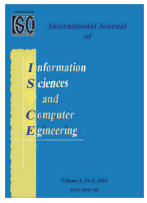
Information Sciences and Computer Engineering, Vol. 1, No. 1, (2010) 77–87

PHARUS

International Journal of
Information Sciences and Computer Engineering

journal homepage: http://www.ijisce.org



A Secure Protocol for Three-Party Authenticated Key Exchange with Provable Belief

Ting-Fang Cheng[a], Chin-Chen Chang[a,b,*], Zhi-Hui Wang[c]

[a]*Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan, 30013, R.O.C.*
[b]*Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan, 40724, R.O.C.*
[c]*School of Software, Dalian University of Technology, Dalian, Liaoning, China.*

***Abstract*– Three-Party authenticated key exchange (3PAKE) protocol is an important cryptographic mechanism in which two clients can request the session key for communicating with each other and one trusted server takes the responsibility for authenticating users and key agreements. In 2007, Lu and Cao proposed a simple 3PAKE protocol. Nevertheless, we find that it is vulnerable to the off-line password guessing attack and the impersonation attack. We therefore propose a novel version using smart cards to withstand more malicious attacks. We also give a formal correctness analysis of mutual authentication to our scheme using BAN authentication logic. What is more, we make detail discussions for highlighting that our proposed scheme can prevent several malicious attacks and is more efficient than other related works.**

Keyword: 3PAKE, off-line password guessing attack, impersonation attack, mutual authentication, key agreement, BAN logic.

1. Introduction

Three-party authenticated key exchange (3PAKE) protocol is one that allows any two clients to share an easy-to-remember key with a trusted server while the server acts as a bridge between the two clients. With such a system, the involved communicating parties can securely contribute and exchange keys via the server. In addition, the server can help the involved parties to authenticate each other by using pre-shared passwords. Only valid users can decrypt messages sent by the server to retrieve correct session keys. Recently, the 3PAKE mechanism has become one of the most important cryptographic tools for supporting secure information exchange on the Internet. Recently, lots of literatures for 3PAKE protocols have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

The following will discuss the properties that a well designed 3PAKE system should contain [1, 3, 5, 7, 9].

Mutual authentication: Considering fairness, each of the involved parties is able to authenticate one another in the network system before they negotiate a common session key.

Session key security: In order to enhance the security of message exchange for preventing eavesdropping, common session key distribution is necessary. Moreover, only the involved participants can construct the valid session key in one protocol run.

Resistance to malicious attacks: The protocol shall be secure enough to prevent against various malicious attacks such as detectable and undetectable on-line password guessing attacks, off-line password guessing attacks, impersonation attacks, etc.

Perfect forward/backward secrecy: A protocol has perfect forward/backward secrecy if the compromise of passwords does not divulge previous/following session keys.

Known-key security: The compromise of one session key does not divulge other session keys in different protocol runs.

In 1995, Steiner et al. proposed a 3PAKE protocol [10] based on Bellovin and Merritt's encrypted key exchange concept [13]. Unfortunately, Lin et al. found that Steiner et al.'s scheme may suffer undetectable on-line password guessing attacks and off-line password guessing attacks. Hence, Lin et al. proposed an improved version using the server's public key to prevent various attacks in 2000 [6]. Subsequently, in order to enhance efficiency, Lin et al. proposed some 3PAKE protocols without adopting public key cryptographic technology in 2001 and 2004 [4, 7]. Thereafter, Wen et al. utilized the Weil pairing concept to design a 3PAKE protocol in 2005 [14]. However, in 2007, Nam et al. detected that Wen et al.'s protocol was vulnerable to man-in-the-middle attacks[15].

Furthermore, according to Abdalla and Pointcheval's simple password-based encrypted key exchange protocol, Lu and Cao designed a new 3PAKE protocol in 2007 [9] (hereafter referred to as S-3PAKE). Nevertheless, we find that their scheme cannot resist the off-line password guessing attacks and the impersonation attacks. Motivated by the weaknesses on S-3PAKE, we consequently provide a more secure 3PAKE protocol using smart cards without raising computation costs. In particular, we adopt the BAN logic [16, 17] to demonstrate the accuracy of mutual authentication and key distribution in our proposed scheme. The BAN logic is a formal tool provided by Burrows et al. that can

*Corresponding author:
Email address: ccc@cs.ccu.edu.tw, Ph: +24 517250, Fax: +27 066495

help researchers in analyzing the accuracy of an authentication protocol.

The remainder of this article is organized as follows. We briefly review Lu and Cao's S-3PAKE scheme and point out vulnerabilities in it in Section 2. The description of our proposed BP-3PAKE scheme is introduced in Section 3, followed by the demonstration of the accuracy of our scheme using BAN logic in Section 4. The security analyses and performance comparisons of BP-3PAKE are given in Section 5. Finally, we make conclusions in Section 6.

2. Related Works

In this section, we review Lu and Cao's S-3PAKE [9] and explain the vulnerability in their scheme in Subsections 2.1 and 2.2, respectively.

2.1. Review of S-3PAKE

The S-3PAKE system is composed of three participants: one trusted authentication center (AC) and two users who want to communicate with each other. AC takes the responsibility for authenticating the validity of any two users and assists them in negotiating a common session key. Assuming that Alice and Bob want to communicate with each other, they need to register at AC first. Then they can authenticate each other through AC before coordinating their common session key.

We assume that this protocol run is started by Alice. The flowchart of Lu and Cao's S-3PAKE is depicted in Fig. 1. The notations used throughout S-3PAKE are defined as follows.

G: a finite cyclic group
q: a large prime
g: a generator with order q in G
x/y: two elements in G
pw_A: Alice's password shared with AC
pw_B: Bob's password shared with AC
$h_1(\cdot)$/$h_2(\cdot)$: two secure one-way hash functions
$||$: the concatenation symbol

Note that G, q, g, x, y, $h_1(\cdot)$, and $h_2(\cdot)$ are public information.

Step 1. Alice randomly selects a number $r_A \in Z_q$. Then she computes $\alpha = g^{r_A} \cdot x^{pw_A}$ and sends it to Bob along with her identity.

Step 2. Upon receiving the message, Bob also randomly chooses a number $r_B \in Z_q$ and computes $\beta = g^{r_B} \cdot y^{pw_B}$. Subsequently, he sends $\{\text{Alice}||\ \alpha||\text{Bob}||\ \beta\}$ to AC.

Step 3. Once AC obtains the message, it can use pw_A and pw_B to retrieve g^{r_A} and g^{r_B} by computing $\frac{\alpha}{x^{pw_A}}$ and $\frac{\beta}{y^{pw_B}}$, respectively. AC then also randomly selects a number $r_{\in}Z_q$ and computes $\alpha' = (g^{r_B})^r \cdot h_1(\text{Alice,AC}, g^{r_A})^{pw_A}$ and $\beta' = (g^{r_A})^r \cdot h_1(\text{Bob,AC}, g^{r_B})^{pw_B}$. Finally, AC transfers $\{\alpha'||\beta'\}$ back to Bob.

