against password guessing attack with lost smart cardand lack of good properties for ubiquitous environment in Yang, *et al.,* protocol. Then, this paper proposes anew privacy preserving 3AKA (P_3PAKA) protocol using smart cards to solve the security problems in Yang, *et al.,* protocol. It provides user anonymity and un-traceability by adopting dynamic identifier depending on each session's nonce.

The rest of this paper is organized as follows. In Section 2, Yang, *et al.,* 3PAKA protocol is reviewed after introducing definitions and notations used in the paper. Section 3 presents security analyses on Yang, *et al.,* 3PAKA protocol. Some required security criteria for 3PAKA are summarized for the goal of a new 3PAKA protocol and a P_3PAKA protocol is proposed to solve the security problems in Yang, *et al.,* protocol and to provide the required security criteria in Section 4. In Section 5, we provide security and performance analyses for P_3PAKA protocol by comparing with the other related protocols. Section 6 concludes the paper.

2. Yang, *et al.,* 3PAKA Protocol

This section reviews Yang, *et al.,* provably secure 3PAKA protocol using smart cards [18]. Yang, *et al.,* 3PAKA protocol is consistedwith four phases: registration phase, login phase, password updating phase and key agreement phase.

2.1. Definitions and Notations

The definitions and notations used throughout this paper are as shown in Table 1. The parameters (p, q, g) and hash functions $H_0 \sim H_9$ are common to all participants. The operator'mod p' will henceforth be omitted as the same in [18].

2.2 Registration Phase

This subsection reviews registration phase by using user$A$'s registration as an example. A

first submits his/her identity information and his/her password pw_A to server$S$ for the registration. If *S* accepts this request, *S* will perform the following steps:

Step 1 :Computes $PW_A = H_0(A, pw_A)$, $x_A = H_1(A, s)$, $x_A' = H_2(A, s)$, $y_A = PW_A \oplus x_A$, $y_A' = PW_A \oplus x_A'$ and $h_A = H_3(x_A, x_A')$.

Step 2 :Destroys x_A, x_A', PW_A and pw_A, and then stores *A*, y_A, y_A' and h_A to the memory of

Table 1. Notations

Notation	Meaning
p	Large prime number
q	Large prime divisor of p - 1
G	Prime order subgroup of Z_p^* and $\|G\| = q$
g	Generator of *G*
$x \leftarrow {}_R Z_p^*$	Random selection of an integer *x* from the set Z_q^*
$x \leftarrow y$	Simple assignment statement
$\oplus$	Bitwise exclusive-OR operator
A, B	Client users; they also denote the identities of users
S	Trusted server; it also denotes the identity of the server
s	Secret key of *S*, which is an integer chosen randomly from Z_q^*
x_A, x_A'	Two long-term private keys of *A*; $x_A = H_1(A, s)$ and $x_A' = H_2(A, s)$
x_B, x_B'	Two long-term private keys of *B*; $x_B = H_1(B, s)$ and $x_B' = H_2(B, s)$
pw_A, pw_B	Passwords with which the smart cards of *A* and *B* are protected
a, b	Ephemeral private keys of *A* and *B*; $a, b \in {}_R Z_q^*$
R_A, R_B	Ephemeral public keys of *A* and *B*; $R_A = g^a \bmod p$, $R_B = g^b \bmod p$
sk_A, sk_B	Session key computed by *A* and *B*, respectively
$H_0 \sim H_9$	Nine independent cryptographic hash functions
$H_0 : \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q^*$, $H_1 : \{0,1\}^* \times Z_q^* \rightarrow Z_q^*$,	
$H_2 : \{0,1\}^* \times Z_q^* \rightarrow Z_q^*$, $H_3 : Z_q^* \times Z_q^* \rightarrow Z_q^*$,	
$H_4 : \{0,1\}^* \times \{0,1\}^* \times G \times G \rightarrow \{0,1\}^{l_4}$	
$H_5 : \{0,1\}^* \times \{0,1\}^* \times Z_q^* \times G \times G \rightarrow \{0,1\}^{l_5}$	
$H_6 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \rightarrow \{0,1\}^{l_6}$	
$H_7 : \{0,1\}^* \times G \times G \rightarrow \{0,1\}^{l_4}$	
$H_8 : \{0,1\}^* \times Z_q^* \times G \times G \rightarrow \{0,1\}^{l_5}$	
$H_9 : \{0,1\}^* \times G \times G \times G \rightarrow \{0,1\}^{l_6}$, where l_4, l_5 and l_6 denote the bit-length of the outputs of H_4, H_5, H_6, H_7, H_8 and H_9, respectively. Hash functions H_1, H_2, H_4, H_5, H_6, H_7, H_8 and H_9 are modeled as random oracles [19]. Note that hash functions H_0 and H_3 do not need to be modeled as random oracles, because they are only used to protect the shared keys in the smart card and do not be used during each run of the protocol.	

A's smart card.

After this, the smart card is protected with the password and user *A* can freely choose and change the password.

2.3. Login Phase

After inserting their smart card into their card reader, *A* and *B* must input their passwords to their smart cards to use them. Their smart cards perform the following verification process,

respectively.

A's smart card computes $PW_A = H_0(A, pw_A)$, $x_A = PW_A \oplus y_A$, $x_A^{'} = PW_A \oplus y_A^{'}$, and $h_A^{'} = H_3(x_A, x_A^{'})$.If$h_A^{'} \neq h_A$,it aborts the execution and destroys $x_A$and $x_A^{'}$.

