An Enhanced Three-Party Authentication Key Exchange Protocol for Mobile Commerce Environments

Zuowen Tan[1,2]
1. Department of Computer Science & Technology, School of Information Technology,
Jiangxi University of Finance and Economics, Nanchang City 330032,Jiangxi Province, P.R. China
2.Key Lab of Network Security and Cryptology, School of Mathematics and Computer Science,
Fujian Normal University, Fuzhou 350007,Fujian Province, P.R. China
tanzyw@gmail.com

***Abstract*—Recently, Yang et al. proposed a three-party encrypted key exchange protocol (3PAKE) which is based on Elliptic curve cryptography. Their 3PAKE protocol is efficient because it requires less computation cost and less communication cost, which is well suitable for mobile commerce environments. However, Yang et al.'s 3PAKE protocol is susceptible to parallel attacks and impersonation attacks. We presented an enhancement to resolve such security problems. Detailed analyses show that our proposed protocol is a secure 3PAKE protocol and more efficient.**

***Index Terms*—key exchange, unknown key-share attacks, impersonation attacks, authentication**

I. INTRODUCTION

Communication network has brought convenience to people. However, the communication channel could be eavesdropped and the message transmitted could be modified. Impersonation attacks could be mounted in the open environment. Bellovin and Merritt [1] developed a two-party password-based authentication key exchange (2PAKE) protocol in which party authentication and key exchange techniques always are adopted. Two parties in communication share a password, authenticate each other and obtain a common ephemeral session key[1]. Since then, many 2PAKE protocols are proposed [2,3].

However, 2PAKE protocols have the poor scalability. If 2PAKE protocols are applied in a multi-party environment, there must be the high maintenance cost problems. Because 2 PAKE protocols require each pair to share one password, in order to communicate with many parties, each party has to remember a larger number of passwords. Much research has been made to generalize 2PAKE protocols to 3PAKE protocols.

Manuscript received August 27, 2009; revised December 20, 2009; accepted March 8, 2010

3PAKE protocols can be classified into two categories: with password and without password. In a 3PAKE password-based protocol[4,5,7,8,11,13], every party shares only a single password with a trusted server which provides authentication services for the pair of parties, thus the parties can authenticate each other and share an authenticated session key. Only valid parties can decrypt message to derive correct session keys. In 3PAKE password-based protocols, each party does not need to remember and store multiple passwords. The other category of 3PAKE protocols don't use any password, but apply symmetric key cryptosystems such as DES, AES etc or public key cryptosystems [14,15,16]. In the second 3PAKE protocols, encryption [14,15,16] or signature [18] techniques are used as the authentication methods. Such 3PAKE protocols often lead to high computation cost. 3PAKE protocols can also be classified into two classes: without servers [18] and with a server. The former is a special case of multi-party key agreement protocols. In 3PAKE protocols with a server, two parties can cooperate to produce a common session key with the help of the server. In the following, the protocols to be discussed are 3PAKE protocols with a server.

A research direction in 3PAKE public key cryptosystem based protocols aims to improve the efficiency. Based upon Schnorr's digital signature scheme [17], Chen et al. [15] proposed a 3PAKE protocol with fewer rounds. But Chen et al.'s protocol still has the high computation cost and communication cost. Moreover, their protocol cannot resist against stolen-verifier attacks [16]. Yang et al. [16] use elliptic curve cryptography to present an enhancement to Chen et al.'s 3PAKE protocol. Their proposed protocol requires smaller transmitted message size and less communication times. But Yang et al.'s 3PAKE protocol suffers from unknown key-share attacks [19]. An improvement on it is proposed in [19]. However, the proposed protocol is not consistent to Yang et al.'s 3PAKE protocol. Because the proposed protocol applies password, smart card, and the

doi:10.4304/jcm.5.5.436-443

public key cryptosystem, it is not a 3PAKE protocol only based on public/secret key cryptosystem. In fact, the 3PAKE protocols with password authentication will suffer from some security threats. For example, weak passwords always incur the offline guessing attack, the online guessing attack and the online undetectable guessing attack [5,6-10,12].

Assume *A* and *B* are two honest entities and S is the server which help *A* and *B* to build a session key. In the 3PAKE protocol, the server is trustworthy. A 3PAKE protocol should satisfy the following security attributes [13,18,20,21]:

(1) *Known-Key Security*. After each execution of the 3PAKE protocol, *A* and *B* can generate a unique secret session key. Each session key of one execution of the 3PAKE protocol is independent of that one generated in another execution of the 3PAKE protocol. Moreover, the compromise of one session key should not lead to compromise of other session keys.

(2) *Forward Secrecy*. If secret keys of the three parties including the server are compromised, the secrecy of previously established session keys should not be affected.

(3) *Key-Compromise Impersonation Resilience*. Even if an adversary has corrupted one party, e.g. *A*, and obtained *A*'s secret key, the adversary still can not impersonate the other party, e.g. *B*, and communicate with a party *C*.