Step 4. Upon receiving the message from AC, Bob computes $g^{r \cdot r_A} = \frac{\beta'}{h_1(\text{Bob,AC},g^{r_B})^{pw_B}}$. Consequently, he is able to compute $g^{r \cdot r_A \cdot r_B}$ and $\Gamma = h_1(\text{Alice,Bob}, g^{r \cdot r_A \cdot r_B})$. He subsequently sends $\{\alpha'||\Gamma\}$ to Alice.

Step 5. When Alice receives the message, she computes $g^{r \cdot r_B} = \frac{\alpha'}{h_1(\text{Alice,AC},g^{r_A})^{pw_A}}$ and $g^{r \cdot r_B \cdot r_A}$ in the same way. Then she computes and verifies whether $\Gamma = h_1(\text{Alice,Bob}, g^{r \cdot r_B \cdot r_A})$. If it does not hold, Alice halts the communication; otherwise, she calculates a session key $SK = h_2(\text{Alice,Bob}, g^{r \cdot r_A \cdot r_B})$ and a verification token $\Gamma' = h_1(\text{Bob,Alice}, g^{r \cdot r_A \cdot r_B})$. Afterward, she returns Γ' to Bob.

Step 6. After Bob receives this message, he computes and checks if $\Gamma' = h_1(\text{Bob,Alice}, g^{r \cdot r_A \cdot r_B})$. If it does not hold, Bob terminates the communication; otherwise, he also computes a session key as $SK = h_2(\text{Alice,Bob}, g^{r \cdot r_A \cdot r_B})$. Undoubtedly, the session keys which computed by Alice and Bob shall be equal if they both follow these procedures.

2.2. Analyses of S-3PAKE

Lu and Cao asserted that their proposed S-3PAKE can achieve the requirements of general 3PAKE mechanisms. However, we show that it is vulnerable to the off-line password guessing attacks. In addition, there still stands a design weakness, which leads to incurring impersonation attacks.

2.2.1. Off-Line Password Guessing Attack

The prevention of off-line password guessing attacks means that "anyone can not get useful information to check the correctness of the guessed passwords off-line" [6]. Assume that semi-honest Bob intends to mount such an attack for guessing Alice's password. He can easily procure his purpose as follows.

Step 1. When Bob receives $\{\text{Alice}||\ \alpha\}$, he first replaces α with x^{r_B} and computes $\beta = x \cdot y^{pw_B}$. Then he sends $\{\text{Alice}||\ x^{r_B}||\text{Bob}||\ \beta\}$ to AC.

Step 2. Upon receiving the massage, according to the protocol rules, AC computes

$$\frac{x^{r_B}}{x^{pw_A}} = x^{r_B - pw_A}$$

and

$$\frac{\beta}{y^{pw_B}} = x.$$

AC then selects a random number r and computes

$$\alpha' = (x)^r \cdot h_1(\text{Alice,AC}, x^{r_B - pw_A})^{pw_A}$$

and

$$\beta' = (x^{r_B - pw_A})^r \cdot h_1(\text{Bob,AC}, x)^{pw_B}.$$

Finally, AC transfers $\{\alpha'||\beta'\}$ back to Bob.

Step 3. Once Bob obtains the message $\{\alpha'||\beta'\}$, he terminates the connection and uses pw_B and β' to retrieve $(x^{r_B - pw_A})^r = \frac{\beta'}{h_1(\text{Bob,AC},x)^{pw_B}}$. Bob then executes the attack by following sub-steps.

Step 3.1. He guesses a password, pw.

Step 3.2. He computes $\sigma = (x^{r_B - pw_A})^r \cdot h_1(\text{Alice,AC}, x^{r_B - pw})^{pw \cdot (r_B - pw)}$.

Step 3.3. He computes and compares whether $\sigma = (\alpha')^{r_B - pw}$. If this equation holds, Bob hits the correct pw_A; otherwise, returns to Step 3.1.

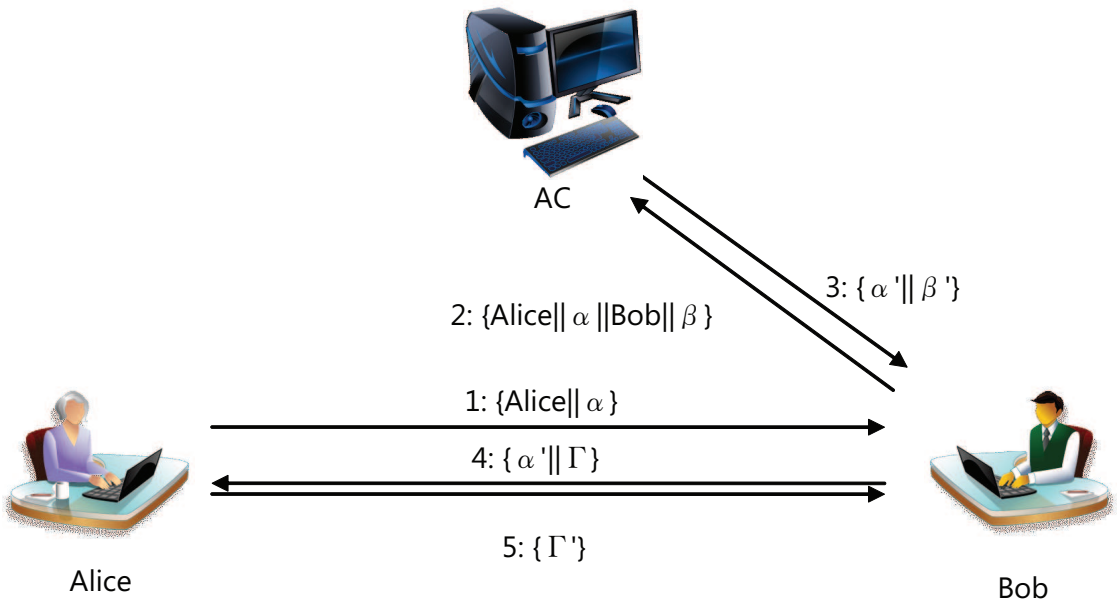


Fig. 1. The flowchart of S-3PAKE.

2.2.2. Impersonation Attack on a Design Weakness

In general, the concept of the 3PAKE protocol is that any two clients are able to communicate with each other through a trusted server. This trusted server is responsible to authenticate the validity of any two involved parties and help in coordinating a common session key. The trusted server is an indispensable role in a 3PAKE system. However, in Lu and Cao's S-3PAKE scheme, the trusted server (AC) never verifies the validity of all involved users and the accuracy of the receiving messages. AC just collects the requested messages from Alice and Bob and further computes responses to them for negotiating their common session key. Such a circumstance may allow a valid but dishonest user to make an impersonation attack and fool the communication sponsor.

We assume that there exists a dishonest user, Eve, who attempts to impersonate Bob to spoof Alice by performing the following processes.

Step 1. Eve intercepts the message {Alice || α}, which Alice sent to Bob and chooses a random number $r_E \in Z_q$. She then computes $\beta = g^{r_E} \cdot y^{pw_E}$ and sends {Alice||α||Eve||β} to AC.