B's smart card computes $PW_B = H_0(B, pw_B)$, $x_B = PW_B \oplus y_B$, $x_B^{'} = PW_B \oplus y_B^{'}$, and $h_B^{'} = H_3(x_B, x_B^{'})$. If $h_B^{'} \neq h_B$,it aborts the execution and destroys $x_B$and $x_B^{'}$.

2.4 Password Updating Phase

Suppose A wants to change his/her password pw_A to a new password pw_A^*, he/she will perform the following steps:

Step 1 :A inserts his/her smart card into his/her card reader. Then, A inputs his/her old password $pw_A$and new password pw_A^* to the smart card.

Step 2 :The smart card computes $PW_A = H_0(A, pw_A)$ and $PW_A^* = H_0(A, pw_A^*)$ and then replaces y_A and $y_A^{'}$ with $PW_A^* \oplus PW_A \oplus y_A$ and $PW_A^* \oplus PW_A \oplus y_A^{'}$, respectively.

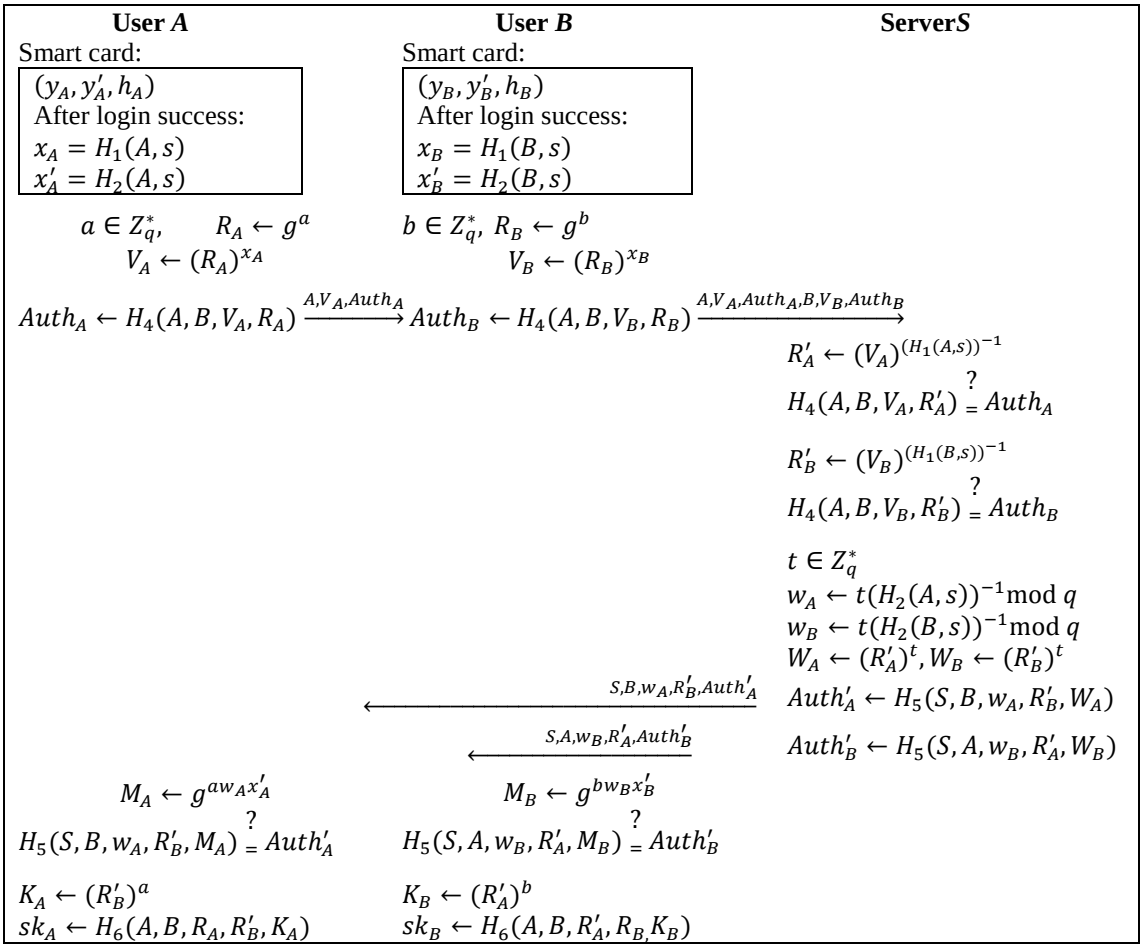

Figure 1. Yang, *et al.*, Protocol

2.5. Key Agreement Phase

Suppose A and B wish to agree on a shared session key. The process of the key agreement is shown in Figure 1. There are three rounds and four messages during each run of the

protocol. The details are as follows:

Round 1: Message 1 ($A \rightarrow B$): $A,\ V_A,\ Auth_A$
A randomly chooses an integer $a \in Z_q^*$ and computes$R_A = g^a$and$V_A = (R_A)^{x_A}$. A then computes $Auth_A = H_4(A,\ B,\ V_A,\ R_A)$ and sends $(A,\ V_A,,\ R_A,\ Auth_A)$ to *B*.

