International Journal of Innovative Computing, Information and Control
Volume **8**, Number **8**, August **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **5657–5675**

IMPROVING THE LEE-LEE'S PASSWORD BASED AUTHENTICATED KEY AGREEMENT PROTOCOL

Eun-Jun Yoon[1] and Kee-Young Yoo[2,*]

[1]Department of Cyber Security
Kyungil University
33 Buho-Ri, Hayang-Ub, Kyungsan-Si, Kyungsangpuk-Do 712-701, Republic of Korea
ejyoon@kiu.ac.kr

[2]Department of Computer Engineering
Kyungpook National University
1370 Sankyuk-dong, Buk-gu Daegu 702-701, Republic of Korea
*Corresponding author: yook@knu.ac.kr

Received December 2010; revised April 2011

Abstract. *Password based authenticated key agreement protocols have been the most widely used methods for user authentication, since it allows people to choose and remember their own passwords without any assistant device. Password based authenticated key agreement protocols, however, are vulnerable to password guessing attacks since users usually choose easy-to-remember passwords. Recently, Lee and Lee pointed out that N. Y. Lee et al.'s password based authenticated key agreement protocol is vulnerable to a man-in-the-middle attack, and then proposed an improvement to overcome the attack. The current paper, however, demonstrates that Lee-Lee's password based authenticated key agreement protocol is still vulnerable to off-line password guessing attacks, and then proposes an improvement of the protocol in order to overcome such security attacks. Compared with Lee-Lee's protocol, the proposed protocol is very useful in password-based Internet and wire/wireless communication environments to access remote information systems since it provides security, reliability and efficiency.*

Keywords: Cryptography, Authenticated key agreement, Cryptanalysis, Password guessing attacks

1. **Introduction.** Password based authenticated key agreement protocols [1] have been the most widely used method for user authentication, since it allows people to choose and remember their own passwords without any assistant device. The objective of a password based authenticated key agreement protocol is the same as a conventional authenticated key agreement protocol: after two communicating parties successfully execute the protocol, each of them should have certain assurances of the other's true identity (authentication), and it shares a new and random session key only with each other and the key is derived from contributions of both parties (key agreement) [2-10].

Password based authenticated key agreement protocols, however, are vulnerable to password guessing attacks [11] since users usually choose easy-to-remember passwords. Unlike typical private keys, the password has limited entropy, and is constrained by the memory of the user. For example, one alphanumerical character has 6 bits of entropy. Therefore, the goal of the attacker, which is to obtain a legitimate communication parties' password, can be achieved within a reasonable time. Thus, the password guessing attacks on password based authenticated key agreement protocols should be considered realistic [12-17].

In 1999, Seo and Sweeney [18] proposed a new key agreement protocol based on the Diffie-Hellman protocol [19] called simple authenticated key agreement algorithm (SAKA). In SAKA protocol, two parties have a pre-shared password for data communication, produce a session key by exchanging messages and confirm each other. Because of the advantages that can simplify key agreement, the SAKA-like protocols are widely concerned in the researches for key agreement, and therefore, there are several articles proposed to continuously enhance SAKA-like protocols [20,21].

In 2000, Tseng [20] pointed out that the SAKA protocol is vulnerable to the replay attack and then proposed an improved protocol. Later, Ku and Wang [21] showed that the Tseng's protocol is still vulnerable to the backward replay attack without modification and the modification attack. To prevent these attacks, Ku and Wang also proposed another improvement. However, Hsu et al. [22] pointed out that the Ku-Wang's protocol is also vulnerable to the modification attack and then proposed an improved protocol to prevent this security flaw. Unfortunately, in 2004, N. Y. Lee et al. [23] showed that the Hsu et al.'s protocol cannot withstand another modification attack and then proposed its enhanced protocol.

Thereafter, in 2005, Lee and Lee [26] showed that N. Y. Lee et al.'s password based authenticated key agreement protocol is still vulnerable to the man-in-the-middle attack. That is, an active adversary can fool the participants into believing a wrong session key by altering the exchanged messages in the key establishment and validation phase. Then, Lee and Lee proposed a slight improvement to overcome the security flaw. Nevertheless, Lee-Lee's improved protocol is still susceptible to off-line password guessing attacks since users usually choose easy-to-remember passwords [11]. Unlike typical private keys, the password has limited entropy, and is constrained by the memory of the user. For example, one alphanumerical character has 6 bits of entropy, and thus the goal of the attacker, which is to obtain a legitimate communication party's password, can be achieved within a reasonable time. Therefore, the password guessing attacks on Lee-Lee's improved protocol should be considered a real possibility. In general, the password guessing attacks can be divided into three classes [11]:

- Detectable on-line password guessing attacks: an attacker attempts to use a guessed password in an on-line transaction. He/she verifies the correctness of his/her guess using the response from corresponding party (e.g., server). A failed guess can be detected and logged by the corresponding party (e.g., server).
- Undetectable on-line password guessing attacks: similar to above, an attacker tries to verify a password guess in an online transaction. However, a failed guess cannot be detected and logged by the corresponding party (e.g., server), as the corresponding party (e.g., server) cannot distinguish between an honest request and an attacker's request.
- Off-line password guessing attacks: an attacker guesses a password and verifies his/her guess off-line. No participation of corresponding party (e.g., server) is required, so the corresponding party (e.g., server) does not notice the attack as a malicious one.

Accordingly, this paper demonstrates the vulnerabilities of Lee-Lee's protocol to the off-line password guessing attacks, and then proposes an improvement to the protocol in order to overcome such security problems. The proposed protocol has several important features and advantages as follows: (1) It is designed to optimize the computation cost of each participant by using the small communication round. (2) It achieves cryptographic goals only using discrete logarithm problem (DLP), bit-wise exclusive-OR (XOR) operation and collision-free one-way hash functions as main cryptographic operations without

additional requirements such as using public key cryptosystem and digital signatures. (3) It not only is secure against well-known cryptographical attacks such as replay attack, guessing attack, man-in-middle attack, modification attack and impersonation attack, but also provides secure mutual authentication, known-key security, session key security and perfect forward secrecy. Thus, the proposed protocol is very useful in password-based Internet and wire/wireless communication environments to access remote information systems since it provides security, reliability and efficiency.

The remainder of this paper is organized as follows. Section 2 briefly reviews Lee-Lee's password based authenticated key agreement protocol. An outline of the off-line password guessing attacks on Lee-Lee's protocol is proposed in Section 3. Our improved protocol is presented in Section 4, while Section 5 and Section 6 discuss the security and the efficiency of the proposed protocol, respectively. Our conclusions are presented in Section 7.

2. **Review of Lee-Lee's Protocol.**

This section briefly reviews Lee-Lee's password based authenticated key agreement protocol [26].

2.1. **Security requirements.**

Here, eight security properties [32, 33]: replay attack, password guessing attack, man-in-middle attack, modification attack, mutual authentication, known-key security, session key security and perfect forward secrecy, must be considered for the password-based authentication protocol.

1. *Replay attack*: A replay attack is an offensive action in which an adversary impersonates or deceives another legitimate participant through the reuse of information obtained in a protocol.
2. *Guessing attack*: A guessing attack involves an adversary (randomly or systematically) trying long-term private keys (e.g., user password), one at a time, in the hope of finding the correct private key. Ensuring long-term private keys chosen from a sufficiently large space can reduce exhaustive searches. Most users, however, select passwords from a small subset of the full password space. Such weak passwords with low entropy are easily guessed by using the so-called dictionary attack.
3. *Man-in-middle attack*: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection when in fact the entire conversation is controlled by the attacker.
4. *Modification attack*: A modification attack is an attempt to modify information that an attacker is not authorized to modify. This type of attack is an attack against the integrity of the information.
5. *Mutual authentication*: Mutual authentication means that both the communication entities (the client and the server) are authenticated to each other within the same protocol.
6. *Known-key security*: Known-key security means that each run of an authentication and key agreement scheme between two communication entities (the client and the server) should produce unique secret keys; such keys are called session keys.
7. *Session key security*: Session key security means that at the end of the key exchange, the session key is not known by anyone but two communication entities (the client and the server).
8. *Perfect forward secrecy*: Perfect forward secrecy means that if a long-term private key (e.g., client password) is compromised, this does not compromise any earlier session keys.

