Acta Scientiarum

http://www.uem.br/acta
ISSN printed: 1806-2563
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v37i3.25240

Security analysis and enhancements of a three-party authenticated key agreement protocol

Zhiheng Wang[1*], Zhanqiang Huo[1] and Wenbo Shi[2]

*[1]School of Computer Science and Technique, Henan Polytechnic University, Jiaozuo, China. [2]School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao, China. *Author for correspondence. E-mail: wangzhiheng_hpu@163.com*

ABSTRACT. Three-party authenticated key agreement (3PAKA) protocol is an important cryptographic mechanism for secure communication, which allows two clients to generate a shared session key with the help of the server. Recently, Tan proposed a communication and computation-efficient 3PAKA protocol. Compared with related protocols, Tan's protocol requires fewer rounds, lower communication cost and smaller computation cost. Tan claimed that his protocol was secure against various attacks. Unfortunately, we found that his protocol cannot withstand the key compromise impersonation attack. To improve security, we proposed a new 3PAKA protocol. Security analysis and performance analysis show our 3PAKA protocol could overcome weakness in Tan's protocol at the cost of increasing the computational cost slightly.

Keywords: authenticated key agreement, three-party, provable security.

Analise de segurança e realce de um protocolo de consenso de chave autenticada por um conjunto de três pessoas

RESUMO. O protocolo de consenso de chave autenticada por um conjunto de três pessoas (3PAKA) é um mecanismo criptográfico para comunicações seguras que permitem que dois clientes gerem uma chave compartilhada com a ajuda do servidor. Tan sugeriu recentemente um protocolo 3PAKA de comunicação e computador. Quando comparado a protocolos semelhantes, o protocolo de Tan necessita menos rodadas, custos menores computacionais e de comunicação. Tan proclamou que o protocolo é seguro contra diferentes ataques, mas descobrimos que o protocolo não aguentaria o ataque impessoal de chave comprometida. Para melhorar a segurança, propusemos um novo protocolo 3PAKA. Análise de segurança e de desempenho poderia superar o ponto fraco do protocolo de Tan com uma ligeira elevação de custos.

Palavras-chave: consenso de chave autenticada, conjunto de três pessoas, segurança comprovada.

Introduction

The three-party authenticated key agreement (3PAKA) protocol is a variation of the two-party authenticated key agreement (2PAKA) protocol. In such protocol, each client shares a secret value with the server. Using the secret value, two clients could generate a shared session key for future communication with the server's help.

Bellare and Rogaway proposed the first 3PAKA protocol (CHANG et al., 2011). Since then, many 3PAKA protocols (CHANG; CHANG, 2004; CHEN et al., 2008; LO; YEH, 2009; DING; MA, 2010; YANG; CAO, 2012; YANG; CHANG et al., 2011; TAN, 2010; CHEN et al., 2008; TAN, 2013) were proposed to improve security and performance. Generally speaking, these 3PAKA protocols could be divided into three classes: the password-based 3PAKA protocols (CHEN et al., 2008; DING; MA, 2010; LO; YEH, 2009; YANG; CAO, 2012), the public key infrastructure (PKI)-based 3PAKA protocols (CHANG; CHANG, 2004; YANG; CHANG, 2009; TAN, 2010) and the identity (ID)-based 3PAKA protocols (CHEN et al., 2008; TAN, 2013). In the password-based 3PAKA protocol, each client shares an easy-to-remember password. Using shared passwords, two clients generate a session key with the help of the server. In such protocols, the server has to maintain a password table. The system will be broken totally once the password table is lost. In the PKI-based 3PAKA protocol, a certificate generated by the certificate authority is needed to bind the client's identity and his public key. The management of certificates becomes more and more difficult with the increase of the clients' number. The ID-based 3PAKA protocols could overcome the above weaknesses since no password table or certificate is needed in such protocols. Chen et al. (2008) proposed the first ID-based 3PAKA protocol. However, Yang and Chang (2009) pointed out that Chen et al.'s scheme is not secure against the stolen-verifier attack. Very recently, Tan (2013) proposed a new ID-based 3PAKA protocol. Compared with previous protocols, Tan's protocol is more practical since it requires fewer rounds, lower communication cost and smaller computation cost. Tan claimed his protocol could withstand various attacks. However, in this paper, we will point out that his protocol is vulnerable to the key compromise impersonation attack. We also propose an improved scheme to enhance security.