Round 2: Message 2 ($B \rightarrow S$): $A,V_A,Auth_A,\ B,\ V_B,\ Auth_B$
Upon receiving Message 1 from *A*, *B* randomly selects an integer $b \in Z_q^*$, and computes $R_B = g^b$and$V_B = (R_B)^{x_B}$. *B* then computes $Auth_B = H_4(A,\ B,\ V_B,\ R_B)$ and sends $(A,\ V_A,\ Auth_A,\ B,\ V_B,\ Auth_B)$ to *S*.

Round 3: Message 3 ($S \rightarrow A$): $S, B, w_A,\ R'_B,\ Auth'_A$
Message 4 ($S \rightarrow B$): $S, A,\ w_B,\ R'_A,\ Auth'_B$

Upon receiving Message 2 from *B*, *S* computes $R'_A = (V_A)^{(H_1(A,s))^{-1}}$ and checks whether $H_4(A,\ B,\ V_A,\ R'_A) = Auth_A$ holdsor not. If not, *S* terminates the execution. Otherwise *S* assures that the message $(A,\ V_A,\ Auth_A)$ is sent by *A*. *S* continues to compute $R'_B = (V_B)^{(H_1(B,s))^{-1}}$ and checks whether $H_4(A,\ B,\ V_B,\ R'_B) = Auth_B$ holdsor not. If not, *S* terminates the execution. Otherwise, *S* learned that the message $(B,\ V_B,\ Auth_B)$ is indeed sent by *B*. Next, *S* randomly chooses an integer $t \in Z_q^*$ and computes $w_A = t(H_2(A,s))^{-1}$ mod q and $w_B = t(H_2(B,s))^{-1}$ mod q. *S* further computes $W_A = (R'_A)^t$, $W_B = (R'_B)^t$, $Auth'_A = H_5(S,\ B,\ w_A,\ R'_B,\ W_A)$and $Auth'_B = H_5(S,\ A,\ w_B,\ R'_A,\ W_B)$. Finally, *S* destroys t,$W_A$and W_B, and sends $(S,\ B,\ w_A,\ R'_B,\ Auth'_A)$ and $(S,\ A,\ w_B,\ R'_A,\ Auth'_B)$ to *A* and *B*, respectively.

Upon receiving Message 3 from *S*, *A* computes $M_A = g^{aw_A x'_A}$ and checks whether $H_5(S,\ B,\ w_A,\ R'_B,\ M_A) = Auth'_A$ holdsor not. If not, *A* terminates the execution. Otherwise, *A* assures that the message $(S,\ B,\ w_A,\ R'_B,\ Auth'_A)$ is sent by *S*. Subsequently, *A* computes $K_A = (R'_B)^a$ and $sk_A = H_6(A,\ B,\ R_A,\ R'_B,\ K_A)$. Finally, *A* destroys *a* and K_A.

Upon receiving Message 4 from *S*, *B* computes $M_B = g^{bw_B x'_B}$ and verifies whether $H_5(S,\ A,\ w_B,\ R'_A,\ M_B) = Auth'_B$ holdsor not. If not, *B* terminates the execution. Otherwise, $wB,\ \ RA',\ \ AuthB')$ is sent by *S*. *B* then computes $KB=(RA')b$ and $skB=H6A,\ \ B,\ \ RA',\ RB,\ \ KB$. Finally, *B* destroys *b*and KB.

3. Security Analysis on Yang et al.'s 3PAKA Protocol

This section provides security analyses and privacy issue analysis on Yang, *et al.*, 3PAKA protocol. The protocol is weak against offline password guessing attack with lost smartcard and does not provide authentication in the password updating phase. Furthermore, it is possible to be tracked by attacker because the protocol does not provide user anonymity.

3.1 Password Guessing Attack

One of the most important security requirements for password-based authentication protocols is to resist against password guessing attack. In Yang, *et al.*, protocol, a user is allowed to choose his/her own password during the registration and the password change phases. The user usually tends to select passwords with easy-to-remember, which has low entropy. Hence, these passwords are potentially vulnerable to password guessing attack. Yang, *et al.*, argued that their 3PAKA protocol is strong against password guessing attack. However, we will show that Yang, *et al.*, 3PAKA protocol is weak against password guessing attack with the assumption that an attacker could get a legal user's smartcard and read the memory on it as the same assumptions in the other papers [12-18].

For the off-line password guessing attack, attacker with legal user *A*'s information {*A*, y_A, y_A', h_A} on the smartcard can compute PW_A' by guessing a password candidate pw_A',$X_A$from y_A, $X_A'$from y_A' and $H_3(X_A, X_A')$,and compares it with $h_A$by following the detailed steps

Step 1 : An attacker steals a legal user *A*'s smartcard and gets the user's information {*A*, y_A, y_A', h_A} on it.

Step 2 : The attacker chooses a password candidate pw_A' from the dictionary and computes $PW_A' = H_0(A, pw_A')$, $X_A = y_A \oplus PW_A'$, $X_A' = y_A' \oplus PW_A'$, and$H_3(X_A, X_A')$, where X_A and X_A' are used to distinguish from the original x_A and x_A' computed by the server.

Step 3 : The attacker could verify the correctness of the guessed password pw_A' by checking whether the equation $H_3(X_A, X_A')$?= h_A holds or not. If it does not hold, the attacker chooses a wrong password candidate and retries Steps 2 and 3. Otherwise, the attacker finishes the guessing.

After the password guessing attack, the attacker could disguise as legal user *A*to the server successfully. Thereby, Yang, *et al.,* 3PAKA protocol is definitely weak against password guessing attack.