2.2. **Notations.** Notations used in this paper are defined as follows:

- Alice, Bob: two communicating parties;
- ID_A, ID_B: two communicating parties' identities;
- Eve: an adversary or attacker;
- p: a large prime number;
- g: a generator $\in {}_RZ_p^*$ with the order $p-1$;
- D: a uniformly distributed dictionary of size $|D|$;
- PW: a low-entropy password shared between Alice and Bob, which is randomly chosen from D;
- Q, Q^{-1}: two integers computed from PW by a pubic predesignated method such that $Q \cdot Q^{-1} = 1 \bmod p-1$;
- a: a secret random integer $\in {}_RZ_p^*$ chosen by Alice;
- b: a secret random integer $\in {}_RZ_p^*$ chosen by Bob;
- sid: a session identifier;
- SK: a shared common session key between Alice and Bob;
- $h(\cdot)$: a secure one-way hash function;
- $\oplus$: a bit-wise exclusive-or (XOR) operation.

2.3. **Lee-Lee's protocol.** Assume that the system has two public parameters p and g, where p is a large prime and g is a generator with order $p-1$ in $GF(p)$. Let Alice and Bob, two communicating parties, share a common password PW in advance. Alice and Bob can pre-compute two integers Q and $Q^{-1} \bmod (p-1)$ from the common password PW in any predetermined way and are relatively prime to $p-1$. There are two phases in the Lee-Lee's protocol: key establishment and key validation. Figure 1 illustrates Lee-Lee's password based authenticated key agreement protocol. Their protocol proceeds as follows:

2.3.1. *Key establishment phase.* Let a and b be random integers chosen by Alice and Bob, respectively. The key establishment phase of the Lee-Lee's protocol goes as follows:

KE1. Alice computes $X = aQ$ and $A_1 = g^X \bmod p$, and then sends A_1 to Bob.
KE2. Bob computes $Y = bQ$ and $B_1 = g^Y \bmod p$, and then sends B_1 to Alice.
KE3. Alice computes the session key $K_A = B_1^{Q^{-1}a} \bmod p = g^{ab} \bmod p$.
KE4. Bob computes the session key $K_B = A_1^{Q^{-1}b} \bmod p = g^{ab} \bmod p$.

2.3.2. *Key validation phase.* Let ID_A and ID_B be an identifier of Alice and Bob, respectively. Let $h(\cdot)$ be a one-way hash function [33]. The key validation phase of the Lee-Lee's protocol goes as follows:

KV1. Alice checks whether $K_A \bmod p \neq 1$ holds or not. If it holds, Alice computes $A_2 = h(ID_A, A_1, K_A)$ and sends it to Bob.
KV2. Bob checks whether $K_B \bmod p \neq 1$ holds or not. If it holds, Bob computes $B_2 = h(ID_B, B_1, K_B)$ and sends it to Alice.
KV3. Alice validates K_A by checking if $B_2 = h(ID_B, B_1, K_B)$ sent by Bob is equal to her own $h(ID_B, B_1, K_A)$.
KV4. Bob validates K_B by checking if $A_2 = h(ID_A, A_1, K_A)$ sent by Alice is equal to his own $h(ID_A, A_1, K_B)$.

Now, Alice and Bob are convinced the common secret session key $K_A = K_B = g^{ab} \bmod p$.

3. **Off-Line Password Guessing Attacks on Lee-Lee's Protocol.** This section shows that Lee-Lee's password based authenticated key agreement protocol is vulnerable to off-line password guessing attacks [11].

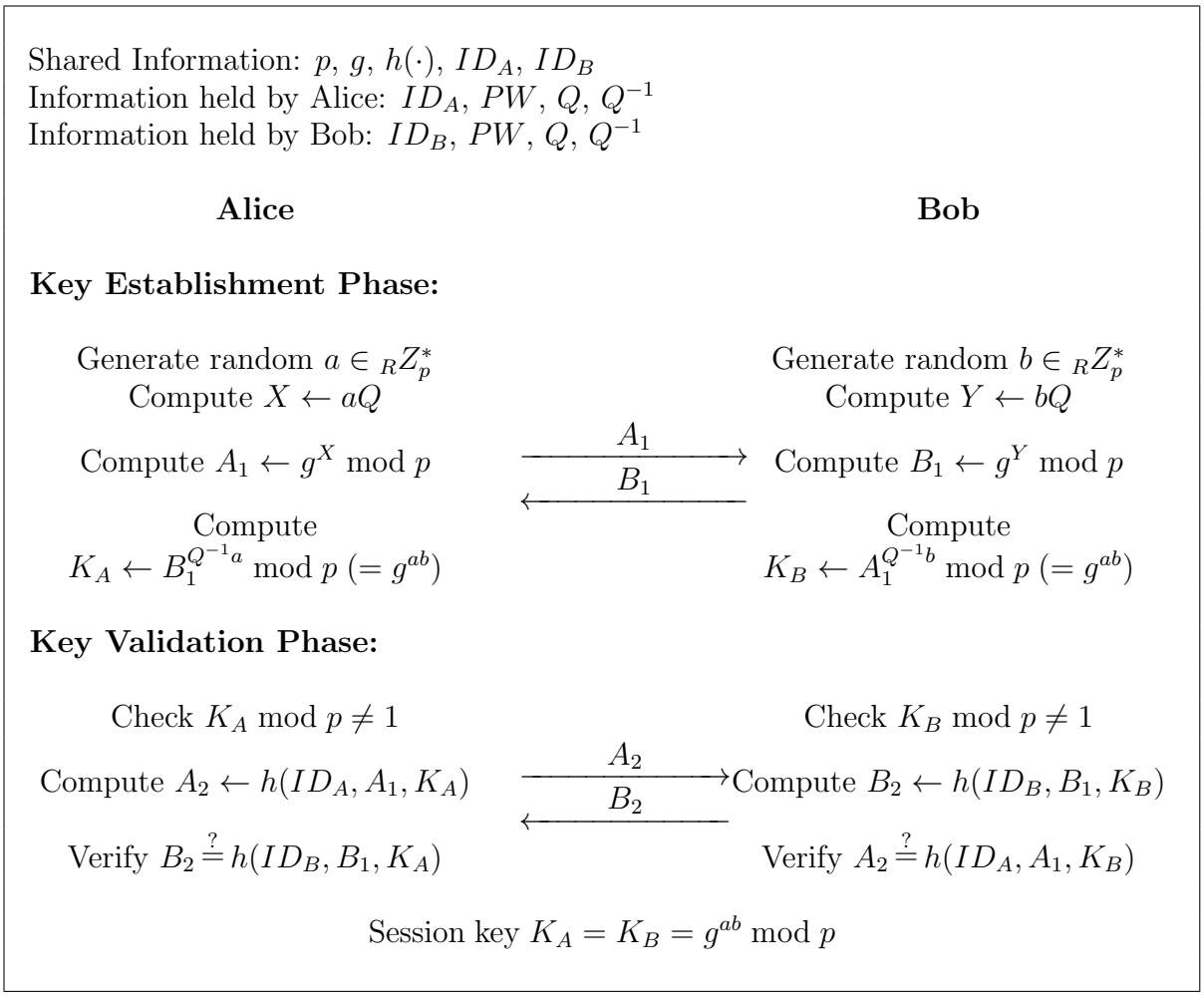

FIGURE 1. Lee-Lee's password based authenticated key agreement protocol

3.1. **Off-line password guessing attack 1.** Let Eve be an active adversary who interposes the communication between Alice and Bob. Then, Eve can easily obtain a legitimate communication parties' password PW. The off-line password guessing attack 1 proceed as follows:

3.1.1. *Attack on key establishment phase.*

KE1*. When Alice sends $A_1 = g^X \bmod p$ to Bob, where $X = aQ$, Eve replaces it with $A_1' = g \bmod p$, and then sends A_1' to Bob.

KE2*. Upon receiving A_1', Bob will compute $Y = bQ$ and $B_1 = g^Y \bmod p$, and then send B_1 to Alice.

KE3*. Eve intercepts B_1 and replaces it with $B_1' = g \bmod p$, and then sends B_1' to Alice.

Then, Alice will compute the session key $K_A = (B_1')^{Q^{-1}a} \bmod p = g^{Q^{-1}a} \bmod p$. Bob also will compute the session key $K_B = (A_1')^{Q^{-1}b} \bmod p = g^{Q^{-1}b} \bmod p$.