(4) *Unknown Key-Share Resilience*. After the protocol run, one party, say *A*, believes that she shares a key with a party, say *B*, but while *B* mistakenly believe that the key is shared with another party, say *C*. Therefore, a secure 3PAKE protocol should resist against the unknown key-share attacks.

(5) *Key Control*. The key should be determined jointly by both the parties *A* and *B*. Even the server cannot decide the session key.

In the paper, we show further analysis on the security of Yang et al.'s 3PAKE protocol. We found that an adversary can impersonate the session initiator to request the communication with other parties and can also impersonate the session responder to build the communication with the initiator. In addition, Yang et al.'s 3PAKE protocol suffers from parallel attacks.

To overcome those security weaknesses, we propose an enhanced 3PAKE protocol based on Yang et al.'s scheme. The proposed protocol using Elliptic curve cryptography (ECC) inherits the advantages of Yang et al.'s scheme. We integrated the time stamp and the identities of the sender into the hash function, the proposed protocol removes the security weaknesses of Yang et al.'s scheme. Detailed cryptanalysis demonstrates that our 3PAKE protocol can satisfy all the security properties which a secure 3PAKE protocol posses.

The rest of the paper is organized as follows. In Section 2, we review Yang et al.'s 3PAKE protocol using ECC for mobile-commerce environments. In Section 3, we analyze the security flaws of their protocol. In Section 4, an enhanced 3PAKE scheme is proposed. In Section 5, we analyze the security of the proposed 3PAKE protocol. Finally, conclusion will be given in Section 6.

II. REVIEW OF YANG ET AL.'S 3PAKE PROTOCOL

Now, we briefly review Yang et al.'s three-party authenticated key exchange protocol using ECC for mobile-commerce environments. Yang et al.'s 3PAKE protocol is divided into two phases: the initialization phase and the authenticated key exchange phase. And the protocol is involved with three roles: the party *A*, the party *B* and a trusted server *S*.

First, we introduce some notations used throughout the paper in Table 1.

Table 1 The notations of 3PAKE protocol

ID_x	The identity of the communication party x
p,q	Two large primes satisfying $q\|p$-1
g	An element of order q in F_q
x,y	The private/public key pair, $y \equiv g^x \bmod p$
T_X	The time stamp of the party x

In the initialization phase, the server *S* initializes and selects some parameters. Both *A* and *B* register to *S*. The system parameters includes a finite field F_q over a large prime q and an elliptic curve group by an order n point Q over the curve $E_q(a,b): y^2 \equiv x^3 + ax + b \pmod q$, where $a,b \in F_q$ and $4a^3 + 27b^2 \neq 0 \pmod q$.Let $E_k(\cdot)/D_k(\cdot)$ be a symmetric encryption/decryption algorithm, where k is the symmetric key.

In the registration phase, the parties *A* and *B* register to the server *S* to generate their private/public key pairs d_A/U_A and d_B/U_B, where $U_A = d_A Q$, $U_B = d_B Q$, and $d_A, d_B \in Z_n^*$. The server chooses its private key $d_S \in Z_n^*$ and computes its public key $U_S = d_S Q$.

The authenticated key exchange phase can be depicted as follows.

R1 $A \rightarrow B$: $\{ID_A, Request\}$

A: $r_A \in Z_q^*, w_A \in Z_q^*$

$R_A = r_A U_A, \widehat{R}_A = r_A U_s$

$K_A = d_A \widehat{R}_A = (k_{Ax}, k_{Ay})$

$W_A = w_A Q, C_A = E_{K_{Ax}}(R_A, W_A)$

$A \rightarrow S$: $\{ID_A, ID_B, C_A, R_A\}$

R2 $B \rightarrow A$: $\{ID_B, Response\}$

B: $r_B \in Z_q^*, w_B \in Z_q^*$

$R_B = r_B U_B$, $\widehat{R}_B = r_B U_s$

$K_B = d_B \widehat{R}_B = (k_{Bx}, k_{By})$

$W_B = w_B Q$, $C_B = E_{K_{Bx}}(R_B, W_B)$

$B \rightarrow S$: { ID_B,ID_A, C_B, R_B }

R3 *S*: $K_A = d_S R_A = (k_{Ax}, k_{Ay})$

$K_B = d_S R_B = (k_{Bx}, k_{By})$

$(R_A, W_A) = D_{K_{Ax}}(C_A)$

$(R_B, W_B) = D_{K_{Bx}}(C_B)$

Check: received R_A =? decrypted R_A

Check: received R_B =? decrypted R_B

$C_{SA} = E_{K_{Ax}}(R_A, W_B)$, $C_{SB} = E_{K_{Bx}}(R_B, W_A)$

S $\rightarrow$*A*: { C_{SA} }

S $\rightarrow$*B*: { C_{SB} }

A: $(R_A, W_B) = D_{K_{Ax}}(C_{SA})$

Check: selected R_A =? decrypted R_A

$SK = w_A W_B$

B: $(R_B, W_A) = D_{K_{Bx}}(C_{SB})$

Check: selected R_B =? decrypted R_B

$SK = w_B W_A$.