3.2 Lack of Authentication for Password Updating

Authentication is accepting proof of identity given by a credible person who has evidence on the said identity, or on the originator and the object under assessment as the originator's artifact respectively [20]. Thereby, any services should check authenticity of user before the system provides them. However, the password updating phase in Yang, *et al.,* 3PAKAprotocol does not perform authentication before the smartcard changes user's password. Any user with the smartcard could perform the phase and thereby it influence to the denial of service to the registered user due to the lack of authentication in the password updating phase.

3.3Lack of Privacy Preserving

Privacy, especially focused on the anonymity for the secrecy of the identities of communicating parties, is becoming a major concern in many multiuser network environments. For example, anonymity is one of the crucial focuses for ubiquitous computing and requires that the identity of any user should be protected from the outsiders. Many AKA protocols have been proposed, which implement different aspects of anonymity [13, 21-22].

As for Yang, *et al.,* 3PAKA protocol, the user's identities *A* and *B* areexposed in the authentication messages as {*A*, V_A, $Auth_A$}, {*A*,V_A,$Auth_A$, *B*, V_B, $Auth_B$} and {{*S*, *B*,w_A, R_B', $Auth_A'$}, {*S*, *A*, w_B, R_A', $Auth_B'$}} at rounds 1 to 3, respectively. Anybody could easily trace the message and get some important information by tracking the identities of each message, which knows who send the message and how many times does the user send to when and whom. Comparatively, a more ideal anonymity property for the AKA protocol is un-traceability, which meansthat the adversary can know neither who the sender is norwhether two conversations originate from the same user.

Table 2. Required 12 Security Criteria

Criteria	Required features

C1	The server needs not to maintain a security-sensitive verification table
C2	The password is memorable and can be chosen freely by the user
C3	The password cannot be derived by the privileged administrator of the server
C4	The security of the protocol is not based on the tamper resistance assumption of the smart card
C5	The protocol can resist various kinds of sophisticated attacks, such as offline password guessing attack, replay attack, parallel session attack, denial of service attack, stolen verifier attack and user/server impersonation attack
C6	The password cannot be broken by guessing attack even if the smart card is lost/stolen and compromised
C7	The client and the server can establish a common session key during the authentication process
C8	The protocol is not prone to the problems of clock synchronization and time delay
C9	The user can change the password locally without any interaction with the authentication server
C10	The protocol can achieve mutual authentication
C11	The protocol preserves privacy by providing user anonymity and un-traceability
C12	The protocol provides the property of forward secrecy

4. P_3PAKA Protocol

This section proposes anew privacy preserving 3PAKA (P_3PAKA) protocol using smart cards to solve the security problems in Yang et al.'s protocol after summarizes the required design criteria for P_3PAKA protocol. P_3PAKAprotocol adopts dynamic identifier depending on each session's nonce to provide user anonymity and un-traceability. P_3PAKA protocol is composed of 4 phases, registration, login, key agreement, and password updating.

4.1 Required Security Criteria

This sub-section summarizes12 required criteria for smart card based 3PAKA protocol in terms of security and efficiency as shown in Table 2, which is the design goal of the proposed protocol.

4.2 Registration Phase

Let s and $PU_s = g^s$ denote the server S's private key and its corresponding public key, where s is kept secret by S and PU_s is stored inside each user's smart card. Here, we take user A's registration as an example. First of all,A submits his/heridentifier A and hashed password $DPW_A = H_0(d, pw_A)$to S, where $pw_A$and $d$are the password of A and random number, respectively. If S accepts this request, it will perform the following steps

Step 1 : Computes $PW_A = H_0(A, DPW_A)$, $x_A = H_1(A, s)$, $y_A = PW_A \oplus x_A$, and $h_A = H_3(x_A, PU_s)$.

Step 2 : Stores y_A,h_A and $PU_s$to the memory of A's smart card.

After this, A should store d on his/her smart card. Note that the smart card is protected with user's identifier A and password pw_A.

4.3 Login Phase

After inserting their smart card into their card reader, A and B must input theiridentifier and password to their smart cards to use them. Then their smart cards perform the following verification process, respectively.

Step 1 :A's smart card computes $DPW_A = H_0(d,\ pw_A), PW_A = H_0(A, DPW_A), x_A = y_A \oplus PW_A$ and $h_A^{'} = H_3(x_A,\ PU_s)$. If $h_A^{'} \neq h_A$, it aborts the execution and destroys x_A.

Step 1 :B's smart card computes $DPW_B = H_0(d,\ pw_B), PW_B = H_0(A,\ DPW_B),\ x_B = y_B \oplus PW_B$ and $h_B^{'} = H_3(x_B,\ PU_s)$. If $h_B^{'} \neq h_B$,it aborts the execution and destroys x_B.

4.4. Password Updating Phase

Whenever user wants to change his/her password, he/she could perform this phase without helping of S. Smart card performs the password change only if the user authentication is successful as the same as in the first step on the login phase. The steps for the password updating is as follows

Step 1 :A inserts his/her smart card into his/her card reader. Then, A inputs his/heridentifier A, old password pw_A and new password pw_A^* to the smart card.

Step 2 : The smart card computes $DPW_A = H_0(d,\ pw_A), PW_A = H_0(A, DPW_A)$, $DPW_A^* = H_0(d,\ pw_A^*)andPW_A^* = H_0(A, DPW_A^*)$, and updates$y_A = y_A \oplus PW_A \oplus PW_A^*$.