The rest of this paper is organized as follows. Tan's 3PAKE protocol is introduced and analyzed in Section 2 and Section 3 separately. Then, our 3PAKE protocol is proposed in Section 4. The security and performance are discussed in Section 5 and Section 6 separately. At last, some conclusions are given in Section 7.

Review of Tan's 3PAKA protocol

In this section, we review Tan's 3PAKA protocol. His protocol consists of two phases, i.e. the initialization phase and the authenticated key exchange phase. The detail is described as follows.

The initialization phase

In this phase, the server S generates the system parameters first. Then, both of the clients A and B get their private key through registering in the server.

S chooses two prime numbers p, n and a elliptic curve E defined by the equation $y^2 = x^3 + ax + b$ on the finite filed F_p, where $4a^3 + 27b^2 \neq 0 \bmod p$. S chooses a group G and a point P with order n over E. S chooses a random number $x \in Z_n^*$ as his master key. S also chooses a symmetric encryption/decryption algorithm $E_k(\cdot)/D_k(\cdot)$ and a secure hash function $h(\cdot)$. At last, S keeps x secretly and publishes the system parameters $params = \{p, E, G, P, n, E_k(\cdot)/D_k(\cdot), h(\cdot)\}$.

The client A/B sends a registration request and his identity ID_A/ID_B to S. S computes the private key $d_A = h(ID_A \| x)/d_B = h(ID_B \| x)$ and transmits it to A/B through a secure channel.

The authenticated key exchange phase

As shown in Figure 1, the client A and B generate a shared session key with the help of the server S.

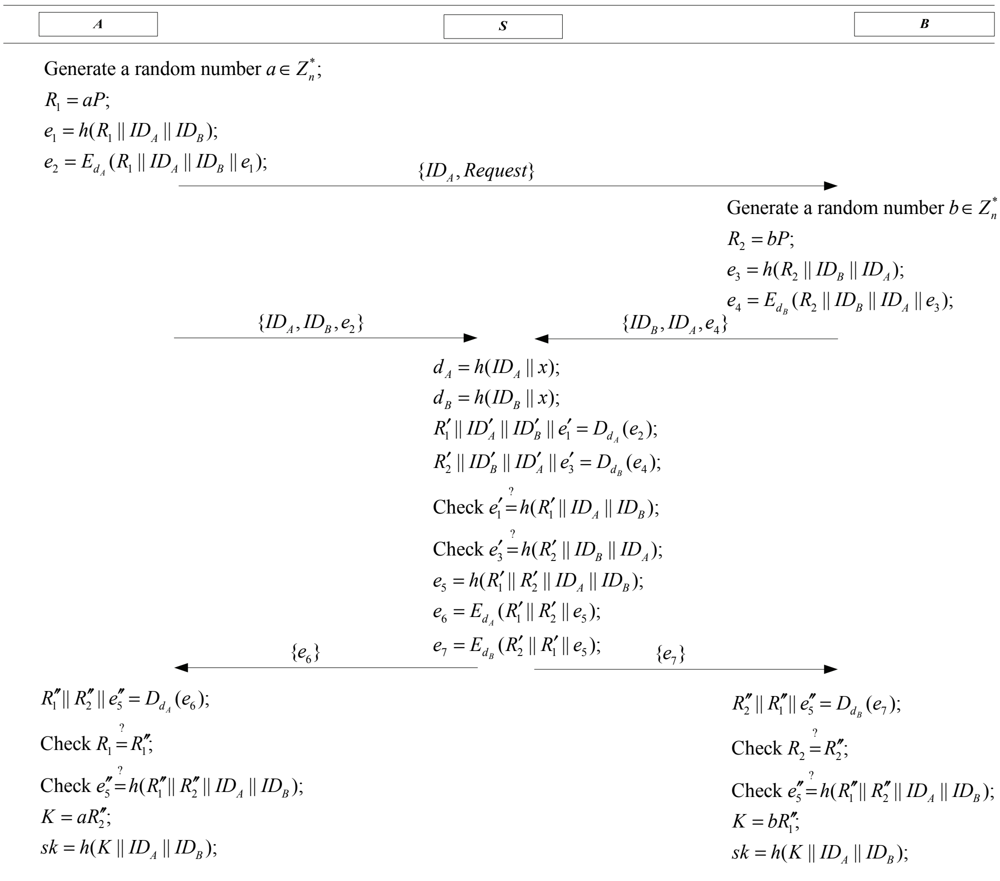
A
S
B
Generate a random number $a \in Z_n^*$;
$R_1 = aP$;
$e_1 = h(R_1 \| ID_A \| ID_B)$;
$e_2 = E_{d_A}(R_1 \| ID_A \| ID_B \| e_1)$;
$\{ID_A, Request\}$
Generate a random number $b \in Z_n^*$;
$R_2 = bP$;
$e_3 = h(R_2 \| ID_B \| ID_A)$;
$e_4 = E_{d_B}(R_2 \| ID_B \| ID_A \| e_3)$;
$\{ID_A, ID_B, e_2\}$
$\{ID_B, ID_A, e_4\}$
$d_A = h(ID_A \| x)$;
$d_B = h(ID_B \| x)$;
$R_1' \| ID_A' \| ID_B' \| e_1' = D_{d_A}(e_2)$;
$R_2' \| ID_B' \| ID_A' \| e_3' = D_{d_B}(e_4)$;
Check $e_1' \overset{?}{=} h(R_1' \| ID_A \| ID_B)$;
Check $e_3' \overset{?}{=} h(R_2' \| ID_B \| ID_A)$;
$e_5 = h(R_1' \| R_2' \| ID_A \| ID_B)$;
$e_6 = E_{d_A}(R_1' \| R_2' \| e_5)$;
$e_7 = E_{d_B}(R_2' \| R_1' \| e_5)$;
$\{e_6\}$
$\{e_7\}$
$R_1'' \| R_2'' \| e_5'' = D_{d_A}(e_6)$;
Check $R_1 \overset{?}{=} R_1''$;
Check $e_5'' \overset{?}{=} h(R_1'' \| R_2'' \| ID_A \| ID_B)$;
$K = aR_2''$;
$sk = h(K \| ID_A \| ID_B)$;
$R_2'' \| R_1'' \| e_5'' = D_{d_B}(e_7)$;
Check $R_2 \overset{?}{=} R_2''$;
Check $e_5'' \overset{?}{=} h(R_1'' \| R_2'' \| ID_A \| ID_B)$;
$K = bR_1''$;
$sk = h(K \| ID_A \| ID_B)$;