Step 2. Once AC obtains the message, it computes $g^{r_A} = \frac{\alpha}{x^{pw_A}}$ and $g^{r_E} = \frac{\beta}{y^{pw_E}}$ and selects a random number $r_{\in} Z_q$. Subsequently, AC computes $\alpha' = (g^{r_E})^r \cdot h_1(\text{Alice,AC}, g^{r_A})^{pw_A}$ and $\beta' = (g^{r_A})^r \cdot h_1(\text{Eve,AC}, g^{r_E})^{pw_E}$ and transfers $\{\alpha'||\beta'\}$ back to Eve.

Step 3. Upon receiving the message, Eve retrieves $g^{r \cdot r_A} = \frac{\beta'}{h_1(\text{Eve,AC},g^{r_E})^{pw_E}}$ and computes $\Gamma = h_1(\text{Alice,Bob}, g^{r \cdot r_A \cdot r_E})$. She subsequently sends $\{\alpha'||\Gamma\}$ to Alice.

Step 4. After Alice receives the message, she is able to retrieve $g^{r \cdot r_E} = \frac{\alpha'}{h_1(\text{Alice,AC},g^{r_A})^{pw_A}}$. Then Eve computes and verifies whether $\Gamma = h_1(\text{Alice,Bob}, g^{r \cdot r_E \cdot r_A})$. Obviously, the verification would pass and the session key $SK = h_2(\text{Alice,Bob}, g^{r \cdot r_A \cdot r_E})$ computed by her would be the same as Eve's.

Consequently, Alice will then believe that Eve is Bob, whom she wants to communicate with. As the result, S-3PAKE is weak in resisting the impersonation attack.

3. Belief-Provable 3PAKE (BP-3PAKE)

In this section, we present a more secure 3PAKE scheme using smart cards (BP-3PAKE). Similar to the S-3PAKE, this system consists of one trusted authentication center (AC) and two users who want to communicate with each other. AC is responsible for issuing a smart card to each user and for helping any two users with mutual authentication and key agreements. The smart card considered in this article is the tamper-resistant IC Processor Card with 256 KB of programmable ROM, and a 16-bit microprocessor. Furthermore, this smart card contains cryptography algorithms such as AES, SHA-256, and random numbers [18].

Our scheme is composed of a registration phase and a communication phase. The details of these phases are described in the following subsections with the flowchart depicted in Fig. 2. The notations used in BP-3PAKE are defined as follows.

k: AC's secret key
pw_A: Alice's password
pw_B: Bob's password
$h(\cdot)$: a secure one-way hash function
$||$: the concatenation symbol
$E_K(\cdot)$: an AES-based encryption with key K
$D_K(\cdot)$: an AES-based decryption with key K

3.1. Registration Phase

In BP-3PAKE, if Alice and Bob want to communicate with each other, they would need to register at AC first by following the steps below.

Step 1. Alice selects a password, pw_A, and then sends the request message with her identifier and pw_A to the server, AC, through a secure channel.

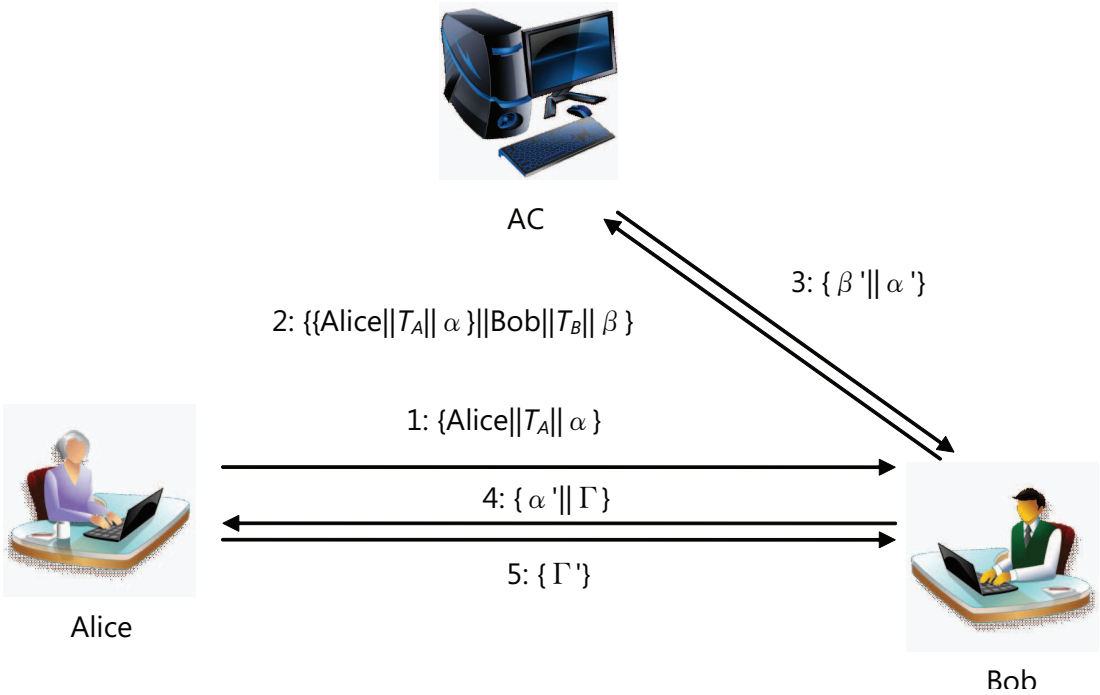


Fig. 2. The flowchart of BP-3PAKE.

Step 2. Upon receiving the registration request, AC checks the validity of Alice. If Alice's credit is invalid, the server rejects the registration; otherwise, it goes to the next step.

Step 3. AC computes a transformed password

$$TPW_A = h(pw_A)$$

and

$$\delta_A = h(\text{Alice}||k) \oplus pw_A$$

and embeds Alice, $h(\cdot)$, TPW_A, and δ_A into a smart card $Card_A$. AC subsequently sends $Card_A$ to Alice through a secure channel.

Similarly, Bob would be able to obtain a smart card $Card_B$ after he registers at AC through the above registration steps.

3.2. Communication phase

We assume that this protocol run is initiated by Alice. Then Alice, Bob, and AC perform following procedures.

Step 1. Alice inserts her smart card, $Card_A$, into the input device and keys in the password pw_A^* and the identifier, Bob, whom she want to communicate with. Next, $Card_A$ carries out the following sub-steps.

Step 1.1. $Card_A$ computes $h(pw_A^*)$ and compares it with TPW_A. If they are different, it ceases the procedure; otherwise, it goes to the next sub-step.

Step 1.2. $Card_A$ generates a random number r_A and computes

$$K_{AS} = h((\delta_A \oplus pw_A^*)||T_A)$$

and

$$\alpha = E_{K_{AS}}(\text{Bob}||T_A||r_A),$$

where T_A is the current timestamp. $Card_A$ then transmits $\{\text{Alice}||T_A||\alpha\}$ to Bob.