4.5. Key Agreement Phase

Suppose A and B wish to agree on a shared session key sk_{AB}. The process of the key agreement is shown in Figure 2. There are three rounds and four messages during each run of P_3PAKAprotocol. The details are as follows

Round 1: Message 1 ($A \rightarrow B$):$\{CID_A, R_A, Auth_A\}$

A randomly chooses an integer $a \in Z_q^*$, computes $R_A = g^a,\ V_A = PU_s^a \bmod p, CID_A = A \oplus h(R_A,\ V_A)andAuth_A = H_7(A,\ R_A\ , x_A)$ and sends $\{CID_A,\ R_A,\ Auth_A\}$to B.

Round 2: Message 2 ($B \rightarrow S$): $\{CID_A, R_A,\ Auth_A, CID_B, R_B, Auth_B\}$

Upon receiving Message 1 from A, B randomly selects an integer $b \in Z_q^*$, and computes $R_B = g^b,\ V_B = PU_s^b \bmod pandCID_B = B \oplus h(R_B,\ V_B)$. B then computes $Auth_B = H_7(B,\ R_B,\ x_B)$ and sends $\{CID_A, R_A,\ Auth_A, CID_B, R_B, Auth_B\}$ to S.

Round 3: Message 3 ($S \rightarrow A$):$\{S, w, R_B', Auth_A'\}$

Message 4 ($S \rightarrow B$):$\{\ S, w, R_A', Auth_B'\}$

Upon receiving Message 2 from B, S computes $V_A' = (R_A)^s$and checks whether $Auth_A = H_7\big(A',\ R_A\ ,\ H_1(A',\ s)\big)$holds or not. If not,$S$ terminates the execution. Otherwise,S assures that the message $\{CID_A,\ R_A\ ,\ Auth_A\}$ is sent by A. S continues to compute $V_B' = (R_B)^s$ and checks whether $Auth_B = H_7\big(B',\ R_B,\ H_1(B',\ s)\big)$holds or