III. WEAKNESSES OF YANG ET AL.'S 3PAKE PROTOCOL

Yang et al claimed that their scheme [19] is secure. However, we show that Yang et al.'s 3PAKE protocol still suffers from some attacks.

A. Impersonation-of-initiator attacks

Any adversary *C* can impersonate *A* to request the communication with *B*. The initialization phase is the same as that in Yang et al.'s 3PAKE protocol. The authenticated key exchange phase with *C* can be described as follows.

R1 *C* $\rightarrow$*B*: {ID_A, *Request* }

C: $r_A \in Z_q^*, w_A \in Z_q^*$

$R_A = r_A Q$, $K_A = r_A U_S = (k_{Ax}, k_{Ay})$

$W_A = w_A Q$, $C_A = E_{K_{Ax}}(R_A, W_A)$

C $\rightarrow$*S*: {ID_A, ID_B, C_A, R_A}

R2 *B* $\rightarrow$*C*: {ID_B, *Response*}

B: $r_B \in Z_q^*, w_B \in Z_q^*$

$R_B = r_B U_B$, $\widehat{R}_B = r_B U_s$

$K_B = d_B \widehat{R}_B = (k_{Bx}, k_{By})$

$W_B = w_B Q$, $C_B = E_{K_{Bx}}(R_B, W_B)$

B $\rightarrow$*S*: { ID_B, ID_A, C_B, R_B }

R3 *S*: $K_A = d_S R_A = (k_{Ax}, k_{Ay})$

$K_B = d_S R_B = (k_{Bx}, k_{By})$

$(R_A, W_A) = D_{K_{Ax}}(C_A)$

$(R_B, W_B) = D_{K_{Bx}}(C_B)$

Check: received R_A =? decrypted R_A

Check: received R_B =? decrypted R_B

$C_{SA} = E_{K_{Ax}}(R_A, W_B)$

$C_{SB} = E_{K_{Bx}}(R_B, W_A)$

S $\rightarrow$*C*: { C_{SA} }

S $\rightarrow$*B*: { C_{SB} }

C: $(R_A, W_B) = D_{K_{Ax}}(C_{SA})$

Check: selected R_A =? decrypted R_A

$SK = w_A W_B$

B: $(R_B, W_A) = D_{K_{Bx}}(C_{SB})$

Check: selected R_B =? decrypted R_B

$SK = w_B W_A$.

Finally, *B* will mistake *C* for *A* and communicate with *C* by using the session key *SK*.

B. Impersonation-of-responder attacks

Any adversary *E* can also impersonate *B* to accomplish the session key exchange with *A*. During the authenticated key exchange phase, *E* impersonates the party *B* to share a session key with the party *A*. The whole phase is composed of three rounds.

R1 *A* $\rightarrow$*B*: {ID_A, *Request* }

A: $r_A \in Z_q^*, w_A \in Z_q^*$

$R_A = r_A Q$, $K_A = r_A U_S = (k_{Ax}, k_{Ay})$

$W_A = w_A Q$, $C_A = E_{K_{Ax}}(R_A, W_A)$

A $\rightarrow$*S*: {ID_A, ID_B, C_A, R_A}

Adversary *E* intercepts the message (ID_A,*Request*)

R2 *E* $\rightarrow$*A*: {ID_B, *Response*}

E: $r_B \in Z_q^*, w_B \in Z_q^*$

$R_B = r_B Q$, $K_B = r_B U_S = (k_{Bx}, k_{By})$

$W_B = w_B Q$, $C_B = E_{K_{Bx}}(R_B, W_B)$

E $\rightarrow$*S*: {ID_B, ID_A, C_B, R_B }

R3 *S*: $K_A = d_S R_A = (k_{Ax}, k_{Ay})$

$K_B = d_S R_B = (k_{Bx}, k_{By})$

$(R_A, W_A) = D_{K_{Ax}}(C_A)$

$(R_B, W_B) = D_{K_{Bx}}(C_B)$

Check: received R_A =? decrypted R_A

Check: received R_B =? decrypted R_B

$C_{SA} = E_{K_{Ax}}(R_A, W_B)$, $C_{SB} = E_{K_{Bx}}(R_B, W_A)$

S $\rightarrow$*A*: { C_{SA} }

$S \rightarrow E$: $\{ C_{SB} \}$

A: $(R_A, W_B) = D_{K_{Ax}}(C_{SA})$

Check: selected R_A =? decrypted R_A

$SK = w_A W_B$

E: $(R_B, W_A) = D_{K_{Bx}}(C_{SB})$

Check: selected R_B =? decrypted R_B

$SK = w_B W_A$.

Thus, *A* will mistake *C* for *B* and communicate with *C* by using the session key *SK*.

C. Parallel attacks

Suppose that the adversary *C* monitors the communication channel between *A* and *S* and that communication channel between *B* and *S*. When *A* and *B* have finished 3PAKE protocol runs, the adversary intercepts the message flow (ID_A, ID_B, C_A, R_A) and (ID_B, ID_A, C_B, R_B) sent to *S*. *C* could mount the following attack.