Figure 1. Authenticated key exchange phase of Tan's protocol.

1) A generates a random number $a \in Z_n^*$, computes $R_1 = aP$, $e_1 = h(R_1 \| ID_A \| ID_B)$ and $e_2 = E_{d_A}(R_1 \| ID_A \| ID_B \| e_1)$. Then, A sends $\{ID_A, Request\}$ and $\{ID_A, ID_B, e_2\}$ to B and S separately, where $Request$ is a request that A wants to generate a session key with B.

2) Upon receiving $\{ID_A, Request\}$, B generates a random number $b \in Z_n^*$, computes $R_2 = bP$, $e_3 = h(R_2 \| ID_B \| ID_A)$ and $e_4 = E_{d_B}(R_2 \| ID_B \| ID_A \| e_3)$. Then, B sends $\{ID_B, ID_A, e_4\}$ to S.

3) Upon receiving $\{ID_A, ID_B, e_2\}$ and $\{ID_B, ID_A, e_4\}$, S computes $d_A = h(ID_A \| x)$, $d_B = h(ID_B \| x)$, $R_1' \| ID_A' \| ID_B' \| e_1' = D_{d_A}(e_2)$ and $R_2' \| ID_B' \| ID_A' \| e_3' = D_{d_B}(e_4)$. S checks whether both of the two equations $e_1' = h(R_1' \| ID_A \| ID_B)$ and $e_3' = h(R_2' \| ID_B \| ID_A)$ hold. If so, S computes $e_5 = h(R_1' \| R_2' \| ID_A \| ID_B)$, $e_6 = E_{d_A}(R_1' \| R_2' \| e_5)$ and $e_7 = E_{d_B}(R_2' \| R_1' \| e_5)$. At last, S sends $\{e_6\}$ and $\{e_7\}$ to A and B separately.

4) Upon receiving $\{e_6\}$, A computes $R_1'' \| R_2'' \| e_5'' = D_{d_A}(e_6)$ and checks whether both of the two equations $R_1 = R_1''$ and $e_5'' = h(R_1'' \| R_2'' \| ID_A \| ID_B)$ hold. If so, A computes $K = aR_2''$ and the session key $sk = h(K \| ID_A \| ID_B)$.