User *A*	User *B*	Server *S*

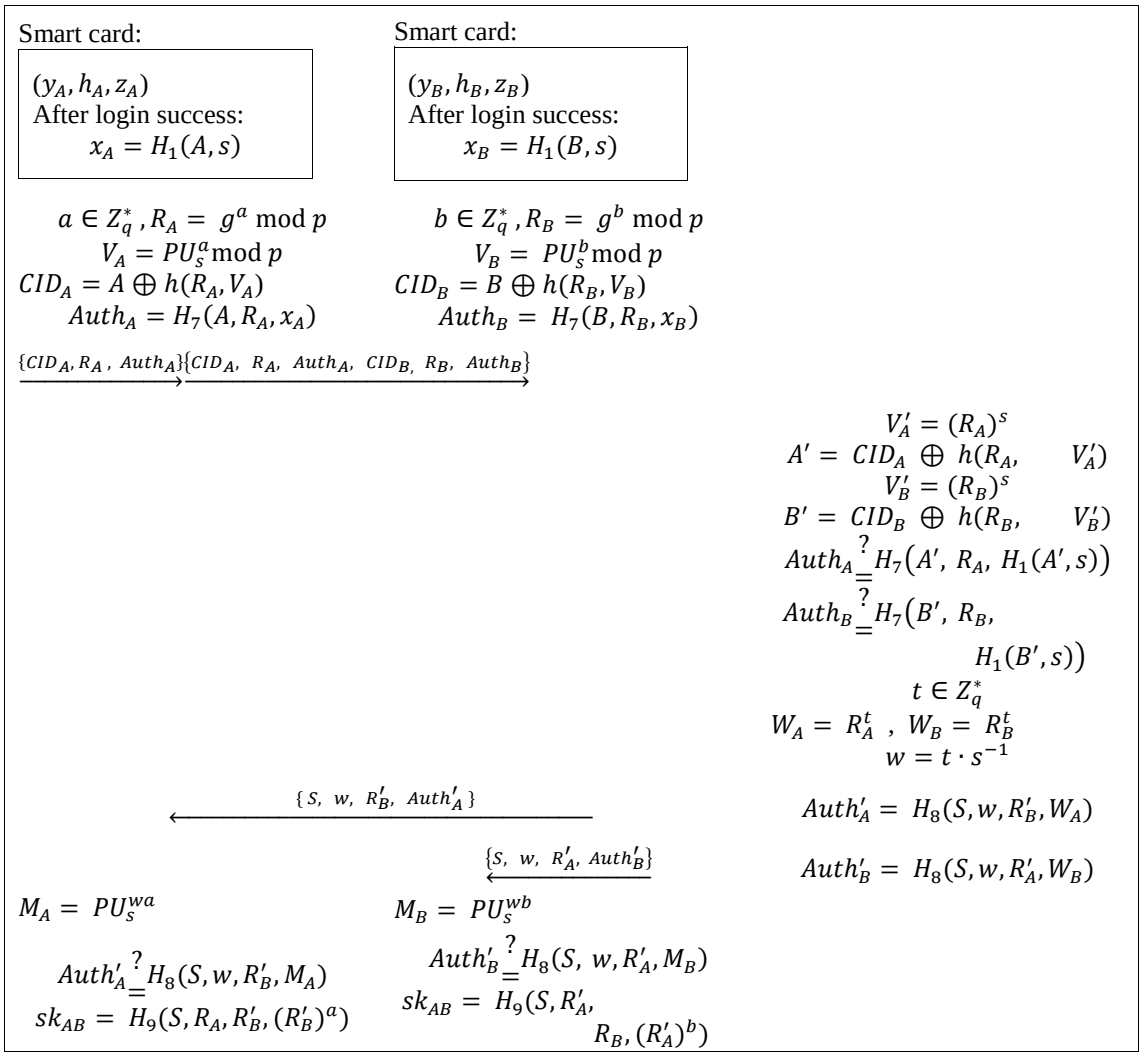

Figure 2. P_3PAKA Protocol

not. If not, *S* terminates the execution. Otherwise, *S* learned that the message $\{CID_B, R_B, Auth_B\}$ is indeed sent by *B*. Next, *S* randomly chooses an integer $t \in Z_q^*$and computes $W_A = R_A^t$and$W_B = R_B^t$.*S* further computes $w = t \cdot s^{-1}$,$Auth_A' = H_8(S, w, R_B', W_A)$, and $Auth_B' = H_8(S, w, R_A', W_B)$ .Finally, *S* destroys *t*, W_A and W_B , and sends $\{S, w, R_B', Auth_A'\}$to *A*and $\{S, w, R_A', Auth_B'\}$ to*B*, respectively.

Upon receiving Message 3 from *S*, *A* computes $M_A = PU_S^{wa}$and checks whether $Auth_A' = H_8(S, w, R_B', M_A)$holds or not. If not,*A* terminates the execution. Otherwise, *A* assures that the message $\{S, w, R_B', Auth_A'\}$is sent by *S*. Subsequently, *A* computes $sk_{AB} = H_9(S, R_A, R_B', (R_B')^a)$ and destroys *a*.Upon receiving Message 4 from *S*, *B* computes $M_B = PU_S^{wb}$and verifies whether $Auth_B' = H_8(S, w, R_A', M_B)$holds or not. If not, *B* terminates the execution. Otherwise, *B* ensures that the message $\{S, w_B, R_A', Auth_B'\}$ is sent by *S*. *B* then computes $sk_{AB} = H_9(S, R_A', R_B, (R_A')^b)$.*B* destroys *b*.

5. Security and Performance Analyses

This section provides the security analysis and the performance analysis of P_3PAKA protocol by comparing it with the related protocols in [14-15] and [17-18]. The performance analysis isfocused on the computational and communicational overhead.

5.1. Security Analysis

This sub-section provides the security analysis of P_3PAKA protocol focused on password guessing attack, replay attack, parallel session attack, password disclosure to server, stolen verifier attack, server impersonation attack and user impersonation attack, and the aspects that P_3PAKA provides privacy and forward secrecy. Table 3 shows the security comparison based on the criteria in the sub-section 4.1 among the related protocols in [14-15] and [17-18]. The reason why the protocols in [14-15] and [17-18] are selected to compare with is that these protocols have similar aspects with ours.

5.1.1. Password Guessing Attack: We could have the same assumption from the sub-Section 3.1 that an attacker could get a legal user's smart card and read the memory on it. Then only information the attacker could get are $\{y_A, h_A, PU_s, d\}$ from the memory of the smart card. Additionally, the attacker could have intercepted messages of $\{ CID_A, R_A, Auth_A \}$,$\{ CID_A, R_A, Auth_A, CID_B, R_B, Auth_B \}$, $\{ S, w, R'_B, Auth'_A \}$, and $\{ S, w, R'_A, Auth'_B \}$from the previous sessions. Even if the attacker could get the information, it is not possible to derive the password pw_A or the identifier A from them due to the one-wayness of the hash function. There is only $y_A$related to the password pw_Athat the attacker has. To find the correct password pw_A, the attacker needs to know x_A to derive PW_A from y_A and A from PW_A. However, there is no way that the attacker knows these two values at the same time. In the other aspect, the attacker could have the identifier related value CID_A. However, the attacker could not get any valuable identifier information from $CID_A = A \oplus h(R_A, V_A)$ due to the one-wayness of the hash function and the discrete logarithm problem. Thereby, it is impossible to perform password guessing attack against P_3PAKA protocol.

5.1.2. Replay Attack and Parallel Session Attack: P_3PAKA protocol is secure against replay attack because the login and key agreement messages $\{ CID_A, R_A, Auth_A \}$,$\{ CID_A, R_A, Auth_A, CID_B, R_B, Auth_B \}$, $\{ S, w, R'_B, Auth'_A \}$, and $\{ S, w, R'_A, Auth'_B \}$ are secured by using session dependent fresh random numbers*a*, *b* and *t*.

On the other hand, P_3PAKA protocol could resist against parallel session attackthat an adversary pretend as legitimate user A or B by resending a previously intercepted authentication message. The server could check the freshness of session dependent random number related value R_A or $R_B$in the authentication message. Furthermore, the attacker cannot compute session key related information due to the lack of knowledge related with a or$b$. Thereby, P_3PAKA protocol is secure against replay attack and parallel session attack.

5.1.3. Password Disclosure to Server and Stolen Verifier Attack : For the registration, a user *A*sends*A* and$H_0(d, pw_A)$instead of using plaintext password pw_A. It is computationally infeasible the server to derive pw_A from $H_0(d, pw_A)$ without knowing d due to the one-wayness of the hash function.On the other hand, P_3PAKA protocol does not need to keep verifier for the password in the server as the same as in Yang et al.'s registration phase. Thereby, P-3PAKA protocol is secure against password disclosure to server and stolen verifier attack.