3.1.2. *Attack on key validation phase.*

KV1*. Alice will check whether $K_A \neq 1$ holds or not. Since it holds, Alice will compute $A_2 = h(ID_A, A_1, K_A)$ and send it to Bob.

KV2*. Eve intercepts $A_2 = h(ID_A, A_1, K_A)$.

KV3*. Bob also will check whether $K_B \neq 1$ holds or not. Since it holds, Bob will compute $B_2 = h(ID_B, B_1, K_B)$ and sends it to Alice.

KV4*. Eve intercepts $B_2 = h(ID_B, B_1, K_B)$.

3.1.3. *Off-line password guessing attack 1.*

P1*. In order to obtain the password PW shared between Alice and Bob, Eve makes a guess at the secret password PW^* from dictionary D and derives corresponding Q^* and Q^{*-1}, where $Q^* \cdot Q^{*-1} = 1 \bmod (p-1)$.

P2*. By using the intercepted value $A_2 = h(ID_A, A_1, K_A)$, Eve checks if $A_2 \stackrel{?}{=} h(ID_A, A_1, (A_1')^{(Q^{*-1})^2} \bmod p)$, where ID_A and A_1 are the information that Eve captured. If it holds, Eve has guessed the correct secret password $PW^* = PW$.

P3*. If it is not correct, Eve repeatedly performs the Steps (P1*) and (P2*) until $A_2 \stackrel{?}{=} h(ID_A, A_1, (A_1')^{(Q^{*-1})^2} \bmod p)$.

Furthermore, if Eve uses $B_2 = h(ID_B, B_1, K_B)$, ID_B and B_1, then Eve can also obtain the password PW shared between Alice and Bob as described above.

The algorithm of the off-line password guessing attack 1 is as follows:

Off-line Password Guessing Attack 1 $(ID_A, A_1, A_1', A_2, D)$
{
 for $i := 0$ to $|D|$
 {
  $PW^* \leftarrow D$;
  $Q^*, Q^{*-1} \leftarrow PW^*$;
  if $A_2 = h(ID_A, A_1, (A_1')^{(Q^{*-1})^2} \bmod p)$ **then return** PW^*
 }
}

3.2. **Off-line password guessing attack 2.** If Alice and Bob convince whether $A_1' = g \bmod p$ and $B_1' = g \bmod p$ in (KE2*) and (KE3*) of the key establishment phase, respectively, above mentioned off-line password guessing attack 1 will fail. However, Eve can still succeed the off-line password guessing attack 2 by using the following method:

3.2.1. *Attack on key establishment phase.*

KE1′. When Alice sends A_1 to Bob, Eve chooses an integer $a' \in {}_RZ_P^*$, replaces A_1 with $A_1' = g^{a'} \bmod p$, and then sends A_1' to Bob.

KE2′. Upon receiving A_1', Bob will compute $Y = bQ$ and $B_1 = g^Y \bmod p$, and then send B_1 to Alice.

KE3′. Eve intercepts B_1, chooses an integer $b' \in {}_RZ_P^*$, replaces B_1 with $B_1' = g^{b'} \bmod p$, and then sends B_1' to Alice.

Then, Alice will compute the session key $K_A = (B_1')^{Q^{-1}a} \bmod p = g^{b'Q^{-1}a} \bmod p$. Bob also will compute the session key $K_B = (A_1')^{Q^{-1}b} \bmod p = g^{a'Q^{-1}b} \bmod p$.

3.2.2. *Attack on key validation phase.*

KV1′. Alice will check whether $K_A \neq 1$ holds or not. Since it holds, Alice will compute $A_2 = h(ID_A, A_1, K_A)$ and send it to Bob.

KV2′. Eve intercepts $A_2 = h(ID_A, A_1, K_A)$.

KV3′. Bob also will check whether $K_B \neq 1$ holds or not. Since it holds, Bob will compute $B_2 = h(ID_B, B_1, K_B)$ and sends it to Alice.

KV4′. Eve intercepts $B_2 = h(ID_B, B_1, K_B)$.

3.2.3. *Off-line password guessing attack 2.*

P1′. In order to obtain the password PW shared between Alice and Bob, Eve makes a guess at the secret password PW^* from dictionary D and derives corresponding Q^* and Q^{*-1}, where $Q^* \cdot Q^{*-1} = 1 \bmod (p-1)$.

P2′. By using the intercepted value $A_2 = h(ID_A, A_1, K_A)$, Eve checks if $A_2 \stackrel{?}{=} h(ID_A, A_1, (A_1')^{(Q^{*-1})^2 b'} \bmod p)$, where ID_A and A_1 are the information that Eve captured. If it holds, Eve has guessed the correct secret password $PW^* = PW$.

P3′. If it is not correct, Eve repeatedly performs the Steps (P1′) and (P2′) until $A_2 \stackrel{?}{=} h(ID_A, A_1, (A_1')^{(Q^{*-1})^2 a'} \bmod p)$.

Furthermore, if Eve uses $B_2 = h(ID_B, B_1, K_B)$, ID_B and B_1, then Eve can also obtain the password PW shared between Alice and Bob as described above.

The algorithm of the off-line password guessing attack 2 is as follows:

Off-line Password Guessing Attack 2 $(ID_A, A_1, A_1', b', A_2)$
{
 for $i := 0$ to $|D|$
 {
  $PW^* \leftarrow D$;
  $Q^*, Q^{*-1} \leftarrow PW^*$;
  if $A_2 = h(ID_A, A_1, (A_1')^{(Q^{*-1})^2 b'} \bmod p)$ **then return** PW^*

 }
}

Therefore, the off-line password guessing attacks are effective to the Lee-Lee's password based authenticated key agreement protocol.

3.3. **Real applications for the proposed password guessing attacks.** In the modern life which the Internet has strong influence to people, passwords are the most common means of user authentication on the Internet. For practical applications, password-based authentication protocols are required when making use of Internet network services like E-learning, on-line polls, on-line ticket-order systems, roll call systems and on-line games. Suppose that the password PW of user can be revealed by the attacker due to the above described password guessing attacks. In real applications, users offer the same password as above to access several application servers for their convenience. Thus, an attacker may try to use the guessed password PW to impersonate the user to login to other systems that the user has registered with outside this Lee-Lee's protocol-based server. If the targeted outside server adopts the normal authentication protocol, it is possible that the attacker can successfully impersonate the user to login to it by using the guessed password PW. Therefore, the password breach cannot be revealed by the attacker's actions.

In Lee-Lee's protocol, the attacker can successfully recover the target password PW within a reasonable time on the typical pentium computer in a case where the password types are lower/upper case letters and purely random combinations of alphabet/numeric characters [34, 35]. In case where the password types are purely random combinations of alphabet/numeric/special characters, the attacker can also recover the target password PW within a reasonable time except when the password length is 10. Moreover, the attacker can successfully recover the target password PW in a reasonable time when on the powerful supercomputer, and this includes all password types [34, 35]. As a result, we can see that the proposed password guessing attacks are feasible. For this reason, the Lee-Lee's protocol is insecure for practical applications.

4. **Proposed Protocol.** This section proposes an improved password based authenticated key agreement protocol that, unlike Lee-Lee's protocol, can withstand the off-line password guessing attacks. Figure 2 illustrates the proposed password based authenticated key agreement protocol.