5) Upon receiving $\{e_7\}$, B computes $R_2'' \| R_1'' \| e_5'' = D_{d_B}(e_7)$ and checks whether both of the two equations $R_2 = R_2''$ and $e_5'' = h(R_1'' \| R_2'' \| ID_A \| ID_B)$ hold. If so, B computes $K = bR_1''$ and the session key $sk = h(K \| ID_A \| ID_B)$.

Weakness of Tan's 3PAKA protocol

It is well known that a 3PAKA protocol could provide five basic security attributes, i.e. known-key security, perfect forward secrecy, key-compromise impersonation resilience, unknown key-share resilience and no key control (CHEN; HAN, 2013; HE et al., 2014; HE et al. 2015, HE; ZEADALLY, 2015, MENEZES et al., 1997; TAN, 2013;). In the 3PAKA, key-compromise impersonation resilience means that the adversary A cannot impersonate the client B and the server S to the client A when he gets A's private key. In this section, we will show that Tan's protocol cannot provide key-compromise impersonation resilience by proposing a concrete key compromise impersonation attack. Once A gets A's private key $d_A = h(ID_A \| x)$, he could carry out the attack as follows.

1) A generates a random number $a \in Z_n^*$, computes $R_1 = aP$, $e_1 = h(R_1 \| ID_A \| ID_B)$ and $e_2 = E_{d_A}(R_1 \| ID_A \| ID_B \| e_1)$. Then, A sends $\{ID_A, Request\}$ and $\{ID_A, ID_B, e_2\}$ to B and S separately, where $Request$ is a request that A wants to generate a session key with B.

2) A intercepts the message $\{ID_A, Request\}$ and $\{ID_A, ID_B, e_2\}$.

3) A computes $R_1' \| ID_A' \| ID_B' \| e_1' = D_{d_A}(e_2)$ and checks whether the equations $e_1' = h(R_1' \| ID_A' \| ID_B')$ holds. If so, A generates a random number $b \in Z_n^*$, computes $R_2 = bP$, $e_5 = h(R_1' \| R_2 \| ID_A \| ID_B)$, $e_6 = E_{d_A}(R_1' \| R_2 \| e_5)$, $K = bR_1'$ and the session key $sk = h(K \| ID_A \| ID_B)$. A sends $\{e_6\}$ to A.

4) Upon receiving $\{e_6\}$, A computes $R_1'' \| R_2'' \| e_5'' = D_{d_A}(e_6)$ and checks whether both of the two equations $R_1 = R_1''$ and $e_5'' = h(R_1'' \| R_2'' \| ID_A \| ID_B)$ hold. It is easy to say both of the above two equations hold. Then, A computes $K = aR_2''$ and the session key $sk = h(K \| ID_A \| ID_B)$.

Since $K = aR_2'' = bR_1' = abP$, then A could impersonate B to generate a shared session key with A. Therefore, Tan's protocol cannot withstand the key compromise impersonation attack.

Our 3PAKA protocol

In Tan's scheme, the adversary does not need the server's private key to generate the response message for the user A if he gets A's private key. Therefore, the adversary could impersonate B to A. To overcome such weakness, we should let the server's private play an important role in generating response message. Based on the observation, we propose an improved 3PAKA protocol to overcome weakness in Tan's protocol. Like his protocol, our protocol also consists of two phases, i.e. the initialization phase and the authenticated key exchange phase. The detail is described as follows.

The initialization phase

In this phase, the server S generates the system parameters first. Then, both of the clients A and B get their private key through registering in the server.

S chooses two prime numbers p, n and a elliptic curve E defined by the equation $y^2 = x^3 + ax + b$ on the finite filed F_p, where $4a^3 + 27b^2 \neq 0 \bmod p$. S chooses a group G and a point P with order n over E. S chooses a random number $x \in Z_n^*$ as his master key. S also chooses a secure hash function $h(\cdot)$. At last, S keeps x secretly and publishes the system parameters $params = \{p, E, G, P, n, h(\cdot)\}$.

The client A / B sends a registration request and his identity ID_A / ID_B to S. S computes the private key $D_A = h(ID_A \| x)P / D_B = h(ID_B \| x)P$ and transmits it to A / B through a secure channel.

The authenticated key exchange phase

As shown in Figure 2, the client A and B generate a shared session key with the help of the server S.