5.1.4. Server ImpersonationAttack: For this attack, although an attacker can access the transmitted messages, the attacker cannot compute the correct $V'_A = (R_A)^s$ and $V'_B = (R_B)^s$from them because the attacker does not know the server's secret key *s*.In the case that

the attacker knows all information without users'identifier*A* and *B*, and server's secret key *s*, the attacker has to solve the hash function $CID_A = A \oplus h(R_A, V_A)$ and $CID_B = B \oplus h(R_B, V_B)$ as well as$Auth_A = H_7(A, R_A, x_A)$and $Auth_B = H_7(B, R_B, x_B)$.Moreover, the attackeris impossible to compute $sk_{AB} = H_9(S, R_A, R'_B, (R'_B)^a)orsk_{AB} = H_9(S, R'_A, R_B, (R'_A)^b)$because the attacker cannot obtain $H_0(d, pw_A)$. Therefore,P_3PAKA protocol is safe from server impersonation attack.

5.1.5. User Impersonation Attack: The transmitted messages $\{CID_A, R_A, Auth_A\}$, $\{CID_A, R_A, Auth_A, CID_B, R_B, Auth_B\}$, $\{S, w, R'_B, Auth'_A\}$, and $\{S, w, R'_A, Auth'_B\}$ are all protected by secure one-way hash function and discrete logarithm problem. When certain modifications are performed by an attacker to these parameters of legitimated authentication messages from legal user *A*or*B*, the modifications could be detected by *S*. Although the attacker can change the values of$CID_A, R_A, Auth_A, CID_B, R_B$and $Auth_B$, the attacker does not know any information about *A*, *B*, PW_A and V_A due to the one-wayness of the hash functions and the discrete logarithm problem. Therefore, the attacker cannot fabricate the valid $CID_A, R_A, Auth_A, CID_B, R_BandAuth_B$. Hence, P_3PAKA protocol is secure against user impersonation attack.

5.1.6. User Anonymity: Suppose that an attacker could intercept *A*'s authentication messages $\{CID_A, R_A, Auth_A\}$ and $\{S, w, R'_B, Auth'_A\}$.Then the attacker tries to get certain parameters from these messages, but these messages are treated to be random strings due to the randomness of *a*, *b* and *t.* Therefore, in case of the attacker does not know about these *a* and *b*, the attacker will face to solve the discrete logarithm problem to get the correct values of *A* or *B* from CID_A and CID_B. Hence,P_3PAKA protocol can preserve user anonymity.

5.1.7. Forward Secrecy: In P_3PAKA protocol, user *A* and user *B*can establish the same session key $sk_{AB} = H_9(S, g^a, g^b, g^{ab})$based on the session dependent g^{ab}. It is impossible to compute an attacker the session key sk_{AB} in P_3PAKA protocol due to difficulty of the computational Diffie-Hellman problem even if the attacker knows the server's long term secret *s*. Furthermore, the server even could not compute the session key due to the same difficulty problem to the attacker. As a result, P_3PAKA protocol provides the forward secrecy.

Table 3. Criteria Comparison among Relevant Authentication Protocols

Criteria / Protocol	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Jaung in [14]	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	No	No
Chang et al. in [15]	Yes	Yes	No	Yes	No	No	Yes	No	Yes	Yes	No	No
Yoon et al. in [17]	Yes	Yes	No	Yes	No	No	Yes	No	Yes	Yes	No	No
Yang et al. in [18]	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	No	Yes
P_3PAKA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

5.2. Performance Analysis

To evaluate the proposed P_3PAKA protocol, we compare the communication overhead

and the computation overhead by comparing with the related protocols in [14-15] and [17-18]. Since the login phase and key agreement phase are executed more frequently than the other two phases, only the computation cost and communication overhead during them are taken into consideration.

P_3PAKA protocol only needs three rounds of message communication while the protocols in [14- 15] need five rounds of messages. This means that P_3PAKA protocol can enable shorter communication latency and more rapid response than the protocols in [14-15]. Protocols in [17-18] require the same communication overhead but they have lack of securities as shown in Table 3.

For the computation overhead, we do not take XOR operation into account because the time complexity of $\oplus$ is negligible as compared to the other three operations. Typically, time complexity associated with these operations can be roughly expressed as time for modular exponentiation > time for encryption > time for hash. P_3PAKA protocol has a bit more computational overhead compared to the protocols in [14, 15, 17] but similar with the protocol in [18] as shown in Table 4. The overhead is for the costs from the additional required properties in the criteria at the sub-section 4.1.