Step 2. When Bob receives the message from Alice, he also inserts his $Card_B$ into the input device and enters pw_B^* and Alice's identifier. $Card_B$ subsequently runs the following sub-steps.

Step 2.1. $Card_B$ computes $h(pw_B^*)$ and compares it with TPW_B. If they are different, it halts the procedure; otherwise, it goes to the next sub-step.

Step 2.2. $Card_B$ then generates a random number r_B and computes

$$K_{BS} = h((\delta_B \oplus pw_B^*)||T_B)$$

and

$$\beta = E_{K_{BS}}(\text{Alice}||T_B||r_B),$$

where T_B is the current timestamp. Finally, $Card_B$ forwards $\{\{\text{Alice}||T_A||\alpha\}||\text{Bob}||T_B||\beta\}$ to AC.

Step 3. After receiving the message from Bob, AC executes following sub-steps. Note that AC can easily divide the receiving message into six sub-message strings in sequence: Alice, T_A α Bob, T_B and β.

Step 3.1. AC checks the freshness of T_B. If it is fresh, it computes

$$K_{BS} = h(h(\text{Bob}||k)||T_B)$$

and uses this result as the key to decrypt β. Once AC retrieves $(\text{Alice}||T_B||r_B)$, it verifies whether the retrieved timestamp T_B equals the received fifth sub-message. Additionally, it checks if the decrypted identifier equals the first sub-message string received from Bob. If both they hold, AC then determines that Bob is a legal user and the person he wants to communicate with is Alice.

Step 3.2. Similarly, AC verifies T_A. If it is fresh, it then computes

$$K_{AS} = h(h(\text{Alice}||k)||T_A)$$

and

$$D_{K_{AS}}(\alpha) = (\text{Bob}||T_A||r_A).$$

If the decrypted T_A and identifier are the same as the second and the fourth sub-message strings, which were received from Bob, AC would be convinced that Alice is valid and the person she wants to communicate with is Bob.

Step 3.3. AC subsequently generates a random number r and computes

$$\beta' = E_{K_{BS}}(T_B + 1||r_A||r)$$

and

$$\alpha' = E_{K_{AS}}(T_A + 1||r_B||r).$$

Finally, AC sends $\{\beta'||\alpha'\}$ back to Bob.

Step 4. Upon receiving the message, Bob decrypts β' to obtain T_B+1, r_A, and r. He then checks if T_B+1 is fresh. If it is valid, Bob computes the common session key as

$$SK = h(r||r_A||r_B)$$

for this protocol run and the verification token

$$\Gamma = h(SK||(r_A + 1)).$$

Further, Bob transfers $\{\alpha'||\Gamma\}$ to Alice.

Step 5. Once Alice receives the message from Bob, she follows the same steps and computes

$$D_{K_{AS}}(\alpha') = (T_A + 1||r_B||r)$$

then checks whether the retrieved T_A+1 is fresh. If it is valid, Alice also can compute the common session key as

$$SK = h(r||r_A||r_B).$$

Subsequently, she uses this session key to verify whether

$$h(SK||(r_A + 1)) = \Gamma.$$

If it holds, Alice would be convinced that she and Bob have the same session key. Eventually, Alice computes and sends $\Gamma' = h(SK||(r_B + 1))$ to Bob to persuade him that the calculated session key is, in fact, correct.

4. Accuracy of BP-3PAKE by BAN Logic

Here, we adopt BAN logic [16, 17] for the demonstration of the accuracy of BP-3PAKE, which contains the correctness of mutual authentication and key distribution. In the BAN logic, typically P and Q refer to principals, X and Y are statements, and K ranges over the cryptographic key. First, we introduce the constructs of the BAN logic in Table 1.

In addition, we take advantage of following logical postulates in the BAN logic for our proofs.

Sight-projection: $\frac{P \triangleleft (X, Y)}{P \triangleleft X}$;

Freshness-propagation: $\frac{P|\equiv \#(X)}{P|\equiv \#(X, Y)}$;

Session Key: $\frac{P|\equiv \#(K),\ P|\equiv Q|\equiv X}{P|\equiv P \overset{K}{\longleftrightarrow} Q}$;

Message-meaning: $\frac{P|\equiv Q \overset{K}{\leftrightarrow} P,\ P \triangleleft \{X\}_K}{P|\equiv Q|\sim X}$ and

$$\frac{P| \equiv Q \overset{Y}{\Leftrightarrow} P,\ P \triangleleft \langle X \rangle_Y}{P| \equiv Q| \sim X};$$

Saying: $\frac{P|\equiv Q|\sim (X, Y)}{P|\equiv Q|\sim X}$;

Nonce-verification: $\frac{P|\equiv \#(X),\ P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$;

Jurisdiction: $\frac{P|\equiv Q|\Rightarrow X,\ P|\equiv Q|\equiv X}{P|\equiv X}$; and

Message-decryption: $\frac{P|\equiv Q \overset{K}{\leftrightarrow} P,\ P \triangleleft \{X\}_K}{P \triangleleft X}$.

With regard to the *Session Key* rule, it is an extended postulate of the BAN logic for the combination key [19], where X is a fundamental element of the combination key K.

Before beginning the proof, we expand our proposed protocol into the generic form as follows.

M_1: Alice $\rightarrow$ Bob:$\{Alice||T_A||E_{K_{AS}}(\text{Bob}||T_A||r_A)\}$

M_2: Bob $\rightarrow$ AC: $\{\{\text{Alice}||T_A||E_{K_{AS}}(\text{Bob}||T_A||r_A)\}||\text{Bob}||T_B||E_{K_{BS}}(\text{Alice}||T_B||r_B)\}$

M_3: AC $\rightarrow$ Bob: $\{E_{K_{BS}}(T_B + 1||r_A||r)||E_{K_{AS}}(T_A + 1||r_B||r)\}$

M_4: Bob $\rightarrow$ Alice: $\{E_{K_{AS}}(T_A + 1||r_B||r)||h(SK||(r_A + 1))\}$

M_5: Alice $\rightarrow$ Bob: $\{h(SK||(r_B + 1))\}$

Subsequently, for simplicity, we translate this generic form into the idealized form as follows.

I_1: Alice $\rightarrow$ Bob:$T_A,\ \{T_A,\ r_A\}_{K_{AS}}$

I_2: Bob $\rightarrow$ AC: $T_A,\ \{T_A, r_A\}_{K_{AS}},\ T_B,\ \{T_B, r_B\}_{K_{BS}}$

I_3: AC $\rightarrow$ Bob: $\{T_B,\ r_A,\ r\}_{K_{BS}},\ \{T_A,\ r_B,\ r\}_{K_{AS}}$

I_4: Bob $\rightarrow$ Alice: $\{T_A,\ r_B,\ r\}_{K_{AS}},\ \left\langle r_A,\ \text{Alice} \overset{\text{SK}}{\leftrightarrow} \text{Bob} \right\rangle_{\text{SK}}$

I_5: Alice $\rightarrow$ Bob: $\left\langle r_B, \text{Alice} \overset{\text{SK}}{\leftrightarrow} \text{Bob} \right\rangle_{\text{SK}}$

Here, for the hash function representation, we use $\langle X \rangle_Y$ to describe the result of hashing X with secret Y. In a cryptosystem, because a hash operation includes some secrets in general, we believe this is feasible.