1) A generates a random number $a \in Z_n^*$, computes $R_1 = aP$, $R_2 = aD_A$ and $e_1 = h(R_1 \| R_2 \| ID_A \| ID_B)$. Then, A sends $\{ID_A, Request\}$ and $\{ID_A, ID_B, R_1, e_1\}$ to B and S separately, where $Request$ is a request that A wants to generate a session key with B.

2) Upon receiving $\{ID_A, Request\}$, B generates a random number $b \in Z_n^*$, computes $R_3 = bP$, $R_4 = bD_B$ and $e_2 = h(R_3 \| R_4 \| ID_B \| ID_A)$. Then, B sends $\{ID_B, ID_A, R_3, e_2\}$ to S.

3) Upon receiving $\{ID_A, ID_B, R_1, e_1\}$ and $\{ID_B, ID_A, R_3, e_2\}$, S computes $d_A = h(ID_A \| x)$, $d_B = h(ID_B \| x)$, $R_2' = d_A R_1$ and $R_4' = d_B R_3$. S checks whether both of the two equations $e_1 = h(R_1 \| R_2' \| ID_A \| ID_B)$ and $e_2 = h(R_3 \| R_4' \| ID_B \| ID_A)$ hold. If so, S computes $e_3 = h(R_1 \| R_2' \| ID_A \| ID_B \| R_3)$ and $e_4 = h(R_3 \| R_4' \| ID_B \| ID_A \| R_1)$. At last, S sends $\{e_3, R_3\}$ and $\{e_4, R_1\}$ to A and B separately.

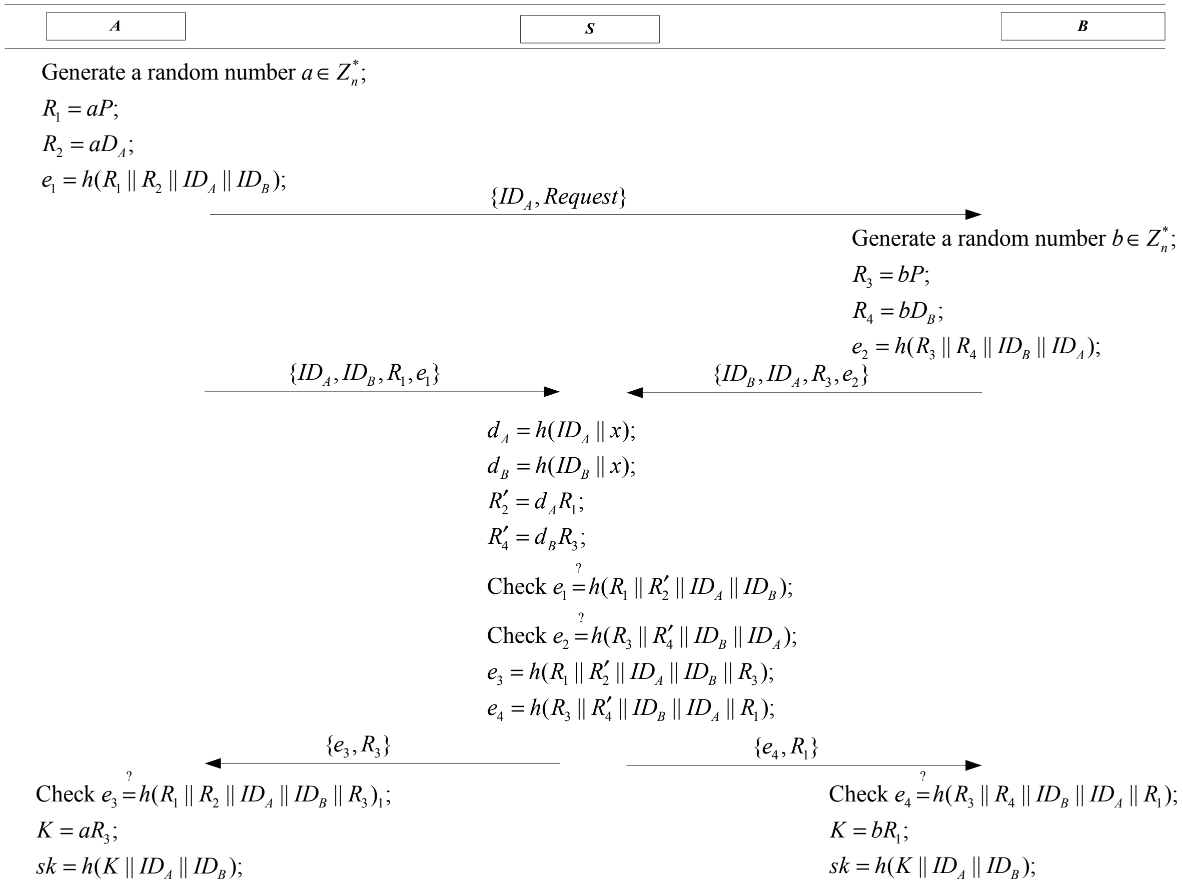
A
S
B
Generate a random number $a \in Z_n^*$;
$R_1 = aP$;
$R_2 = aD_A$;
$e_1 = h(R_1 \| R_2 \| ID_A \| ID_B)$;
$\{ID_A, Request\}$
Generate a random number $b \in Z_n^*$;
$R_3 = bP$;
$R_4 = bD_B$;
$e_2 = h(R_3 \| R_4 \| ID_B \| ID_A)$;
$\{ID_A, ID_B, R_1, e_1\}$
$\{ID_B, ID_A, R_3, e_2\}$
$d_A = h(ID_A \| x)$;
$d_B = h(ID_B \| x)$;
$R_2' = d_A R_1$;
$R_4' = d_B R_3$;
Check $e_1 \overset{?}{=} h(R_1 \| R_2' \| ID_A \| ID_B)$;
Check $e_2 \overset{?}{=} h(R_3 \| R_4' \| ID_B \| ID_A)$;
$e_3 = h(R_1 \| R_2' \| ID_A \| ID_B \| R_3)$;
$e_4 = h(R_3 \| R_4' \| ID_B \| ID_A \| R_1)$;
$\{e_3, R_3\}$
$\{e_4, R_1\}$
Check $e_3 \overset{?}{=} h(R_1 \| R_2 \| ID_A \| ID_B \| R_3)_1$;
$K = aR_3$;
$sk = h(K \| ID_A \| ID_B)$;
Check $e_4 \overset{?}{=} h(R_3 \| R_4 \| ID_B \| ID_A \| R_1)$;
$K = bR_1$;
$sk = h(K \| ID_A \| ID_B)$;