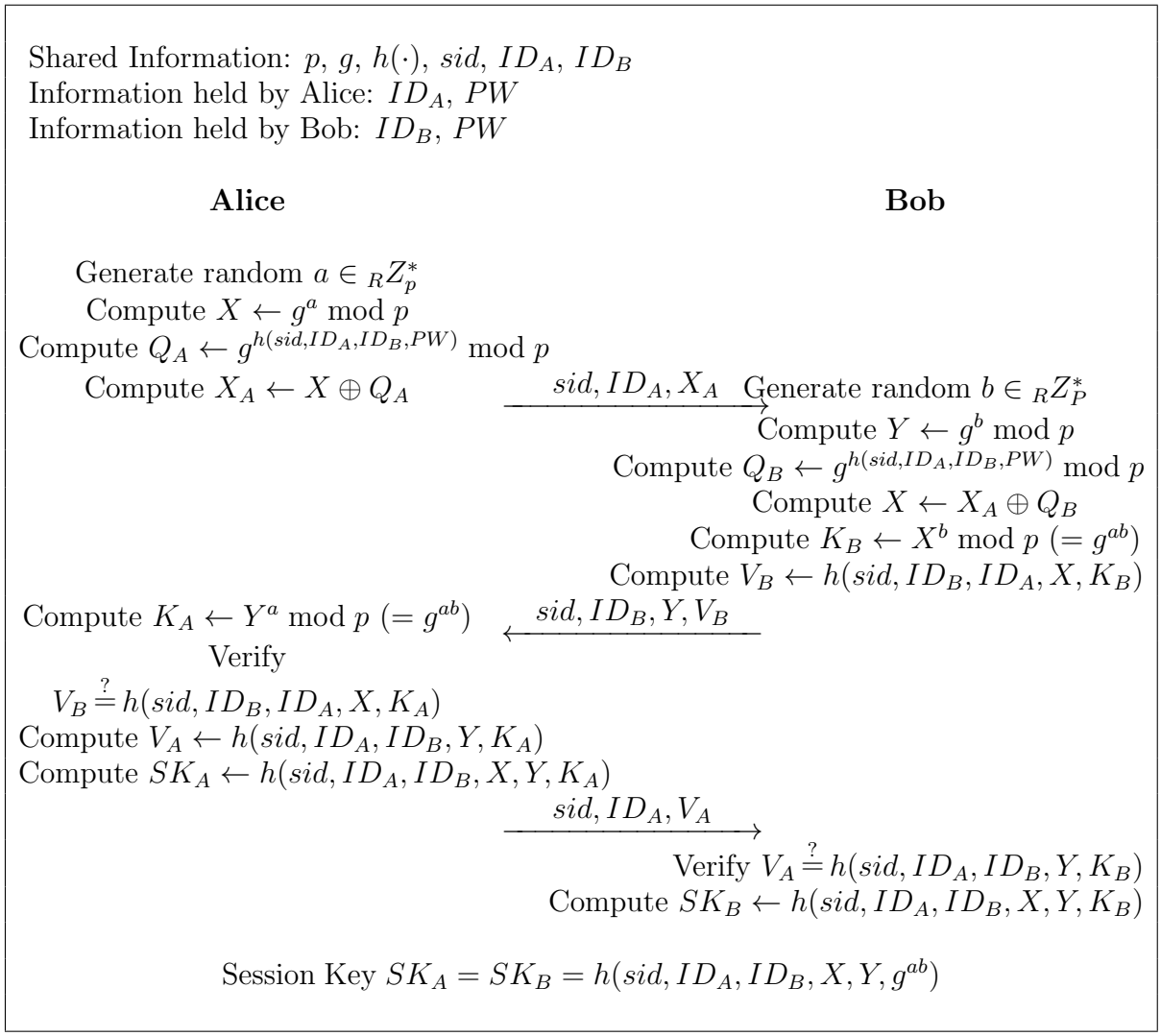

FIGURE 2. Proposed password based authenticated key agreement protocol

4.1. **Protocol.** The proposed password based authenticated key agreement protocol proceeds with the following 4 steps:

KA1. Alice $\rightarrow$ Bob: $\{sid, ID_A, X_A\}$

Alice chooses a random integer number $a \in {}_RZ_p^*$, and then computes $X = g^a \bmod p$ and $Q_A = g^{h(sid,ID_A,ID_B,PW)} \bmod p$, where sid is a session identifier. Finally, Alice computes $X_A = X \oplus Q_A$, and sends it with her identity ID_A and session identifier sid to Bob.

KA2. Bob $\rightarrow$ Alice: $\{sid, ID_B, Y, V_B\}$

Upon receiving sid, ID_A and X_A from Alice, Bob chooses a random integer number $b \in {}_RZ_p^*$, and then computes $Y = g^b \bmod p$ and $Q_B = g^{h(sid,ID_A,ID_B,PW)} \bmod p$. Bob extracts $X = g^a \bmod p$ by computing $X = X_A \oplus Q_B$. Then, Bob computes the shared secret key $K_B = X^b = g^{ab} \bmod p$ and a message authentication code

(MAC) $V_B = h(sid, ID_B, ID_A, X, K_B)$, and sends back Y and V_B with his identity ID_B and session identifier sid to Alice.

KA3. Alice $\rightarrow$ Bob: $\{sid, ID_A, V_A\}$

Upon receiving sid, ID_B, Y and V_B from Bob, Alice computes the shared secret key $K_A = Y^a = g^{ab} \bmod p$ and MAC value $h(sid, ID_B, ID_A, X, K_A)$, and verifies if the received MAC value V_B is equal to her computed MAC value $h(sid, ID_B, ID_A, X, K_A)$. If they match each other, Alice authenticates Bob. Finally, Alice computes a message authentication code (MAC) $V_A = h(sid, ID_A, ID_B, Y, K_A)$ and sends it with her identity ID_A and session identifier sid to Bob.

KA4. Bob verifies the received $\{sid, ID_A, V_A\}$

Upon receiving sid, ID_A and V_A from Alice, Bob computes MAC value $h(sid, ID_A, ID_B, Y, K_B)$, and verifies if the received MAC value V_A is equal to his computed MAC value $h(sid, ID_A, ID_B, Y, K_B)$. If they match each other, Bob also authenticates Alice.

After mutual authentication and session key agreement between Alice and Bob, $SK_A = h(sid, ID_A, ID_B, X, Y, K_A)$ and $SK_B = h(sid, ID_A, ID_B, X, Y, K_B)$ are used as a session key, where $K_A = K_B = g^{ab} \bmod p$, respectively.

4.2. **Efficiency considerations.** For providing computational efficiency in the proposed protocol, we can change $Q_A = g^{h(sid, ID_A, ID_B, PW)} \bmod p$ with $h(sid, ID_A, ID_B, PW)$ in Step KA1 and $Q_B = g^{h(sid, ID_A, ID_B, PW)} \bmod p$ with $h(sid, ID_A, ID_B, PW)$ in Step KA2, respectively. It means that a total of two-time exponential operations can be reduced in the proposed protocol. However, to simple compute $X_A = X \oplus Q_A$ and extract $X = X_A \oplus Q_B$ with the same bit size, there need some expansion algorithm against $Q_A = h(sid, ID_A, ID_B, PW)$ and $Q_B = h(sid, ID_A, ID_B, PW)$ or additional hash function algorithm such as $\{0,1\}^* \rightarrow {}_RZ_p^*$. It is a tradeoff between convenience and efficiency.

5. **Security Analysis.** This section provides the heuristic security analysis and the provable security analysis [36-38] of the proposed password based authenticated key agreement protocol.

5.1. **Heuristic security analysis.** First, we define the security terms [19, 32] needed for security analysis of the proposed protocol as follows:

Definition 5.1. *A weak secret (Password PW) is a value of low entropy $Weak(k)$, which can be guessed in polynomial time.*

Definition 5.2. *The discrete logarithm problem (DLP) is explained by the following: Given a prime p, a generator g of ${}_RZ_p^*$, and an element $\beta \in {}_RZ_p^*$, find the integer α, $0 \leq \alpha \leq p-2$, such that $g^\alpha \equiv \beta \bmod p$.*

Definition 5.3. *The Diffie-Hellman problem (DHP) is explained by the following: Given a prime p, a generator g of ${}_RZ_p^*$, and elements $g^a \bmod p$ and $g^b \bmod p$, find $g^{ab} \bmod p$.*

Definition 5.4. *A secure one-way hash function $y = h(x)$ is one where given x to compute y is easy and given y to compute x is hard.*

Here, eight security properties [19, 32]: replay attack, password guessing attack, man-in-middle attack, modification attack, mutual authentication, known-key security, session key security, and perfect forward secrecy, must be considered for the proposed authenticated key agreement protocol. Under the above four definitions, the following theorems are used to analyze seven security properties in the proposed protocol.