When the party *A* tries a new communication with the party *B*, the adversary immediately replays (ID_A, ID_B, C_A, R_A) to *S*. *S* can verify the identity of the party *A*. Moreover, *S* will confirm that the adversary is *A* and *A* attempts to communicate with *B*. The server *S* will continue to execute the protocol. When *B* receives the message {ID_A, *NewRequest*}, *B* chooses r'_B, $w'_B \in Z_q^*$ in random and computes R'_B, W'_B and C'_B. *B* sends (ID_B, ID_A, C'_B, R'_B) to *S*. Finally *S* computes and sends $C'_{SB} = E_{K'_{Bx}}(R'_B, W_A)$ to the party *B*. *B* computes $w'_B W_A$ as the session key. However, *A* sends (ID_A, ID_B, C'_A, R'_A) to *S* and finally computes the session key $w'_A W'_B$. Thus, *A* and *B* have different session keys.

Likewise, the adversary can also replay *B*'s response. If the party *A* tries a new communication with the party *B* and send {ID_A, *NewRequest*} to B, the adversary makes a response {ID_B, *NewResponse*} to *A* and immediately replays (ID_B, ID_A, C_B, R_B) to *S*. *S* verifies the identity of the party *B*. *S* believes that the adversary is *B* and *A* attempts to communicate with *B*. The server *S* continues the protocol. *A* chooses r'_A, $w'_A \in Z_q^*$ in random and computes R'_A, W'_A and C'_A. *A* sends (ID_B, ID_A, C'_B, R'_B) to *S*. Finally *S* computes and sends $C'_{SA} = E_{K'_{Ax}}(R'_A, W_B)$ to the party *A*. *A* computes $w'_A W_B$ as the session key. *B* sends (ID_B, ID_A, C'_B, R'_B) to *S* and finally computes the session key $w'_B W'_A$. So, *A* and *B* have different session keys.

In addition, if the adversary has already intercepted many enough communication message flows, the adversary can require more enough responses from the server by replay attacks. Thus, the server will be clogged by the seemingly legitimate requests. In fact, the legal parties cannot build up a session key in time without the server's help.

IV. The Enhanced 3PAKE Protocol Using ECC

To overcome the security flaws of Yang et al.'s protocol [19], we propose an improved 3PAKE protocol. The enhanced 3PAKE protocol concerns three parties: party *A*, party *B* and server *S*. The protocol is composed of two phases: the initialization phase and the authenticated key exchange phase.

The initialization phase is the similar to that one in Yang et al.'s protocol. But, all the parties' public keys are built in PKI. The parties *A* and *B* need not register to the server. Here, we omit the detailed description of the initialization phase. The authenticated key exchange phase still consists of three rounds.

Round 1

A executes the following steps.

Step 1. Select a random integer $r_A \in Z_q^*$ and compute

$$R_A = r_A U_A .$$

Step 2. Compute the key

$$K_A = r_A d_A U_S = (k_{Ax}, k_{Ay}) .$$

Step 3. Select a random $w_A \in Z_q^*$ and compute

$$W_A = w_A Q .$$

Step 4. Determine the time T_A and encrypt

$$C_{AS} = E_{K_{Ax}}(R_A, W_A, ID_A, ID_B, T_A) .$$

Step 5. Send (ID_A,*Request*) and (ID_A, C_{AS}, R_A) to *B* and *S*, respectively. The message *Request* denotes a request that *A* asks *B* to share a session key.

Round 2

After *B* receives the message (ID_A, *Request*), *B* performs the following steps.

Step 1. Select a random integer $r_B \in Z_q^*$ and compute

$$R_B = r_B U_B .$$

Step 2. Compute the key

$$K_B = r_B d_B U_S = (k_{Bx}, k_{By}) .$$

Step 3. Select a random $w_B \in Z_q^*$ and compute

$$W_B = w_B Q .$$

Step 4. Determine the time T_B and encrypt

$$C_{BS} = E_{K_{Bx}}(R_B, W_B, ID_B, ID_A, T_B) .$$

Step 5. Send (ID_B, *Response*) and (ID_B, C_{BS}, R_B) to *B* and *S*, respectively. The message *Response* denotes a response that *B* accepts *A*'s request.

Round 3

After *S* receives the message (ID_A, C_{AS}, R_A) and (ID_B, C_{BS}, R_B), *S* performs the following steps.

Step 1. Check if the time stamp T_A and T_B are valid. If they are valid, *S* computes the two keys

$$K_A = d_S R_A = (k_{Ax}, k_{Ay}),$$
$$K_B = d_S R_B = (k_{Bx}, k_{By}).$$

Step 2. Use k_{Ax} and k_{Bx} as the decryption key to decrypt the two cipher texts

$$(R_A, W_A, ID_A, ID_B, T_A) = d_{K_{Ax}}(C_{AS}),$$
$$(R_B, W_B, ID_B, ID_A, T_B) = d_{K_{Bx}}(C_{BS}).$$

Step 3. Check if the decrypted T_A and T_B are the same as the received T_A and T_B, respectively. And *S* checks if the decrypted ID_A and ID_B are the same as the received ID_A and ID_B, respectively.