Recalling once again, the BP-3PAKE system allows any two clients (e.g. Alice and Bob) in this system to communicate with each other through a trusted authentication center (AC). In other words, AC must help Alice and Bob make mutual authentication; Alice and Bob may firmly believe the validity of each other by authenticating AC; and Alice and Bob could coordinate a session key shared between them. Hence, we need to prove that BP-3PAKE must accomplish the following goals.

G_1: AC| $\equiv T_A$

G_2: AC| $\equiv T_B$

G_3: Bob| $\equiv$ AC| $\equiv T_B$

G_4: Alice| $\equiv$ AC| $\equiv T_A$

G_5: Alice| $\equiv$ Alice $\overset{\text{SK}}{\leftrightarrow}$ Bob

G_6: Bob| $\equiv$ Alice $\overset{\text{SK}}{\leftrightarrow}$ Bob

G_7: Alice| $\equiv$ Bob| $\equiv$ Alice $\overset{\text{SK}}{\leftrightarrow}$ Bob

G_8: Bob| $\equiv$ Alice| $\equiv$ Alice $\overset{\text{SK}}{\leftrightarrow}$ Bob

Table 1. The constructs of the BAN logic.

$P\| \equiv X$: P *believes* X
$P \triangleleft X$: P *receives* X
$P\| \sim X$: P once *said* X
$P\| \Rightarrow X$: P has *jurisdiction* over X
$\#(X)$:the formula X is *fresh*
$P \overset{K}{\longleftrightarrow} Q$: P and Q may use the *shared key* K to communicate
$P \overset{X}{\Leftrightarrow} Q$:the formula X is a *secret* known only to P and Q
$\{X\}_K$:the formula X encrypted under the key K
$\langle X \rangle_Y$: X combined with the formula Y; it is implied that Y be a secret

We can now proceed with the proofs of BP-3PAKE by following assumptions:

A_1: $AC| \equiv Alice| \Rightarrow T_A$

A_2: $AC| \equiv Bob| \Rightarrow T_B$

A_3: $Alice| \equiv AC| \equiv M_{AS}$

A_4: $Bob| \equiv AC| \equiv M_{BS}$

A_5: $AC| \equiv \#(T_A)$

A_6: $AC| \equiv \#(T_B)$

A_7: $Alice| \equiv \#(T_A)$

A_8: $Bob| \equiv \#(T_B)$

A_9: $Alice| \equiv \#(r_A)$

A_{10}: $Bob| \equiv \#(r_B)$

Note that we regard M_{AS} and M_{BS} as Alice and Bob's secrets shared with AC, respectively, after they register it, where $M_{AS} = h(\text{Alice}||k)$ and $M_{BS} = h(\text{Bob}||k)$.

Lemma 1. *The authentication center can authenticate all clients that cooperatively run BP-3PAKE.*

Proof: We consider that Alice and Bob want to cooperatively run the proposed protocol. We must infer the goals G_1 and G_2 to show that AC can authenticate them. Our reason is as follows:

F_1: AC *receives* T_B using I_2. (*Sight-projection* rule)

In BP-3PAKE, AC would temporarily be suggested that the received T_B was produced by Bob once it receives and verifies that T_B is fresh. Then AC can use this timestamp to authenticate Bob further. Hence, we can obtain the formula

F_2: AC *believes* that Bob *believes* T_B.

According to A_2, A_6, and I_2, we then can derive the following formulas for the goal G_2:

F_3: AC *believes* that K_{BS} is *fresh* using A_6. (*Freshness-propagation* rule)

Note that $K_{BS} = h(h(\text{Bob}||k)||T_B)$.

F_4: AC *believes* $AC \overset{K_{BS}}{\leftrightarrow} Bob$ using F_3 and F_2. (*Session Key* rule)

F_5: AC *receives* $\{T_B, r_B\}_{K_{BS}}$ using I_2. (*Sight-projection* rule)

F_6: AC *believes* that Bob *said* (T_B, r_B) using F_4 and F_5. (*Message-meaning* rule)

F_7: AC *believes* that Bob *said* T_B using F_6. (*Saying* rule)

F_8: AC *believes* that Bob *believes* T_B using A_6 and F_7. (*Nonce-verification* rule)

F_9: AC *believes* T_B using A_2 and F_8. (*Jurisdiction* rule)

Note that according to the formula F_8, it can be used to support the accuracy of formula F_2. Additionally, we can summarize that Bob can be authenticated by AC based on formula F_9. In terms of goal G_1, the following can be inferred:

F_{10}: AC *receives* T_A using I_2. (*Sight-projection* rule)

Based on the principles of BP-3PAKE, we have the formula

F_{11}: AC *believes* that Alice *believes* T_A.

The remaining inference is:

F_{12}: AC *believes* that K_{AS} is *fresh* using A_5. (*Freshness-propagation* rule)

Note that $K_{AS} = h(h(\text{Alice}||k)||T_A)$.

F_{13}: AC *believes* $AC \overset{K_{AS}}{\leftrightarrow} Alice$ using F_{12} and F_{11}. (*Session Key* rule)

F_{14}: AC *receives* $\{T_A, r_A\}_{K_{AS}}$ using I_2. (*Sight-projection* rule)

F_{15}: AC *believes* that Alice *said* (T_A, r_A) using F_{13} and F_{14}. (*Message-meaning* rule)

F_{16}: AC *believes* that Alice *said* T_A using F_{15}. (*Saying* rule)

F_{17}: AC *believes* that Alice *believes* T_A using A_5 and F_{16}. (*Nonce-verification* rule)

F_{18}: AC *believes* T_A using A_1 and F_{17}. (*Jurisdiction* rule)

Hence, Alice can be authenticated by AC.

Lemma 2. *All clients that run in BP-3PAKE can verify the validity of the authentication center. Precisely, they would believe that the authentication center has helped them to authenticate each other.*

Proof: We also consider that Alice and Bob want to communicate with each other through BP-3PAKE for simplicity. We deduce goals G_3 and G_4 to show that they can trust each other through validating AC. For goal G_3, we derive the following formulas:

F_{19}: Bob *receives* $\{T_B,\ r_A,\ r\}_{K_{BS}}$ using I_3. (*Sight-projection* rules)

F_{20}: Bob *believes* that K_{BS} is *fresh* using A_8. (*Freshness-propagation* rule)

F_{21}: Bob *believes* Bob $\overset{K_{BS}}{\leftrightarrow}$ AC using F_{20} and A_4. (*Session Key* rule)

F_{22}: Bob *believes* that AC *said* $(T_B,\ r_A,\ r)$ using F_{21} and F_{19}. (*Message-meaning* rule)

F_{23}: Bob *believes* that AC *said* T_B using F_{22}. (*Saying* rule)

F_{24}: Bob *believes* that AC *believes* T_B using A_8 and F_{23}. (*Nonce-verification* rule)