Figure 2. Authenticated key exchange phase of our 3PAKA protocol.

4) Upon receiving $\{e_3\}$, A and checks whether both of the equation $e_3 = h(R_1 \| R_2 \| ID_A \| ID_B \| R_3)$ holds. If so, A computes $K = aR_3$ and the session key $sk = h(K \| ID_A \| ID_B)$.

5) Upon receiving $\{e_4\}$, B checks whether both of the equation $e_4 = h(R_3 \| R_4 \| ID_B \| ID_A \| R_1)$ holds. If so, B computes $K = bR_1$ and the session key $sk = h(K \| ID_A \| ID_B)$.

Security analysis of our 3PAKA protocol

Security model for 3PAKA protocol

In this subsection, we proposed a security model for 3PAKA protocol based on Chang et al.'s security mode for password-based 3PAPA protocol (CHANG et al., 2011).

Let $\prod_U^i$ represents the i th instance of a participant U. The security of a 3PAKA protocol is defined by a game between a challenger C and an adversary A. There are two phases in the game. During the first phase, A could issue the following queries at his will.

Hash(m): C maintains an initially empty table $H-$table which contains tuples of the form (m,r).Upon receiving the query, C looks up $H-$table. If there is entry (m,r) in $H-$table, C returns r to A; Otherwise, C generates a random number r, stores (m,r) into $H-$table and returns r to A.

Send$(\prod_U^i, m)$: Through the query, A could send the message m to $\prod_U^i$ and get a response according to the specification of the protocol.

Execute$(\prod_A^i, \prod_B^j, \prod_S^j)$: The query models A could obtain an honest execution of the 3PAKA protocol among $\prod_A^i, \prod_B^j$ and $\prod_S^j$.

Reveal$(\prod_U^i)$: Through the query, A could get the session key of $\prod_U^i$.

Corrupt(ID_U): Through the query, A could get the private key of participant U with identity ID_U.

The adversary A could start the second phase by choosing a fresh instance $\prod_U^i$ and issuing a *Test*$(\prod_U^i)$ query. The fresh session and *Test* query are defined as follows.