Step 4.Check if the decrypted message R_A is valid. If it is invalid, the server stops the protocol and sends an authenticated-failure message to *B*. Then, *S* checks if the decrypted message R_B is valid. If it is invalid, the server stops the protocol and sends an authenticated-failure message to *A*. When *A* and *B* are both valid parties, *S* determines the time T_S and uses k_{Ax} and k_{Bx} to encrypt

$$C_{SA} = E_{K_{Ax}}(R_A, W_B, ID_A, T_S, ID_S),$$
$$C_{SB} = E_{K_{Bx}}(R_B, W_A, ID_B, T_S, ID_S).$$

Step 5. Send C_{SA} and C_{SB} to *A* and *B*, respectively.

After *A* receives C_{SA}, *A* performs the following steps to accomplish the session key exchange.

Step A-1. Decrypt C_{SA} and obtain

$$(R_A, W_B, ID_A, T_S, ID_S) = D_{K_{Ax}}(C_{SA}).$$

Step A-2. Check if T_S is valid and the decrypted R_A is the same as the selected R_A in Round 1. If they are both the same and the identity message ID_S is valid, *A* confirms that *B* has been authenticated by *S*. *A* computes the session key $SK = w_A W_B$. Otherwise, *A* rejects the transaction.

Similarly, after *B* receives C_{SB}, *B* performs the following steps to accomplish the session key exchange.

Step B-1. Decrypt C_{SB} and obtain

$$(R_B, W_A, ID_B, T_S, ID_S) = D_{K_{Bx}}(C_{SB}).$$

Step B-2. Check if T_S is valid and the decrypted R_B is the same as the selected R_B in Round 2. If they are the same and the identity message ID_S is valid, *B* confirms that *A* has been authenticated by *S*. Finally, *B* computes the session key $SK = w_B W_A$. Otherwise, *B* rejects the transaction.

The authenticated key exchange phase can be depicted as follows.

R1 *A* →*B*: {ID_A, *Request* }

A: $r_A \in Z_q^*, w_A \in Z_q^*, R_A = r_A U_A,$
$K_A = r_A d_A U_S = (k_{Ax}, k_{Ay})$
$W_A = w_A Q$
$C_{AS} = E_{K_{Ax}}(R_A, W_A, ID_A, ID_B, T_A)$

A →*S*: { ID_A, C_{AS}, R_A}

R2 *B* →*A*: {ID_B, *Response*}

B: $r_B \in Z_q^*, w_B \in Z_q^*$
$K_B = r_B d_B U_S = (k_{Bx}, k_{By})$
$W_B = w_B Q$
$C_{BS} = E_{K_{Bx}}(R_B, W_B, ID_B, ID_A, T_B)$

B →*S*: { ID_B, C_{BS}, R_B }

R3 *S*: check if T_A and T_B are valid
$K_A = d_S R_A = (k_{Ax}, k_{Ay})$
$K_B = d_S R_B = (k_{Bx}, k_{By})$
$(R_A, W_A, ID_A, ID_B, T_A) = D_{K_{Ax}}(C_{AS})$
$(R_B, W_B, ID_B, ID_A, T_B R_B, W_B) = D_{K_{Bx}}(C_B)$
Check: received ID_A =? decrypted ID_A
received ID_B =? decrypted ID_B
received R_A =? decrypted R_A
received R_B =? decrypted R_B
$C_{SA} = E_{K_{Ax}}(R_A, W_B, ID_A, T_S, ID_S)$
$C_{SB} = E_{K_{Bx}}(R_B, W_A, ID_B, T_S, ID_S)$

S →*A*: { C_{SA} }

S →*B*: { C_{SB} }

A: $(R_A, W_B, ID_A, T_S, ID_S) = D_{K_{Ax}}(C_{SA})$
Check if T_S is valid
selected R_A =? decrypted R_A
$SK = w_A W_B$

B: $(R_B, W_A, ID_B, T_S, ID_S) = D_{K_{Bx}}(C_{SB})$
Check if T_S is valid
selected R_B =? decrypted R_B
$SK = w_B W_A$

V. Performance and Security Analyses

In this section, we give the performance and the security analyses of the proposed 3PAKE protocol.

A. Security analyses

We analyze the security of the enhanced 3PAKE scheme. The enhanced version inherits the security properties of Yang et al.'s 3PAKE protocol [19]. The proposed scheme is secure against man-in-the-middle attack, outsider attack and stolen-verifier attack. For the detailed analysis, see [16].

In the following, we first show the enhanced protocol can resist against the attacks in Section 4 and removes the security weaknesses of Yang et al.'s protocol.