Hence, we summarize that Bob believes that AC has authenticated Alice and he can trust Alice. In the same way, G_4 can be inferred as follows:

F_{25}: Alice *receives* $\{T_A,\ r_B,\ r\}_{K_{AS}}$ using I_4. (*Sight-projection* rules)

F_{26}: Alice *believes* that K_{AS} is *fresh* using A_7. (*Freshness-propagation* rule)

F_{27}: Alice *believes* Alice $\overset{K_{AS}}{\leftrightarrow}$ AC using F_{26} and A_3. (*Session Key* rule)

F_{28}: Alice *believes* that AC *said* $(T_A,\ r_B,\ r)$ using F_{27} and F_{25}. (*Message-meaning* rule)

F_{29}: Alice *believes* that AC *said* T_A using F_{28}. (*Saying* rule)

F_{30}: Alice *believes* that AC *believes* T_A using A_7 and F_{29}. (*Nonce-verification* rule)

Lemma 3. *Any two clients can negotiate a common session key, SK. It implies that they have authenticated to each other.*

Proof: Again, we assume that Alice and Bob want to communicate with each other through BP-3PAKE for simplicity. In our proposed scheme, the session key is expanded as

$$\mathrm{SK} = h(r \| r_A \| r_B).$$

Obviously, in this session key, Alice and Bob each holds a part of secrets r_A and r_B, respectively. When they receive messages encrypted by AC, they can retrieve each other's secrets and AC's r by following a series of formulas.

In terms of Bob:

F_{31}: Bob *receives* $(T_B,\ r_A,\ r)$ using F_{21} and F_{19}. (*Message-decryption* rule)

F_{32}: Bob *receives* $(r_A,\ r)$ using F_{31}. (*Sight-projection* rule)

As evidenced in Lemma 2, once Bob authenticates AC, he must believe that AC has authenticated Alice. Hence, he should believe that Alice also believes the message which he retrieved by the formula F_{32}. The formula expressed as

F_{33}: Bob *believes* that Alice *believes* $(r_A,\ r)$.

In terms of Alice, we infer that

F_{34}: Alice *believes* that Bob *believes* $(r_B,\ r)$.

Then, we can deduce the following formulas for goals G_5 and G_6:

F_{35}: Alice *believes* that *SK* is *fresh* using A_9. (*Freshness-propagation* rule)

F_{36}: Alice *believes* Alice $\overset{\mathrm{SK}}{\leftrightarrow}$ Bob using F_{35} and F_{34}. (*Session Key* rule)

F_{37}: Bob *believes* that *SK* is *fresh* using A_{10}. (*Freshness-propagation* rule)

F_{38}: Bob *believes* Alice $\overset{\mathrm{SK}}{\leftrightarrow}$ Bob using F_{37} and F_{33}. (*Session Key* rule)

Trivially, we can interpret the formula F_{36}, that Alice believes that she has a secret *SK* shared with Bob. Hence, we can derive

F_{39}: Alice *believes* Alice $\overset{\mathrm{SK}}{\Leftrightarrow}$ Bob.

Similarly, we find

F_{40}: Bob *believes* Alice $\overset{\mathrm{SK}}{\Leftrightarrow}$ Bob.

Subsequently, we deduce the following formulas for goals G_7 and G_8:

F_{41}: Alice *receives* $\left\langle r_A,\ \text{Alice} \overset{\mathrm{SK}}{\leftrightarrow} \text{Bob} \right\rangle_{\mathrm{SK}}$ using I_4. (*Sight-projection* rules)

F_{42}: Alice *believes* that Bob *said* $(r_A,\ \text{Alice} \overset{\mathrm{SK}}{\leftrightarrow} \text{Bob})$ using F_{39} and F_{41}. (*Message-meaning* rules)

F_{43}: Alice *believes* that Bob *said* Alice $\overset{\mathrm{SK}}{\leftrightarrow}$ Bob using F_{42}. (*Saying* rule)

F_{44}: Alice *believes* that Bob *believes* Alice $\overset{\mathrm{SK}}{\leftrightarrow}$ Bob using F_{35} and F_{43}. (*Nonce-verification* rule)

F_{45}: Bob *believes* that Alice *said* $(r_B,\ \text{Alice} \overset{\mathrm{SK}}{\leftrightarrow} \text{Bob})$ using F_{40} and I_5. (*Message-meaning* rules)

F_{46}: Bob *believes* that Alice *said* Alice $\overset{\mathrm{SK}}{\leftrightarrow}$ Bob using F_{45}. (*Saying* rule)

F_{47}: Bob *believes* that Alice *believes* Alice $\overset{\mathrm{SK}}{\leftrightarrow}$ Bob using F_{37} and F_{46}. (*Nonce-verification* rule)

According to formulas F_{36} and F_{44}, we assume that Alice believes that she has a session key, *SK*, shared with Bob and Bob also believes such of *SK*. Similarly, based on formulas F_{38} and F_{47}, Bob would believe that he has a session key, *SK*, shared with Alice and Alice also believes such of *SK*. Consequently, we can summarize that Bob and Alice both believe that they have a session key, *SK*, shared with each other.

Theorem 1. *Any two clients can authenticate each other and share a session key by following BP-3PAKE procedures.*

Proof: As demonstrations of Lemmas 1 and 2, any two clients that cooperatively run BP-3PAKE can verify the validity of the authentication center, AC, if and only if AC is able to authenticate them. Because they believe the jurisdiction of the AC, they authenticate each other indirectly. Then these two clients can also negotiate a combined session key for themselves with their partial secrets. According to Lemma 3, the established session key can be retrieved and verified by all involved clients.

Therefore, any two clients can authenticate each other and share a common session key with help from the authentication center.

5. Security and Efficiency Analyses

Lu and Cao proposed a secure three-party key authenticated key exchange protocol in 2007 [9]. They asserted that their proposed scheme can achieve the requirements of general 3PAKE mechanisms. Unfortunately, we find that it is vulnerable to the off-line password guessing attacks and the impersonation attacks. We thus provide a novel version (BP-3PAKE) that can defend against malicious attacks and achieve higher efficiency. Here, we show that BP-3PAKE is able to confirm the essentials of general 3PAKE mechanisms as well as discuss performance in Subsections 5.1 and 5.2, respectively.

5.1. Security Considerations

In this subsection, we describe that BP-3PAKE is able to achieve the security requirements of general 3PAKE mechanisms [1, 3, 5, 7, 9].

5.1.1. Mutual Authentication

As mentioned in Section 4, we have conducted a formal proof for demonstrating the accuracy of BP-3PAKE and came to the conclusion that "*Any two clients can authenticate each other and share a session key by following BP-3PAKE procedures*". In BP-3PAKE, the mutual authentication is based on the message exchange and verification between clients and AC, where a temporary key shared between clients and AC encrypts the message.