An oracle $\prod_U^i$ is fresh if the following three conditions are satisfied. (1) $\prod_U^i$ has been accepted, (2) no oracle has been asked *Corrupt* query before $\prod_U^i$ is accepted., (3) neither $\prod_U^i$ nor his partner has been asked a *Reveal* query.

Test$(\prod_U^i)$: Upon receiving the query, C flips fair coin $b \in \{0,1\}$, and returns the session key held by $\prod_U^i$ if $b = 0$, or a random sample from the distribution of the session key if $b = 1$. The query is only available if $\prod_U^i$ is fresh and the adversary is only allowed to make *Test* query one time.

After making the *Test*$(\prod_U^i)$ query, A could continue querying unless the test oracle $\prod_U^i$ remains fresh. At the end of the game, A outputs a guess bit b'. We say that A wins if and only if $b' = b$. A's advantage $Adv_{\mathsf{A}}(k)$ to win the above game is defined as $Adv_{\mathsf{A}}(k) = \left|\Pr[b' = b] - \frac{1}{2}\right|$, where k is a security parameter.

Definition 1. A 3PAKA protocol is said to be secure if:

(1) In the presence of a benign adversary on $\prod_A^i, \prod_B^j$ and $\prod_S^j$, both of $\prod_A^i$ and $\prod_B^j$ always agree on the same session key, and this key is distributed uniformly at random.

(2) For all probabilistic polynomial time adversary A, $Adv_{\mathsf{A}}(k)$ is negligible.

Security analysis

To prove the security of our 3PAKA protocol in the random oracle model (HE et al., 2012, 2013), we treat h as a random oracle. For the security, the following lemmas and theorems are provided.

Lemma 1. If two oracles $\prod_A^i$ and $\prod_B^j$ are matching, both of them will be accepted and will get the same session key which is distributed uniformly at random in the session key sample space.

Proof. From the description of our 3PAKA protocol, we know if two oracles $\prod_A^i$ and $\prod_B^j$ are matching, then both of them are accepted and have the same session key. The session keys are distributed uniformly since a and b are selected uniformly during the execution of our 3PAKA protocol.

Lemma 2. Assuming that the computational Diffie-Hellman (CDH) problem is hard, the advantage of any adversary against our 3PAKA protocol is negligible.

Proof. Suppose there is an adversary A could win the game described in Section 5.2 with a non-negligible advantage ε. We will show that there is an algorithm C could solve the CDH problem using A's ability.

Given an instance $\psi = (\alpha P, \beta P)$ of CDH problem, C chooses the system parameters $params = \{p, E, G, P, n, h(\cdot)\}$, a random number $x \in Z_n^*$ and a random number $j \in [1, q_{se}]$, where q_{se} denotes the number *Send* query. C keeps x as the master key and use it to generate all participants' private keys. Then, C sends *params* to A, and answers A 's queries.

C answers all A 's queries according the description of our 3PAKA protocol, except the j th *Send* query. In the j th *Send* query, C embeds $(\alpha P, \beta P)$ as (R_1, R_2) and returns the corresponding response to A .

When A outputs a guess b' about b , C looks up the $H-$table to say some queries of format $Hash(*, ID_A, ID_B)$ has been asked. If there is no such query has been asked, C sops the simulation; otherwise, C chooses a random on of such format and return * as the solution of the CDH problem.

In our simulation, the hash function h is treated as a random oracle. Then we could conclude that if A could know the session key sk corresponding to the j th *Send* query, he must have asked a $Hash(*, ID_A, ID_B)$ query which is stored in $H-$table.

C chooses the correct format $(*, ID_A, ID_B)$ from $H-$table with a probability $\frac{1}{q_h}$, where q_h is the number of *Hash* query. The probability that C guesses the correct moment when A wins the game is $\frac{1}{q_{se}}$ since it equal the probability that C guesses the correct j . Therefore, C could solve the CDH problem with a non- negligible advantage $\eta = \frac{1}{q_{se}q_h}\varepsilon$ since ε is non- negligible. This contradicts with the hardness of the CDH problem.

From the above three lemmas, we can get the following theorem.

Theorem 1. Assuming that the computational Diffie-Hellman (CDH) problem is hard, Our protocol is a secure 3PAKA protocol in the random oracle model.