(1)*Resistance to the impersonation-of-initiator attack*

Suppose that an adversary *C* impersonates *A* to request the communication with *B*. As in Section 3.1, *C* selects a random integer $r_A \in Z_q^*$ and computes $R_A = r_A U_A$. Upon the assumption of computational Diffie Hellman, *C* cannot compute the right secret key $K_A = r_A d_A U_S = (k_{Ax}, k_{Ay})$ without the knowledge of *A*'s secret key d_A or the server *S*'s secret key d_S. *C* has to choose a random integer as the secret key k_{Ax} between *A* and *S* and uses it to compute C_{AS}. Next, *C* sends the message (ID_A, C_{AS}, R_A) to *S*. Upon receiving the message, *S* first computes k'_{Ax} through $K'_A = d_S R_A = (k'_{Ax}, k'_{Ay})$ and then uses k'_{Ax} as the decryption key to compute the message

$$(C_A, ID_A, ID_B, T_A) = d_{K'_{Ax}}(C_{AS}).$$

Since *S* obtains a different decryption key k'_{Ax} from the encryption key k_{Ax} with the probability (1-1/q), *S* will find that the decrypted T_A is different from the received T_A and the decrypted ID_A is different from the received ID_A. Thus, *S* confirms that the initiator is not *A*.

Therefore, our proposed protocol can resist against the impersonation-of-initiator attack.

(2) *Resistance to the impersonation-of-responder attack*

Suppose that an adversary *C* impersonates *B* to respond with *A*. As mentioned in Section 3.1, although *C* can compute $R_B = r_B U_B$, *C* is unable to calculate the right secret key k_B through $K_B = r_B d_B U_S = (k_{Bx}, k_{By})$ without the knowledge of *B*'s secret key d_B and *S*'s secret key d_S. *C* has to choose a random integer as the encryption secret key k_{Bx} and produces C_{BS}. When *S* receives the message (ID_B, C_{BS}, R_B) from *C*, *S* computes the decryption key k'_{Bx} through $K'_B = d_S U_B = (k'_{Bx}, k'_{By})$ and computes the message

$$(C_B, ID_B, ID_A, T_B) = D_{K'_{Bx}}(C_{BS}).$$

During the decryption, *S* uses a different decryption key k'_{Bx} from *C*'s encryption key k_{Bx} about C_{BS} with the probability (1-1/q), so *S* will find that the decrypted T_B is different from the received T_B and the decrypted ID_B is different from the received ID_B. Thus, *S* confirms that the responder is not *B*.

Therefore, it is impossible to perform the impersonation-of-responder attack on our enhanced protocol.

(3) *Resistance to parallel attacks*

Assume that an adversary collects the information once being transferred between the parties and the server. Suppose that the adversary pretends *A* to replay the initiation request (ID_A, *Request*) and (ID_A, C_{AS}, R_A) with a fresh time stamp T_A' to *B* and *S*, respectively. When *S* uses k_{Ax} to decrypt the cipher text from the adversary, *S* can obtain the message

$$(C_A, ID_A, ID_B, T_A) = D_{K_{Ax}}(C_{AS}).$$

However, the decrypted time stamp T_A is different from the received time stamp T_A'. Thus, *S* can confirm that the initiation request from the adversary is not valid. So, the replay attacks as an initiator intending to fool the server can be detected.

Likewise, assume that the adversary tries to replay *B*'s response. A similar analysis demonstrates that *S* can confirm that the respond from the adversary is not valid. So, the replay attacks as a responder intending to fool the server can also be detected.

Therefore, the replay attack is infeasible for the enhanced 3PAKE scheme.

Suppose that the adversary *C* intercepts the message (ID_A, C_{AS}, R_A) and (ID_B, C_{BS}, R_B) to *S*. *C* sends C_{AS} and C_{BS} to *A* and *B*, respectively. According to Round 3 of our protocol, *A* decrypts C_{AS} and tries to get $(R_A, W_B, ID_A, T_S, ID_S)$ from $C_{AS} = E_{K_{Ax}}(C_A, ID_A, ID_B, T_A)$. Since *A* uses a different key K'_{Ax} from K_{Ax} to decrypt the cipher text C_{AS} which is encrypted by the secret key K_{Ax}, *A* will obtain a string without meaning. Moreover, C_{AS} is generated from (C_A, ID_A, ID_B, T_A) and is not from $(R_A, W_B, ID_A, T_S, ID_S)$. It is impossible for *A* to obtain the plaintext (ID_A, T_S, ID_S). Thus, *A* can also affirm that C_{SA} is not from the server *S*.

As for the party *B*'s case, we can make a similar detailed analysis as above-mentioned. Moreover, *B* can also affirm that (ID_S, C_{BS}) is not from the server *S*.

Next, we show that the proposed 3PAKE protocol holds the following security properties:

(1) *Known-Key Security*. Since in our 3PAKE protocol, the session key *SK* depends on the secret

random integers w_A and w_B, which are distributed uniformly in Z_q^*. So the session keys are also distributed uniformly. Compromise of one session key in one session will not affect other session keys in other sessions.