Table 4. Performance Comparison between Related Protocols

Property \ Protocol		Protocol in [14]	Protocol in [15]	Protocol in [17]	Protocol in [18]	P_3PAKA
Communication overhead	Number of rounds	5	5	3	3	3
Computation overhead	Encryption/Decryption operation (*A*/*B*/*S*)	4/4/2	4/4/4	2/2/2	0/0/0	0/0/0
	Hash/MAC operation (*A*/*B*/*S*)	2/2/4	2/2/2	3/2/3	5/5/8	7/7/6
	Modular exponentiation (*A*/*B*/*S*)	0/0/0	0/0/0	0/0/0	4/4/4	4/4/4

6. Conclusion

Because AKA protocols have great practical usages in many network environments, especially in financial secure applications, they have been widely deployed. This paper has reviewed Yang, *et al.*, provably secure 3PAKA protocol using smart cards and shown that the protocol is weak against offline password guessing attack with lost smartcard and does not provide authentication in the password updating phase. Furthermore, it is possible to be tracked by attacker because Yang, *et al.*, 3PAKA protocol does not provide user anonymity. In order to solve the weaknesses in Yang, *et al.*, 3PAKA protocol, this paper proposed a privacy preserving 3PAKA (P_3PAKA) protocol using smart cards. P_3PAKAprotocol provides user anonymity and un-traceability by using dynamic identifier depending on each session's nonce.As shown in the analyses, P_3PAKA protocol is more secure while maintaining efficiency than the other previous protocols.

References

[1] S. K. H. Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography", Mathematical and Computer Modeling, vol. 57, **(2013),** pp. 2703–2717.
[2] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, vol. 24, **(1987)**, pp. 770-772.

[3] C. K. Chan and L. M. Cheng, "Cryptanalysis of a remote user authentication scheme using smart cards", IEEE Transactions on Consumer Electronics, vol. 46, **(2000),** pp. 992–993.
[4] C. I. Fan, Y. C. Chan and Z. K. Zhang, "Robust remote authentication scheme with smart cards", Computer and Security, vol. 24, **(2005),** pp. 619–628.
[5] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart cards", IEEE Transactions on Consumer Electronics, vol. 46, **(2000)**, pp. 28–30.
[6] C. T. Li and M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards", Journal of Network and Computer Applications, vol. 33, **(2010)**, pp. 1–5.
[7] H. J. Park and C. Kim, "Enhanced smartcard based multi-server authentication scheme", International Journal of Security and Its Applications, vol. 7, **(2013),** pp. 155-164.
[8] W. Jeon, Y. Lee and D. Won, "An efficient user authentication scheme with smart cards for wireless communications", International Journal of Security and Its Applications, vol. 7, **(2013)**, pp. 1-15.
[9] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22, **(1976)**, pp. 644-654.
[10] W. Diffie, M. Wiener and P. V. Oorschot, "Authentication and authenticated key exchanges", Designs, Codes and Cryptography, vol. 2, **(1992)**, pp. 107-125.
[11] J. Katz, R. Ostrovsky and M. Yung, "Efficient password-authenticated key exchange using human memorable passwords", Lecture Notes in Computer Science, vol. 2045, **(2001),** pp. 475-494.
[12] Y. Yang, R.H. Deng and F. Bao, "A practical password-based two-server authentication and keyexchange system", IEEE Transactions on Dependable and Secure Computing, vol. 3, **(2006),** pp. 105-114.
[13] X. Li, W. Qiu, D. Zheng, K. Chen and J. Li, "Anonymity enhancement on robust and efficientpassword-authenticated key agreement using smart cards", IEEE Transactions on Industrial Electronics, vol. 57, **(2010)**, pp. 793-800.
[14] W. S. Juang, "Efficient three-party key exchange using smart cards", IEEE Transactions on Consumer Electronics, vol. 50, **(2004)**, pp. 619-624.
[15] C. Chang, J. Lee and T. Cheng, "Security design for three-party encrypted key exchange protocol using smart cards", Proc. of the 2nd International Conference on Ubiquitous Information Management and Communication, **(2008)**, pp. 329-333.
[16] J. O. Kwon, I. R. Jeong and D. H. Lee, "Three-round smart card-based key exchange scheme", IEICE Transactions on Communications, vol. E90-B, **(2007)**, pp. 3255-3258.
[17] E. J. Yoon and K. Y. Yoo, "Enhanced three-round smart card-based key exchange protocol", Lecture Notes on Computer Science, vol. 5060, **(2008),** pp. 507-515.
[18] H. Yang, Y. Zhang, Y. Zhou, X. Fu, H. Liu and A. V. Vasilakos, "Provably secure three-party authenticated key agreement protocol using smart cards", vol. 58, **(2014),** pp. 29-38.
[19] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols", Proc. of the 1st ACM Conference on Computer and Communications, Security, **(1993)**, pp. 62-73.
[20] http://en.wikipedia.org/wiki/Authentication.
[21] X. Li, Y. Zhang, X. Liu and J. Cao, "A lightweight three-party privacy-preserving authentication key exchange protocol using smart card", KSII Transactions on Internet and Information Systems, vol. 7, **(2013),** pp. 1313-1327.
[22] K. Kim and M. H. Kim, "An enhanced anonymous authentication and key exchange scheme using smartcard", Lecture Notes in Computer Science, vol. 7839, **(2013)**, pp. 487-494.

Authors

Suyeon Park, she is a student at the School of Computer IT Engineering, Daegu University, and Republic of Korea from 2011.Her research interests include information security, computer network, computer algorithm, machine to machine communication and cloud computing.

Hee-Joo Park, he is a professor at the Department of Cyber Security, Kyungil University, and Republic of Korea from 2012. He received the B.S. and M.S. degreesin Electrical Engineering from Yeungnam University, Republic of Korea, in 1978 and 1981, respectively. Hereceived thePh.D. degree in Computer Science and Statistics from Catholic University of Daegu, Republic of Korea, in 1995. He had been a professor from 1982 to 2012 with the Department of Computer Engineering, Kyungil University. His research interests include information security, neural network, pattern recognition, ad-hoc network and sensor network.