Theorem 5.1. *The proposed protocol can resist the replay attack.*

Proof: Suppose an adversary Eve intercepts $X_A = X \oplus Q_A$ from Alice in Step KA1 and uses it to impersonate Alice by using replay attack. However, Eve cannot compute a correct a message authentication code (MAC) $V_A = h(sid, ID_A, ID_B, Y, K_A)$ and deliver it to Bob unless she can correctly guess password PW to obtain X from X_A and guess the right random integer number b from Y, and then Eve must face the discrete logarithm problem (DLP). On the other hand, suppose Eve intercepts Y and $V_B = h(sid, ID_B, ID_A, X, K_B)$ from Bob in Step KA2, and uses it to impersonate Bob by using replay attack. For the same reason, if Eve cannot gain the correct random integer number a from X_A, Alice will find out that $h(sid, ID_B, ID_A, X, K_A)$ is not equivalent to V_B, and then Alice will not send V_A back to Eve. Therefore, the proposed protocol can resist the replay attack.

Theorem 5.2. *The proposed protocol can resist the password guessing attacks.*

Proof: An on-line password guessing attack cannot succeed since Bob can choose appropriate trail intervals. On the other hand, in an off-line password guessing attack, due to the Definition 5.1, Eve can try to find out a weak password by repeatedly guessing possible passwords and verifying the correctness of the guesses based on information obtained in an off-line manner. In our proposed protocol, Eve can gain the knowledge of X_A, Y, V_B, and V_A in Steps KA1, KA2 and KA3, respectively. Assume that Eve wants to impersonate Alice. She first guesses password PW^* and then finds $X^* = X_A \oplus Q_A^*$ and Y_B, where $Q_A^* = g^{h(sid, ID_A, ID_B, PW^*)} \bmod p$. However, Eve has to break the discrete logarithm problem (DLP) and the Diffie-Hellman problem (DHP) [19] to find the keying material $K_A = K_B$ to verify her guess. Thus, Eve cannot gain the shared session key without X^* and the keying material K_A and K_B. Therefore, the proposed protocol can resist the password guessing attacks.

Theorem 5.3. *The proposed protocol can resist the man-in-middle attack.*

Proof: A mutual password PW between Alice and Bob is used to prevent the man-in-middle attack. The illegal Eve cannot pretend to be Alice or Bob to authenticate the other since she does not own the mutual password PW. Therefore, the proposed protocol can resist the man-in-middle attack.

Theorem 5.4. *The proposed protocol can resist the modification attack.*

Proof: Eve may modify the messages X_A, Y, V_B and V_A being transmitted over an insecure network. However, although Eve forges them, the proposed protocol can detect this modification attack, because it can verify not only the equality of K_A and K_B computed by each party, but also the correctness of X_A and Y transmitted between two parties through validating V_B and V_A in the proposed protocol. Therefore, the proposed protocol can resist the modification attack.

Theorem 5.5. *The proposed protocol provides secure mutual authentication.*

Proof: The proposed protocol performs secure mutual authentication of Alice and Bob because the password PW is not known to the attacker and the random values a and b are large and random. In the proposed protocol, the goal of mutual authentication is to generate an agreed session key SK between Alice and Bob for i-th session. In Step KA3, after Alice receives the response message $\{sid, ID_B, Y, V_B\}$ from Bob, he/she will check if the MAC value contains the $K_B = g^{xy}$ by checking $V_B \stackrel{?}{=} h(sid, ID_B, ID_A, X, K_A)$. Since the MAC value included $K_A = K_B = g^{ab}$, Alice will believe $\{sid, ID_B, Y, V_B\}$ was originally sent from Bob. In Step KA4, after Bob receives the hashed message $\{sid, ID_A, V_A\}$ from Alice, he/she will check if the MAC value contains the Diffie-Hellman $K_A = g^{ab}$ by checking $V_A \stackrel{?}{=} h(sid, ID_A, ID_B, Y, K_B)$. Since the hMAC value included $K_A = K_B = g^{ab}$,

Bob will believe $\{sid, ID_A, V_A\}$ was originally sent from Alice. Therefore, the proposed protocol can provide secure mutual authentication.

Theorem 5.6. *The proposed protocol provides known-key security.*

Proof: Known-key security means that each run of a key agreement protocol between two entities Alice and Bob should produce unique secret keys; such keys are called session keys. In the proposed protocol, after mutual authentication and session key agreement between Alice and Bob, $SK_A = h(sid, ID_A, ID_B, X, Y, K_A)$ and $SK_B = h(sid, ID_A, ID_B, X, Y, K_B)$ are used as a session key, where $K_A = K_B = g^{ab} \bmod p$, respectively. The random integer values a and b are never reused in the future sessions. It means that knowing a session keying materials $K_A = K_B = g^{ab} \bmod p$ and the random integer values a and b are of no use for computing the other session keying materials $K'_A = K'_B = g^{a'b'} \bmod p$, since without knowing a' and b', it is impossible to compute the session keying material $g^{a'b'} \bmod p$. Therefore, the proposed protocol provides known-key security.

Theorem 5.7. *The proposed protocol provides session key security.*

Proof: Session key security means that at the end of the key exchange, the session key is not known by anyone but Alice and Bob. The session key $SK_A = SK_B = h(sid, ID_A, ID_B, X, Y, g^{ab} \bmod p)$ is not known by anyone but Alice and Bob since the random integer values a and b are protected by the Diffie-Hellman problem (DHP) and the secure one-way hash function. None of this session keying material $g^{ab} \bmod p$ is known to anybody but Alice and Bob. Therefore, the proposed protocol provides session key security.

Theorem 5.8. *The proposed protocol provides perfect forward secrecy.*

Proof: Perfect forward secrecy means that if long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities is not affected. If the user's password PW is compromised, it does not allow Eve to determine the session key $SK_A = SK_B = h(sid, ID_A, ID_B, X, Y, g^{ab} \bmod p)$ for past sessions and decrypt them, since Eve is still faced with the Diffie-Hellman problem (DHP) to compute the session keying material $K_A = K_B = g^{ab} \bmod p$. Therefore, the proposed protocol satisfies the property of perfect forward secrecy.

The security properties of Lee-Lee's protocol [26] and the proposed protocol are summarized in Table 1. As shown in Table 1, Lee-Lee's protocol is insecure to the man-in-middle attack, modification attack and password guessing attack. Moreover, Lee-Lee's protocol cannot provide secure mutual authentication because an attacker can easily impersonate a

Table 1. Security properties of Lee-Lee's and the proposed protocols

	Lee-Lee's protocol	Proposed protocol
Replay attack	Secure	Secure
Password guessing attack	Insecure	Secure
Man-in-middle attack	Insecure	Secure
Modification attack	Insecure	Secure
Mutual authentication	No Provide	Provide
Known-key security	Provide	Provide
Session key security	Provide	Provide
Perfect forward secrecy	Provide	Provide

legal communication entity by performing man-in-middle attack or modification attack or password guessing attack. However, the proposed protocol satisfies all security properties compared with Lee-Lee's protocol.

5.2. **Provable security analysis.** In this subsection, we introduce a formal security model, which is mainly adopted from Bellare et al. [36, 37]. In addition, we formally define the special security requirements of password based authenticated key agreement protocol. The proposed formal proof method is based on Bellare et al.'s [36, 37] and Chang et al.'s proof methods [38].

5.2.1. *Formal security model.* Let Alice A and Bob B, two communicating parties, share a common password PW in advance. The model is principally used formally as follows.

1. Define the characteristics of participating entities.

   Protocol Participants. A party may have several *instances*, called oracles, involved in distinct concurrent executions of the protocols. We denote some instance i with an identifier A as Π_A^i.

   Long-lived keys. Two parties A and B share a common password PW. We call PW long-lived key and assume that the password is chosen independently and uniformly at random from the set $\{1, \ldots, D\}$, where D is a constant, independent of the security parameter.

   Session identity and partner identity. The session identity SID is used to uniquely name the ensuing session. $SID(\Pi_A^i)$ is the concatenation of all flows with the oracle Π_A^i. $PID(\Pi_A^i) = B$, denoted as Π_A^i, is the communication with another participant B. Both SID and PID are publicly available.

   Accepting and terminating. There are two states, $ACC(\Pi_A^i)$ and $TERM(\Pi_A^i)$, for an oracle Π_A^i. $ACC(\Pi_A^i) = true$ denotes that Π_A^i has enough information to compute a session key (SK). At any time an oracle can accept messages right away. As soon as Π_A^i is accepted, $SK(\Pi_A^i)$, $SID(\Pi_A^i)$ and $PID(\Pi_A^i)$ are defined. When an oracle sends or receives the last message of the protocol, receives an invalid message, or misses an expected message, the state of $TERM(\Pi_A^i)$ is set to $true$. As long as Π_A^i is terminated, no message will be sent out.
2. Define an adversary's capabilities.