(2) *Forward Secrecy*. In our proposed protocol, even though the long-lived key d_A or d_B or both the keys are concealed, the session keys *SK* can not be computed. This is because the random elements W_A and W_B are encrypted through the symmetric key k_{Ax} and k_{Bx}, respectively. k_{Ax} and k_{Bx} only can be computed by $K_A = r_A d_A U_S$ or $K_A = d_S R_A$ and $K_B = r_B d_B U_S$ or $K_B = d_S R_B$. If one attempts to find r_A from R_A or r_B from R_B, he will be faced with Elliptic Curve Discrete Logarithm Problem (ECDLP). Moreover, suppose that both W_A and W_B are comprised, if one attempts to compute the session key $SK = w_B W_A$ from them, he will have to solve a computational Diffie Hellman problem (CDHP). Therefore, even if *A*'s, *B*'s and *S*'s secret keys are compromised, the secrecy of the session keys could not be computed.

(3) *Key-Compromise Impersonation Resilience*. If an adversary has corrupted one party, e.g. *A*, and obtained *A*'s secret key, the adversary would impersonate the other party, e.g. B. The adversary must produce a valid key $K_B = r_B d_B U_S = (k_{Bx}, k_{By})$ with its chosen $r_B \in Z_q^*$. But given (U_B, U_S), to compute $(r_B)^{-1} K_B$ further K_B is a computational Diffie Hellman problem. The adversary will fail in computing the right encryption key K_B.

(4) *Unknown Key-Share Resilience*. Because the transmitted ciphers include the two parties' identities and the proposed scheme is secure against *Key-Compromise Impersonation attacks*, after the protocol run, *A* believes she shares a key with the party *B*, while *B* also believes that the key is shared with the party *A*. Therefore, the proposed 3PAKE protocol can resist against the unknown key-share attacks.

(5) *Key Control*. As shown in the analysis of *Forward Secrecy*, the session key is determined jointly by both the parties A and B. In our 3PAKE protocol, the session key SK can be computed as $SK = w_B W_A$ or $SK = w_A W_B$. If one wants to know SK, one must know w_A or w_B. Even though the server can obtain W_A and W_B, the server can not compute the secret random integers w_A or w_B on the assumption of ECDLP.

We summarize the functionality of the proposed scheme and make comparisons with Yang et al.'s protocol in Table 2. It demonstrates that our schemes can achieve the essential requirements for 3PAKE.

B. Performance analyses

Compared with other 3PAKE protocols in the literature, Yang et al.'s protocol [19] has less computation costs and is efficient. If the size of *q* used in the ECC of the protocol is 160 bits. The cipher text size of the symmetric encryption/decryption AES is 128 bits. And the identity size if 80bit. Then, the total message size of Yang et al.'s protocol is 1152 bits (In [19], the message size of each party's identity is not concerned). However, in our protocol, the message flow from the party *A*(*B*) to the server *S* does not include the identity ID_B(ID_A) and the message from the server to the party *A*(*B*) does not include the identity ID_A (ID_B). Therefore, the transmitted message size is reduced to 832 bits.

The computation times of the proposed protocol is the same as that of Yang et al.'s protocol. If we ignore the computation costs of symmetric encryption and hash function, the total computation costs of *A* and *B* are 5PM, where PM means point multiplication. The total computation costs of the server *S* are 2PM.

From the above (also see Table 3), the proposed 3PAKE protocol is more efficient.

Besides, it is claimed that the server must store many public keys of the parties [19] in such 3PAKE protocols as [16]. However, it is not true in the proposed 3PAKE protocol. Since every party holds its public key certificate in PKI and does not register to the server, the server can obtain the information of the parties' public key from PKI which will not increase the server's workload.

Table 2 The functionality comparisons of the proposed protocol and Yang et al.'s protocol.

Resistance to attacks	Yang et al.'s protocol.	enhanced protocol
Man-in-the middle	Yes	**Yes**
Outsider attacks	Yes	**Yes**
Stolen-verifier attacks	Yes	**Yes**
Impersonation-of-initiator attack	No	**Yes**
Impersonation-of-responder attack	No	**Yes**
Parallel attacks	No	**Yes**
3PAKE's security attributes	**No Provided**	**Provided**

Table 3 The performance comparisons of the proposed protocol and Yang et al.'s protocol.

	The server stores public keys	Message sizes	A(B)'s computation costs	S's computation costs
Proposed protocol	No	832bits	5PM + 2SE	2PM + 4SE
Protocol in[28]	Yes	1152	5PM + 2SE	2PM + 4SE

PM, point multiplication; SE, symmetric encryption/decryption.

VI. CONCLUSION

In this paper, we have shown that Yang et al.'s protocol is vulnerable to the impersonate attacks and parallel attacks. We propose an enhanced three party key exchange protocol based on elliptic curve discrete logarithm problem. We introduce the time stamp to keep the authentication session key exchange fresh. The improved scheme removes the weakness of Yang et al.'s protocol. The analyses show that the proposed protocol is secure on the assumption of CDHP and ECDLP. In addition, the enhanced protocol is more efficient than Yang et al.'s protocol.