According to the scenario in BP-3PAKE, Alice's temporary key shared with AC is $K_{AS} = h(h(\text{Alice}||k)||T_A)$ and Bob's temporary key shared with AC is $K_{BS} = h(h(\text{Bob}||k)||T_B)$. Since AC receives $E_{K_{BS}}(\text{Alice}||T_B||r_B)$ and $E_{K_{AS}}(\text{Bob}||T_A||r_A)$ from Bob and Alice in Step 2 of the Communication Phase for verification, Alice and Bob must have valid passwords pw_A and pw_B in order to retrieve $h(\text{Alice}||k)$ and $h(\text{Bob}||k)$ from δ_A and δ_B, respectively. That is, only the legal participant who possesses a valid smart card and the corresponding valid password can generate a temporary key shared with AC. By comparing the retrieved timestamp T_A and T_B with the received one, AC can authenticate the validity of Alice and Bob. In the same way, Alice and Bob are also able to authenticate AC by verifying the validity of $T_A + 1$ and $T_B + 1$. It is because only the true AC can generate valid K_{AS} and K_{BS} using its secret key, k.

5.1.2. Session Key Security

The session key for a protocol run is defined as $SK = h(r||r_A||r_B)$, which involves three random numbers r_A, r_B, and r chosen by Alice, Bob, and AC respectively. Furthermore, the deliveries of these random numbers are protected by the secure symmetric encryption system with the input of a 128-bit secret key. Only valid participants who possess the corresponding secret can decrypt the corresponding message and retrieve these random numbers. It is computationally infeasible for an attacker to figure out the session key without knowledge of the random numbers.

Moreover, we use a double check system to allow both communication sides (Alice and Bob) to confirm that their computed session key is equal to the other. For instance, in Steps 4 and 5 of the Communication Phase, Bob computes and sends the verification token $\Gamma = h(SK||(r_A + 1))$ to Alice. After Alice computes the session key SK', she could use it to compute and check whether $h(SK'||(r_A + 1))$ is equal to the received Γ. If it holds, she would believe that Bob also has the same session key. Similarly, Bob can also make sure that Alice has the same session key.

5.1.3. Resistance of Malicious Attacks

Here, we apply some scenarios to demonstrate that our scheme is able to prevent malicious attacks.

On-line/Off-line Password Guessing Attack

We assume that there exists an adversary, Eve, who wants to guess someone's password with a detectable on-line password guessing attack. If she guesses password pw_E to impersonate Alice or Bob, she must further send the guessed password to AC for the on-line correctness checking. However, this attempt must fail because the messages sent to AC are $\{\text{Alice}||T_A||E_{K_{AS}}(\text{Bob}||T_A||r_A)\}$ and $\{\text{Bob}||T_B||E_{K_{BS}}(\text{Alice}||T_B||r_B)\}$ and the messages AC returns to Alice and Bob are $E_{K_{AS}}(T_A + 1||r_B||r)$ and $E_{K_{BS}}(T_B + 1||r_A||r)$. All is fruitless for Eve to check whether the guessed password is correct or not. Hence, she cannot mount a detectable on-line password guessing attack on BP-3PAKE. Similarly, if Eve attempts to make an undetectable on-line password guessing attack by the same way, she also cannot succeed.

On the other hand, if Eve intercepts all communicating messages in the system and attempts to mount an off-line password guessing attack, she will still fail. The reason behind this is that all the transmission messages do not include any information referred to the password. Hence, we could assume that BP-3PAKE can withstand detectable/undetectable on-line password guessing attacks and off-line password guessing attacks.

Impersonation Attack

We assume that there exists a dishonest user, Eve, who attempts to impersonate Bob to fool Alice as described in Subsection 2.2.2. She must intercept the message transfers from Bob to AC and replace the part of this message, which Bob contributed, with $\{\text{Eve}||T_E||E_{K_{ES}}(\text{Alice}||T_E||r_E)\}$. Then she will send the message $\{\{\text{Alice}||T_A||E_{K_{AS}}(\text{Bob}||T_A||r_A)\}||\text{Eve}||T_E||E_{K_{ES}}(\text{Alice}||T_E||r_B)\}$ to AC for verification. Clearly, the verification must fail. It is because the message sent from Alice contains an encrypted user identifier whom Alice wants to communicate with. Once AC decrypts $E_{K_{AS}}(\text{Bob}||T_A||r_A)$ and retrieves the identifier "Bob," it

Table 2. Performance comparisons with related 3PAKE protocols.

Methods	Party	Operations					
		Asy	*Sym*	*Exp*	*Hash*	*XOR*	*Ran*
BP-3PAKE	Alice	0	2	0	5	1	1
	Bob	0	2	0	5	1	1
	AC	0	4	0	4	0	1
[8]	Alice	0	1	3	5	2	2
	Bob	0	1	3	5	2	2
	AC	0	2	4	6	4	1
[5]	Alice	2	3	2	1	0	2
	Bob	2	3	2	1	0	2
	AC	4	2	0	0	0	0
[11]	Alice	1	2	2	0	0	1
	Bob	1	2	2	0	0	1
	AC	2	0	4	0	0	3
[4]	Alice	0	1	3	6	0	1
	Bob	0	1	3	6	0	1
	AC	0	2	4	4	0	2
[7]	Alice	0	1	3	6	0	1
	Bob	0	1	3	6	0	1
	AC	0	2	4	4	0	2
[6]	Alice	1	2	2	0	0	2
	Bob	1	2	2	0	0	3
	AC	2	2	0	0	0	0

would find that this identifier is different from Eve's identifier. Consequently, AC must think that whom Alice wants to communicate with is not Eve but Bob and terminates the connection.

In addition, it is impossible for Eve to falsify Alice's encrypted message as $E_{K_{AS}}(\text{Eve}\|T_A\|r_A)$ for passing the verification without the knowledge of the temporary key K_{AS} and r_A. Actually, our proposed BP-3PAKE scheme can prevent impersonation attacks.

5.1.4. Perfect Forward/Backward Secrecy

A mechanism is said to possess perfect forward/backward secrecy if compromised long-term secrets will not lead to the revelation of previous/following session keys. In BP-3PAKE, the computation of the session key only consists of three random numbers chosen in each protocol run. If an intruder, Eve, gets a session key of a certain protocol run, she still cannot know any message communicated in other sessions since the session key in each session is different. Even if the long-term secret pw_A/pw_B is compromised by Eve somehow, it is helpless for her to construct previous or following session keys since each construction of a session key is contributed by three independent random numbers determined by the involved parties.

5.1.5. Known-Key Security

This is similar to the perfect forward secrecy, because the construction of each session key is based on three random numbers chosen by Alice, Bob, and AC respectively for each protocol run, compromising one session key will not cause the disclosure of other session keys.

5.2. Performance Comparisons

The following will discuss and compare the performance of BP-3PAKE with previous secure 3PAKE mechanisms. The performance comparisons with other related 3PAKE schemes are shown in Table 2. In this table, *Asy* is the asymmetric en/decryption; *Sym* represents the symmetric en/decryption; *Exp* refers to the modular exponentiation; *Hash* denotes the one-way hash function; *XOR* indicates the exclusive-OR operation; and *Ran* is the number of required random numbers involved in the system.