   The adversary $\mathcal{A}$ has an endless supply of oracles and models various queries to them. Each query models a capability of the adversary, such as forward secrecy and know-key security. The six queries and their responses are listed below.

   (a) $Send(\Pi_A^i, m)$: This query models $\mathcal{A}$ sending a message m to Π_A^i. $\mathcal{A}$ gets back from his/her query the response which Π_A^i would have generated in processing message m and updates SID, PID, and its state. $\mathcal{A}$ in the form $Send(\Pi_A^i, start)$ initiates an execution of the protocol.

   (b) $Execute(\Pi_A^i, \Pi_B^j)$: This query models $\mathcal{A}$ obtaining an honest execution of the proposed password based authenticated key agreement (PAKA) protocol in the middle of two oracles Π_A^i and Π_B^j. $Execute(\Pi_A^i, \Pi_B^j)$ models $\mathcal{A}$ obtaining an honest execution of the protocols between two oracles Π_A^i and Π_B^j. This query may at first seem useless since $\mathcal{A}$ already can carry out an honest execution among oracles. Yet, the query is essential for properly dealing with on/off-line password guessing attacks.

   (c) $Reveal(\Pi_A^i)$: This query models $\mathcal{A}$ obtaining a session key (SK) with an unconditional return by Π_A^i. The query is for dealing with know-key security. The Reveal query is only available if the state $ACC(\Pi_A^i) = true$.

(d) $Corrupt(A)$: This query models $\mathcal{A}$ obtaining along-lived key PW with an unconditional return by A. The query is for dealing with perfect forward secrecy.
(e) $Hash(m)$: In the ideal hash model, $\mathcal{A}$ gets hash results by making queries to a random oracle. After receiving this query, the random oracle will check whether m has been queried. If so, it returns the result previously generated to $\mathcal{A}$; otherwise it generates a random number r and sends it to $\mathcal{A}$, and stores (m, r) into the $H-table$, which is a record set used to record all previous hash queries.
(f) $Test(\Pi_A^i)$: This query models the semantic security of the session key (SK) (the indistinguishability between the real session key and a random string). During an execution of the protocol, $\mathcal{A}$ can make any of the above queries, and at once, asks for a test query. Then, Π_A^i flips a coin b and returns SK if $b = 1$ or a random string with length $|SK|$ if $b = 0$. The query is only available if Π_A^i is $fresh$. $\mathcal{A}$ outputs a bit b' and wins the game of breaking the protocol if $b = b'$.

3. Formal specification of the proposed protocol PAKA.

Table 2 shows the initialization of both protocols. Table 3 shows how instances in the PAKA protocol behave in response to messages (runs the PAKA protocol). Before putting the protocol to work, each oracle sets $ACC(\Pi_U^i) \leftarrow TERM(\Pi_U^i) \leftarrow false$ and $SK(\Pi_U^i) \leftarrow SID(\Pi_U^i) \leftarrow PID(\Pi_U^i) \leftarrow null$.

TABLE 2. Specification of protocol initialization

$Initialize(1^k, 1^l)$, where l and k are security parameters and $l < k$ Select p prime with length $\|p\| = k$ and $p - 1 = l$; this defines group G; Choose random generator $g \leftarrow G$; Choose a hash function $h(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^l$ Publish parameters p, g, $h(\cdot)$; $< PW >_{A,B} \leftarrow \{1, \ldots, D\}$

5.2.2. *Definitions of security.* This subsection defines what constitutes the breaking of the our PAKA protocol. To begin with, let's set the formal notions of security as follows.

1. Freshness.

An oracle $\mathcal{A}$ is identified as fresh (or holds a fresh SK) if the following three conditions are satisfied:
(a) Π_A^i has been accepted,
(b) no oracle has been asked for a corrupt query before Π_A^i is accepted, and
(c) neither Π_A^i nor its partner has been asked for a reveal query.

2. Partnering.

In PAKA protocol, we say two oracles Π_A^i and Π_B^j are partnered if the following conditions are satisfied:
(a) Π_A^i and Π_B^j have been accepted,
(b) $SK(\Pi_A^i) = SK(\Pi_B^j)$,
(c) $SID(\Pi_A^i) \cap SID(\Pi_B^j) \neq 0$,
(d) $PID(\Pi_A^i) = B$ and $PID(\Pi_B^j) = A$, and
(e) no other oracle accepts $SK = SK(\Pi_A^i) = SK(\Pi_B^j)$.

3. AKE security (session key security).

We say $\mathcal{A}$ has the probability $Pr(win)$ to win a game of breaking the session key security of PAKA if $\mathcal{A}$ makes a single test query to a fresh oracle and correctly guesses

TABLE 3. Specification of the proposed protocol PAKA

$Execution(\Pi_A^i, \Pi_B^j)$
1. $Send_1(\Pi_A^i, start)$
   $< a > \leftarrow {}_RZ_p^*$; $X = g^a \bmod p$; $X_A = X \oplus PW$;
   $msg\_out_1 \leftarrow < sid, ID_A, X_A >$; $state_A^j \leftarrow < a, X >$;
   return msg_out_1
2. $Send_2(\Pi_B^j, m_1)$
   $< sid, ID_A, X_A > \leftarrow m_1$; $X \leftarrow X_A \oplus PW$; $< s > \leftarrow {}_RZ_p^*$; $Y = g^s \bmod p$;
   $K_B = (X)^s \bmod p$; $V_B = h(sid, ID_B, ID_A, X, K_B)$;
   $msg\_out_2 \leftarrow < sid, ID_B, Y, V_B >$;
   $state_B^j \leftarrow < Y, K_B >$;
   return msg_out_2
3. $Send_3(\Pi_A^i, m_2)$
   $< sid, ID_B, Y, V_B > \leftarrow m_2$; $< a, X > \leftarrow state_A^j$; $K_A \leftarrow (Y)^a \bmod p$;
   if $h(sid, ID_B, ID_A, X, K_A) = V_B$
     $V_A = h(sid, ID_A, ID_B, Y, K_A)$;
     $msg\_out_3 \leftarrow < sid, ID_A, V_A >$;
     $SK(\Pi_A^i) \leftarrow h(sid, ID_A, ID_B, X, Y, K_A)$;
     $SID(\Pi_A^i) \leftarrow < msg\_out_l, m_2, msg\_out_3 >$;
     $PID(\Pi_A^i) \leftarrow B$; $ACC(\Pi_A^i) \leftarrow true$; $TERM(\Pi_A^i) \leftarrow true$;
   else $msg\_out_3 \leftarrow *$;
4. $Send_4(\Pi_B^j, m_3)$
   $< sid, ID_A, V_A > \leftarrow m_3$; $< Y, K_B > \leftarrow state_B^j$;
   if $h(sid, ID_A, ID_B, Y, K_B) = V_A$
     $SK(\Pi_B^j) \leftarrow h(sid, ID_A, ID_B, X, Y, K_B)$; $SID(\Pi_B^j) \leftarrow < m_l, msg\_out_2, m_3 >$;
     $PID(\Pi_B^j) \leftarrow A$; $ACC(\Pi_B^j) \leftarrow true$; $TERM(\Pi_B^j) \leftarrow true$
   return $null$

the bit b used in the game. We denote the AKE advantage of $\mathcal{A}$ in attacking PAKA as $Adv_{PAKA}^{AKE}(\mathcal{A})$; the advantages are taken over all bit tosses. The advantage of $\mathcal{A}$ distinguishing the session key is given by $Adv_{PAKA}^{AKE}(\mathcal{A}) = 2Pr(win) - 1$. Protocol PAKA is AKE-secure if $Adv_{PAKA}^{AKE}(\mathcal{A})$ is negligible.