ACKNOWLEDGMENT

The author would like to thank the reviewers for their useful suggestions. This work was supported in part by a grant from the National Natural Science Foundation of China (10701040).

REFERENCES

[1] S. Bellovin, M. Merritt, "Encrypted key exchange: passwords based protocols secure against dictionary attacks", *Proceedings of the IEEE Symposium on Security and Privacy'92*, 1992, pp.72-84.

[2] M. Bellare, P. Rogaway, "Entity authentication and key distribution", *Advances in Cryptology- Crypto'93*, LNCS, vol. 773, 1993, pp.232-249.

[3] M. Bellare, D. Pointcheval, P. Rogaway, "Authenticated key exchange secure against dictionary attacks", *Advances in Cryptology-Eurocrypt'00*, LNCS, vol. 1807, 2000, pp. 139-155.

[4] M. Bellare, P. Rogaway, "Provably secure session key distribution: the three party case", *Proceedings of the ACM Symposium on the Theory of Computing* (STOC'95), 1995, pp.57-66.

[5] Ding, Y. and Horster, P., "Undetectable on-line password guessing attacks". *ACM Oper. Syst. Rev*., 29, 1995, pp.77–86.

[6] H. Guo, Z. Li, Y. Mu, X. Zhang, "Cryptanalysis of simple three-party key exchange protocol", *Computers & Security*, Vol. 27(1-2), 2008, pp.16-21.

[7] H. B. Chen, T. H. Chen, W. B. Lee, 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, No. 1-2, January 2008, pp.95-99.

[8] Lin, C.L., Sun, H.M. and Hwang,T. "Three party-encrypted key exchange: attacks and a solution". *ACM Oper. Syst. Rev*., 34, 2000, pp.12–20.

[9] Sun, H.M., Chen, B.C. and Hwang, T. "Secure key agreement protocols for three-party against guessing attacks". *J. Syst. Softw*., 75, 2005, pp.63–68.

[10] Chung, H.R. and Ku,W.C. "Three weaknesses in a simple three-party key exchange protocol". *Inf. Sci.*, 178, 2008, pp.220–229.

[11] Lu, R.,Cao, Z. "Simple three-party key exchange protocol". *Comput. Secur*., 26, 2007, pp.94–97.

[12] Nam, J., Kim, S. and Won, D. "Attack on the Sun–Chen–Hwang's three-party key agreement protocols using passwords". *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E89-A, 2006, pp.209–212.

[13] Hung-Yu Chien and Tzong-Chen Wu, "Provably secure password-based three-party key exchange With Optimal Message Steps", *The Computer Journal*, 2008, pp.1-10.

[14] Lin C-L, Sun H-M, Steiner M, Hwang T. "Three-party encrypted key exchange without server public-keys". *IEEE Communication Letter*, 2001,5(12), pp.497-499.

[15] Chen, Y.J., Lee, W.B., Chen, H.B., "A round-and-computation-efficient three party authentication key exchange protocol", *Journal of Systems and Software* 81, 2008, pp.1581-1590.

[16] Jen-Ho Yang, Chin-Chen Chang, "An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments", *Journal of Systems and Software*, 2008, doi:10.1016/j.jss.2009.03.075.

[17] Schnorr, C.P., "Efficient identification and signature for smart cards". in: Proceedings of CRYPTO'89, LNCS, Springer-Verlag, 1989, pp.239-252.

[18] M. Hölbl, T. Welzer, B. Brumen, "Two proposed identity-based three-party authenticated key agreement protocols from pairings", *Computers & Security* 29,2010, pp.244-252,.

[19] S. Wu and Q. Pu, "Weakness and improvement of three-party authenticated key exchange protocol using elliptic curve cryptography". Available: http:// eprint.iacr.org/2009/534.pdf.

[20] Chen Z. "Security analysis on Nalla-Reddy's ID-based tripartiteauthenticated key agreement protocols". Cryptology Eprint Archive 2003, vol.103.

[21] M. Hölbl, T. Welzer, B. Brumen, "Comparative study of tripartite identity-based authenticated key agreement protocols", *Informatica* 33，2009, pp.347–355

Zuowen Tan, born in Yiyang, Hunan, China, 1967. He received the M.S. degree in Fundamental Mathematics from Xiangtan University in 2002, and the Ph.D. degree in Applied Mathematics from Institute of Systems Science, Academy of Mathematics and System Science, CAS in 2005.

He is currently an associate professor at Department of Computer Science & Technology, School of Information Technology, Jiangxi University of Finance & Economics. He has published over 40 papers on information security in international conferences and journals. His current research interests include e-commerce security, information security and cryptography.

Dr. Tan was committee members of some international conferences such as ICCIT 2009, INC2010 and reviewers on Journals such as *International Journal of Computer Mathematics*, *Computer & Security* and *Journal of System Software.*