Since the computation overheads of performing an exclusive-OR operation and generating a random number are far lighter than for other operations, the performance estimation of a 3PAKE scheme mainly depends on the number of demanded asymmetric cryptography operations, symmetric cryptography operations, modular exponentiations, and hash operations.

As introduced in [20, 21], one asymmetric en/decryption is commensurate with 100 symmetric en/decryptions; one symmetric en/decryption is equal to 5/3 modular exponentiation; and one modular exponentiation is similar to performing one hash function 600 times for software consideration. Hence, the number of demanded asymmetric and symmetric en/decryptions dominates the performance evaluation among 3PAKE mechanisms. Accordingly, we can infer that the computation overheads of our scheme are less than those of related works. As illustrated in Table 2, the overall computation loads of our scheme is reduced by 80.01% compared with [4, 7, 8], by 0.99% compared with [5], and by 1.96% compared with [6, 11]. In particular, since we adopt smart card equipment, all computation costs for the user are endured by the smart cards. Users do not need to perform any complicated operations in person. That is, SC-3PEKE can effectively outperform related works.

6. Conclusions

In this paper, we have shown that S-3PAKE is vulnerable on off-line password guessing attacks and impersonation attacks. Further, we have proposed a novel smart-card-based three-party

key exchange protocol (BP-3PAKE). As discussed in Section 5, BP-3PAKE can overcome the security weakness from which S-3PAKE suffered and achieve the security requirements of general 3PAKE mechanisms. In addition, we also use BAN logic to demonstrate the accuracy of mutual authentication and key distribution in a BP-3PAKE system. It is worthwhile to note that by using BP-3PAKE to strengthen the security situation, the computation cost is decreased rather than increased. Our proposed scheme is indeed more practical in application than other related 3PAKE schemes.

References

[1] C. C. Chang and Y. F. Chang, "A novel three-party encrypted key exchange protocol," *Computer Standards & Interfaces*, vol. 26, no. 5, pp. 471–476, 2004.

[2] H. B. Chen, T. H. Chen, W. B. Lee, and C. C. Chang, "Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks," *Computer Standards & Interfaces*, vol. 30, pp. 95–99, Jan. 2008.

[3] W. S. Jaung, "Efficient three-party key exchange using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 619–624, May 2004.

[4] T. F. Lee, T. Hwang, and C. L. Lin, "Enhanced three-party encrypted key exchange without server public keys," *Computer & Security*, vol. 23, pp. 571–577, Oct. 2004.

[5] T. F. Lee, J. L. Liu, M. J. Sung, S. Yang, and C. M. Chen, "Communication-efficient three-party protocols for authentication and key agreement," *Computers & Mathematics with Applications*, vol. 58, pp. 641–648, Aug. 2009.

[6] C. L. Lin, H. M. Sun, and T. Hwang, "Three-party encrypted key exchange: attacks and a solution," *ACM Operating Systems Review*, vol. 34, pp. 12–20, Oct. 2000.

[7] C. L. Lin, H. M. Sun, M. Steiner, and T. Hwang, "Three-party encrypted key exchange without server public-keys," *IEEE Communications Letters*, vol. 5, pp. 497–499, Dec. 2001.

[8] N. W. Lo and K. H. Yeh, "Cryptanalysis of two three-party encrypted key exchange protocols," *Computer Standards & Interfaces*, vol. 31, pp. 1167–1174, Nov. 2009.

[9] R. Lu and Z. Cao, "Simple three-party key exchange protocol," *Computers & Security*, vol. 26, pp. 94–97, Feb. 2007.

[10] G. T. M. Steiner and M. Waidner, "Refinement and extension of encrypted exchange," *ACM Operating Systems Review*, vol. 29, pp. 22–30, Jul 1995.

[11] H. M. Sun, B. C. Chen, and T. Hwang, "Secure key agreement protocols for three-party against guessing attacks," *The Journal of System and Software*, vol. 75, pp. 63–68, Feb. 2005.

[12] E. J. Yoon and K. Y. Yoo, "Improving the novel three-party encrypted key exchange protocol," *Computer Standards & Interfaces*, vol. 30, pp. 309–314, Jul. 2008.

[13] S. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, (Oakland, California), pp. 72–84, May 1992.

[14] H. A. Wen, T. F. Lee, and T. Hwang, "Provably secure three-party password-based authenticated key exchange protocol using weil pairing," *IEE Proceedings of Communications*, vol. 152, pp. 138–143, Apr. 2005.

[15] J. Nam, Y. Lee, S. Kim, and D. Won, "Security weakness in a three-party pairing-based protocol for password authenticated key exchange," *Information Sciences*, vol. 177, no. 6, pp. 1364–1375, 2007.

[16] M. Burrows, M. Abadi, and R. Needham, "Authentication: a practical study in belief and action," in *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge*, (California, USA), pp. 325–342, Mar 1988.

[17] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," in *ACM Transactions on Computer Systems*, vol. 8, pp. 18–36, Feb 1990.

[18] W. Rankl and W. Effing, *Smart Card Handbook*. John Wiley and Sons, 2nd edition ed., 2000.

[19] S. P. Yang and X. Li, "Defect in protocol analysis with ban logic on man-in-the-middle sttacks," *Application Research of Computers*, vol. 24, pp. 149–151, Mar. 2007.

[20] Y. F. Chang, C. C. Chang, and Y. L. Liu, "Password authentication without the server public key," *IEICE Transactions on Communications*, vol. E87-B, no. 10, pp. 3088–3091, 2004.

[21] B. Schneier, *Applied Cryptography, Protocols, Algorithms, and Source Code in C*. 2nd edition ed.

Ting-Fang Cheng received the BS and MS degrees in information engineering and computer science from Feng Chia University, Taichung, Taiwan in 2005 and 2007, respectively. She is currently pursuing her Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, Taiwan. Her current research interests include electronic commerce, information security, cryptography, and mobile communications.

Professor Alan Chin-Chen Chang obtained his Ph.D. degree in computer engineering from Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in Computer and Decision Sciences. Both were awarded in Tsing Hua University. Dr. Chang served in Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in Chung Hsing University, chair professor in Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan.

Professor Chang's specialties include, but not limited to, data engineering, database systems, computer cryptography and information security. A researcher of acclaimed and distinguished services and contributions to his country and advancing human knowledge in the field of information science, Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Youth Award of Taiwan, Outstanding Talent in Information Sciences of Taiwan, AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of Taiwan, Outstanding Engineering Professor Award of Taiwan, Chung-Shan Academic Publication Awards, Distinguished Research Awards of National Science Council of Taiwan, Outstanding Scholarly Contribution Award of the International Institute for Advanced Studies in Systems Research

and Cybernetics, Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. He also published more than 1200 papers in Information Sciences. In the meantime, he participates actively in international academic organizations and performs advisory work to government agencies and academic organizations.

Zhi-Hui Wang received the BS degree in software engineering in 2004 from the North Eastern University, Shenyang, China, and the MS degree in software engineering in 2007 from the Dalian University of Technology, Dalian, China. She is currently pursuing her PhD degree in computer software and theory from the Dalian University of Technology, Dalian, China. Her research interests include data hiding, and image processing.