4. Computational Diffie-Hellman (CDH) assumption.
   Let $G =< g >$ be a cyclic of prime order $p - 1$ and x, y chosen at random in Z_p^*. Let $\mathcal{B}$ be a CDH-adversary that given a challenge $\psi = (g^x, g^y)$, and let ε be the probability that $\mathcal{B}$ can output an element z in G such that $z = g^{xy}$. We denote this success probability as $Succ_G^{CDH}(\mathcal{B})$. The CDH problem is intractable if $Succ_G^{CDH}(\mathcal{B})$ is negligible.
5. Adversary's resources.
   The security can be formulated as a function of the amount of resources $\mathcal{A}$ obtains. The resources are as follows:
   (a) t: time of computing,
   (b) q_{se_i}, q_{ex}, q_{re}, q_{co}, q_h: the number of $Send_i$, $Execute$, $Reveal$, $Corrupt$, and $Hash$ queries separately made. Here, q_{se} is the total number of q_{se_i}.

5.2.3. *Security proof.*

Theorem 5.9. *Let $\mathcal{A}$ be an adversary against the AKE-security of the PAKA protocol within a time bound t, after q_{se} and q_h. Then we have:*

$$Adv^{AKE}_{PAKA}(\mathcal{A}) \leqslant \frac{q_{se}}{|\mathcal{D}|} + q_{se}q_h Succ^{CDH}_{G}(t_1) + \frac{q_{se}}{2^l}$$

where t_1 is the running time of $Succ^{CDH}_{G}$.

Proof: There are three ways that might lead to $\mathcal{A}$ successfully attacking the AKE-security of the PAKA protocol. First, $\mathcal{A}$ might obtain the long-lived key and impersonate A or B by mounting the on/off-line password guessing attack. Second, $\mathcal{A}$ might directly obtain the session key by solving the CDH problem. In the following, we shall analyze the probability of the two situations one by one. To analyze a situation, the others are assumed to be under some known probability.

1. On/off-line password guessing attacks.

   A and B separately chooses $a \leftarrow {}_RZ^*_p$ and $b \leftarrow {}_RZ^*_p$ at random, which implies X $(= g^a \bmod p)$ and Y $(= g^b \bmod p)$ are random numbers. Hence, $\mathcal{A}$ observes that the message $(X_A = X \oplus PW)$ returned from $Send_1$ is independent of other messages. Therefore, the adversary gets no advantage for the off-line password guessing attack. The probability of the on-line password guessing attack making way is bounded by q_se and D as follows:

   $$\lambda \leqslant \frac{q_{se}}{|\mathcal{D}|}$$

   The on-line guessing attack can be prevented by letting the server B take the appropriate intervals between trials.
2. CDH attack (session key).

   $\mathcal{B}$ plays the role of a simulator for indistinguishability. It uses the PAKA protocol to respond to all $\mathcal{A}$'s queries and deal with the CDH problem. $\mathcal{B}$ sets up the long-lived key PW, picks a random number i from $[1, q_{se_1}]$, and sets a counter $cnt = 0$. When $\mathcal{A}$ makes $Send_1$, $\mathcal{B}$ answers according to the protocol to return msg_out_1 to $Send_1$ and increases cnt by 1. If $cnt \neq i$, $\mathcal{B}$ answers with msg_out_2 to $Send_2$. If $cnt = i$, $\mathcal{B}$ answers with $< g^y, h(random, g^x) >$ by using the element g^x from the challenge ψ. When $\mathcal{A}$ makes $Send_3$, if the input is the flow corresponding to challenge ψ, $\mathcal{B}$ answers with $< h(random, g^y) >$ by using the element g^y from the challenge ψ. If not, $\mathcal{B}$ answers with msg_out_3 to $Send_3$.

   When $\mathcal{A}$ makes a $Reveal(\Pi^i_A)$ or $Reveal(\Pi^j_B)$, $\mathcal{B}$ checks whether the oracle has been accepted and is $fresh$. If so, $\mathcal{B}_1$ answers by using the session key SK. However, if the session key has to be constructed from the challenge ψ, $\mathcal{B}$ halts. When $\mathcal{A}$ makes a $Corrupt(A)$, $Corrupt(B)$, $Execute(\Pi^i_A, \Pi^j_B)$, or $Hash(m)$, $\mathcal{B}$ answers in a straightforward way. When $\mathcal{A}$ makes a single test query, $\mathcal{B}$ answers in a straightforward way. However, if the session key has to be constructed from the challenge ψ, $\mathcal{B}$ answers with a random string for the $Test(\Pi^i_A)$ or $Test(\Pi^j_B)$.

   This simulation is perfectly indistinguishable from any execution of the real PAKA protocol except for one execution in which the challenge ψ is involved. The probability α of $\mathcal{B}$ correctly guessing the session key $\mathcal{A}$ will use $test(\Pi^i_A)$ is the probability of $cnt = i$. Then, we have

   $$\alpha = \frac{1}{q_{se_1}} \leqslant \frac{1}{q_{se}}$$

   Assume that $\mathcal{A}$ has broken the CDH problem ($\mathcal{A}$ outputting b' after the test query, $wins$), then at least one of the hash queries equals SK. The probability of $\mathcal{B}$ correctly

choosing among the possible hash queries is

$$\beta \leqslant \frac{1}{q_h}$$

From the above description, the probability $Succ_G^{CDH}(\mathcal{B})$ that $\mathcal{B}$ outputs z from the challenge ψ is the probability ε that $\mathcal{A}$ breaks the AKE-secure protocol multiplied by the probability α that $\mathcal{B}_1$ correctly guesses the moment at which $\mathcal{A}$ breaks the AKE-secure protocol multiplied by the probability β that $\mathcal{B}_1$ correctly chooses among the possible hash queries:

$$Succ_G^{CDH}(\mathcal{B}) = \varepsilon \times \alpha \times \beta \leqslant \varepsilon \times \frac{1}{q_{se}} \times \frac{1}{q_h}$$

6. **Performance Comparison.** This section compared the proposed protocol with other password based key agreement protocols submitted to IEEE P1363.2 (Password-based Techniques) [39, 40] and Lee-Lee's protocol [26]. Table 4 shows the comparison results of the computational costs of the proposed protocol and of various password-based protocols based on an asymmetric model. In order to compare the computational workload, we considered the number of exponentiations that consume the most execution time. In Table 4, we use this counting method for a number of exponentiations.

Table 4. The comparison of computational costs

	B-SPEKE	SRP6	AMP2	PAK-Y	Lee-Lee	Proposed
# of passes	3	4	4	3	4	**3 (3)**
# of random numbers	3	2	2	3	2	**2 (2)**
# of Alice's exponentiations	3	3	3	5	2	**3 (2)**
# of Bob's exponentiations	4	3	4	5	2	**3 (2)**
# of hash operations	6	6	9	8	4	**8 (8)**

( ): Costs of the proposed protocol considering computational efficiency

SRP6, AMP2 and Lee-Lee's protocol are four-pass protocols for password-based authenticated key agreement, but B-SPEKE and PAK-Y are three-pass protocols. SRP6, AMP2 and Lee-Lee's protocol require the smallest random numbers, B-SPEKE and PAK-Y require the smallest computational passes, SRP6 and Lee-Lee's protocol require the smallest exponentiations, and Lee-Lee's protocol require the smallest hash operations among the previously proposed protocols, respectively.

By contrast, the proposed protocol implements a three-pass protocol. In the proposed protocol, each party performs approximately two random number generations, three exponentiations, and four hash operations. The exchanged data size is only $3|p| + 2|h(\cdot)|$, where p is a generator of the group of points of order q. Furthermore, the computational costs of the proposed protocol considering computational efficiency can be more reduced in case of exponentiations. Therefore, as in Table 4, we can see that the proposed protocol has the smallest computational and communicational workloads.

7. **Conclusions.** The current paper demonstrated the vulnerabilities of Lee-Lee's password based authenticated key agreement protocol to off-line password guessing attacks. Then, to resolve such security problems, we presented an improved protocol. As a result, the proposed password based authenticated key agreement protocol is not only secure

against well-known cryptographical attacks and but also provides good performance since it provides mutual authentication and a session key agreement between two communication parties. Accordingly, the proposed protocol has several important features and advantages. For example, (1) the proposed protocol is designed to optimize the computation cost of each participant by using the small communication round; (2) the proposed protocol does not require public key cryptosystem and digital signatures; (3) the proposed protocol is secure against well-known cryptographical attacks; (4) the proposed protocol provides secure mutual authentication, known-key security, session key security, and perfect forward secrecy. Thus, the proposed protocol is very useful in password-based Internet and wire/wireless communication environments to access remote information systems since it provides security, reliability and efficiency.

Acknowledgements. Eun-Jun Yoon was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0010106) and partially supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2012-H0301-12-2004) supervised by the NIPA (National IT Industry Promotion Agency). Kee-Young Yoo was supported by Kyungpook National University Research Fund, 2012.

REFERENCES

[1] S. Bellovin and M. Merritt, Encrypted key exchange: Password-based protocols secure against dictionary attacks, *Proc. of IEEE Conf. on Research in Security and Privacy*, pp.72-84, 1992.

[2] Z. Zhang, B. Fang, M. Hu and H. Zhang, Security analysis of session initiation protocol, *International Journal of Innovative Computing, Information and Control*, vol.3, no.2, pp.457-469, 2007.

[3] Y.-F. Chang, A practical three-party key exchange protocol with round efficiency, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.953-960, 2008.

[4] C.-Y. Chen, H.-F. Lin and C.-C. Chang, An efficient generalized group-oriented signature scheme, *International Journal of Innovative Computing, Information and Control*, vol.4, no.6, pp.1335-1345, 2008.

[5] J.-S. Lee, Y.-F. Chang and C.-C. Chang, Secure authentication protocols for mobile commerce transactions, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp.2305-2314, 2008.

[6] H.-F. Huang and W.-C. Wei, A new efficient and complete remote user authentication protocol with smart cards, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.2803-2808, 2008.

[7] C.-L. Chen, Y.-Y. Chen and Y.-H. Chen, Group-based authentication to protect digital content for business applications, *International Journal of Innovative Computing, Information and Control*, vol.5, no.5, pp.1243-1251, 2009.

[8] R.-C. Wang, W.-S. Juang and C.-L. Lei, A robust authentication scheme with user anonymity for wireless environments, *International Journal of Innovative Computing, Information and Control*, vol.5, no.4, pp.1069-1080, 2009.

[9] T.-H. Chen, An authentication protocol with billing non-repudiation to personal communication systems, *International Journal of Innovative Computing, Information and Control*, vol.5, no.9, pp.2657-2664, 2009.

[10] C.-T. Li, C.-H. Wei and Y.-H. Chin, A secure event update protocol for peer-to-peer massively multiplayer online games against masquerade attacks, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4715-4723, 2009.

[11] Y. Ding and P. Horster, Undetectable on-line password guessing attacks, *ACM Operating Systems Review*, vol.2, no.4, pp.77-86, 1995.

[12] C. C. Yang, T. Y. Chang and M. S. Hwang, Security of improvement on methods for protecting password transmission, *International Journal of Informatica*, vol.14, no.4, pp.551-558, 2003.

[13] W. C. Ku and H. C. Tsai, Weaknesses and improvements of Yang-Chang-Hwang's password authentication scheme, *International Journal of Informatica*, vol.16, no.2, pp.203-212, 2005.

[14] E. J. Yoon, E. K. Ryu and K. Y. Yoo, Attacks and solutions of Yang et al.'s protected password changing scheme, *International Journal of Informatica*, vol.16, no.2, pp.285-294, 2005.
[15] K. K. Raymond Choo, On the security analysis of Lee, Hwang & Lee (2004) and Song & Kim (2000) key exchange agreement protocols, *International Journal of Informatica*, vol.17, no.4, pp.467-480, 2006.
[16] T. H. Chen, G. Horng and K. C. Wu, A secure YS-like user authentication scheme, *International Journal of Informatica*, vol.18, no.1, pp.27-36, 2007.
[17] Y. M. Tseng, T. Y. Wu and J. D. Wu, A pairing-based user authentication scheme for wireless clients with smart cards, *International Journal of Informatica*, vol.19, no.2, pp.285-302, 2008.
[18] D. H. Seo and P. Sweeney, Simple authenticated key agreement algorithm, *Electronics Letters*, vol.35, no.13, pp.1073-1074, 1999.
[19] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transaction on Information Theory*, vol.22, no.6, pp.644-654, 1976.
[20] Y. M. Tseng, Weakness in simple authenticated key agreement protocol, *Electronics Letters*, vol.36, no.1, pp.48-49, 2000.
[21] W. C. Ku and S. D. Wang, Cryptanalysis of modified authenticated key agreement protocol, *Electronics Letters*, vol.36, no.21, pp.1770-1771, 2000.
[22] C. L. Hsu, T. S. Wu, T. C. Wu and C. Mitchell, Improvement of modified authenticated key agreement protocol, *Applied Mathematics and Computation*, vol.142, no.2-3, pp.305-308, 2003.
[23] N. Y. Lee and M. F. Lee, Further improvement on the modified authenticated key agreement scheme, *Applied Mathematics and Computation*, vol.157, no.3, pp.729-733, 2004.
[24] T. Y. Chang, C. C. Yang and C. M. Chen, Improvement on pretty-simple password authenticated key-exchange protocol for wireless networks, *International Journal of Informatica*, vol.15, no.2, pp.161-170, 2004.
[25] Y. S. Kim, E. N. Huh, J. H. Hwang and B. W. Lee, An efficient key agreement protocol for secure authentication, *ICCSA 2004, LNCS*, vol.3043, pp.746-754, 2004.
[26] K. J. Lee and B. J. Lee, Cryptanalysis of the modified authenticated key agreement scheme, *Applied Mathematics and Computation*, vol.170, no.1, pp.280-284, 2005.
[27] R. Lu and Z. Cao, Off-line password guessing attack on an efficient key agreement protocol for secure authentication, *International Journal of Network Security*, vol.2, no.2, pp.126-129, 2006.
[28] E. J. Yoon and K. Y. Yoo, Robust two-party password based authenticated key agreement protocol, *Proc. of IEEE Computer Society Conf. on Convergence Information Technology*, pp.2078-2084, 2007.
[29] C. C. Chang and S. Y. Lin, An improvement on authenticated key agreement scheme, *Proc. of IEEE Conf. on Intelligent Pervasive Computing*, pp.3-6, 2007.
[30] X. Zuo, F. Liu and C. Ren, Cryptanalysis of a password-based key exchange protocol, *Proc. of the 2nd International Conference on Innovative Computing, Information and Control*, Kumamoto, Japan, pp.610-613, 2007.
[31] E. J. Yoon and K. Y. Yoo, SAKA$_{WP}$: Simple authenticated key agreement protocol based on weil pairing, *Proc. of IEEE Computer Society Conf. on Convergence Information Technology*, pp.2096-2101, 2007.
[32] B. Schneier, *Applied Cryptography Protocols, Algorithms and Source Code in C*, 2nd Edition, John Wiley & Sons Inc, 1995.
[33] A. J. Menezes, P. C. Oorschot and S. A. Vanstone, *Handbook of Applied Cryptograph*, CRC Press, New York, 1997.
[34] *BruteForceCalc, Brute Force Attack Estimator*, Mandylion Research Labs, http://www.mandy lion-labs.com/PRCCalc/BruteForceCalc.htm, 2006.
[35] E. J. Yoon and K. Y. Yoo, Cryptanalysis of a simple three-party password-based key exchange protocol, *International Journal of Communication Systems*, 2010.
[36] M. Bellare and P. Rogaway, Provably secure session key distribution: The three party case, *Proc. of the 27th ACM Symposium on the Theory of Computing*, pp.57-66, 1995.
[37] M. Bellare, D. Pointcheval and P. Rogaway, Authenticated key exchange secure against dictionary attacks, *Proc. of Eurocrypt 2000, LNCS*, vol.1807, pp.139-155, 2000.
[38] T. Y. Chang, W. P. Yang and M. S. Hwang, Simple authenticated key agreement and protected password change protocol, *Computers & Mathematics with Applications*, vol.49, no.5-6, pp.703-714, 2005.
[39] Y. H. Hwang, D. H. Yum and P. J. Lee, EPA: An efficient password-based protocol for authenticated key exchange, *ACISP 2003, LNCS*, vol.2727, pp.452-463, 2003.

[40] *The Latest Draft of IEEE P1363.2. Standard Specifications for Password-Based Public Key Cryptography Techniques, Draft D19*, http://grouper.ieee.org/groups/1363/passwdPK/ index.html, 2004.