Other discussion

In this subsection, we will show the proposed 3PAKA protocol could provide known-key security, perfect forward secrecy, key-compromise impersonation resilience, unknown key-share resilience and no key control (TAN, 2013; CHEN; HAN, 2013; FU et al., 2015; GUO et al., 2014; HE et al., 2014, 2015; HE; ZEADALLY, 2015; MENEZES et al., 1997; SHEN et al., 2015).

Known-key security

The known-key security means that the execution of a protocol should result a unique secret session key and the compromise of this key has no impact on other session keys.

From the description of our 3PAKA protocol, we could get that the client A and B compute $K = aR_3 = abP$ and $K = bR_1 = abP$ separately. Then, the execution of our 3PAKA protocol should result a unique secret session key $sk = h(K \| ID_A \| ID_B)$. Because the random numbers a and b are generated by A and B separately for every session, then the compromise of this key has no impact on other session keys. Therefore, our 3PAKA protocol could provide the known-key security.

Perfect forward secrecy

The perfect forward secrecy means that the previous session keys cannot be compromised even all three parties' long-term private keys are compromised.

In our 3PAKA protocol, the session key is $sk = h(K \| ID_A \| ID_B)$, where $K = abP$ and the random numbers a and b are generated by A and B separately. Even the adversary gets all three parties' long-term private keys, he still cannot compute $K = abP$ from $R_1 = aP$ and $R_3 = bP$ since he will face with the CDH problem. Therefore, our 3PAKA protocol could provide the perfect forward secrecy.

Key-compromise impersonation resilience

The key-compromise impersonation resilience means that the adversary A cannot impersonate the client B and the server S to the client A when he gets A 's long-term private key.

Assume that the adversary could get A 's long-term private key $D_A = h(ID_A \| x)P$. He could intercept the message $\{ID_A, Request\}$ and $\{ID_A, ID_B, R_1, e_1\}$ sent by A , where $R_1 = aP$, $R_2 = aD_A$ and $e_1 = h(R_1 \| R_2 \| ID_A \| ID_B)$. To carry out the key-compromise impersonation attack, he has to $R_2' = h(ID_A \| x)aP$ from $D_A = h(ID_A \| x)P$ and $R_1 = aP$ to generate a legal message $\{e_3, R_3\}$, where $e_3 = h(R_1 \| R_2' \| ID_A \| ID_B \| R_3)$. Then, he will face with the CDH problem. Therefore, our 3PAKA protocol could provide the key-compromise impersonation resilience.

Unknown key-share resilience

The unknown key-share resilience means that the client A believes he generate a session key with the client B , it is impossible that A is tricked to generate a session key with the client C .

In our 3PAKA protocol, the client A and the server S could authenticate each other through checking e_3 and e_1 separately. The client B and the server S could authenticate each other through checking e_3 and e_4 separately. Then, A and B could authenticate each other with the help of S. Therefore, our 3PAKA protocol could provide the unknown key-share resilienc.

No key control

The no key control means that none of three parties cannot force the session key to be a pre-choose value.

In our 3PAKA protocol, the session key is $sk = h(K \| ID_A \| ID_B)$, where $K = abP$ and the random numbers a and b are generated by A and B separately. Then, none of A, B and S cannot determine the session key of a execution of our 3PAKA protocol. Therefore, our 3PAKA protocol could provide the no key control.

Performance analysis

In this section, we will compare the performance of our 3PAKA protocol with that of Chen et al.'s protocol (CHEN et al., 2008) and Tan's protocol (TAN, 2013). For convenience, some notations are defined as follows.

T_m : the running time of a elliptic curve point multiplication operation;

T_h : the running time of a hash function operation;

T_s : the running time of a symmetric encryption/decryption operation;

To achieve 1024-bit RSA level security, we employed a Koblitz elliptic curve $y^2 = x^3 + ax^2 + b$ defined on $F_{2^{163}}$ with a = 1. To give a fair comparison, we transferred Chen et al.'s protocol into the elliptic curve analogue version. We assume that the size of p, the output size of hash function, the output size of symmetric encryption/decryption algorithm, the size of timestamp, the size of "*Request*" and the size of client's identity is 160 bits, 160 bits, 128 bits, 32 bits, 32 bist and 32 bits separately. The comparisons in term of communicational cost and computational cost are listed in Table 1. Our 3PAKA protocol has better performance in term of the communicational cost than Chen et al.'s protocol and Tan's protocol. Tan's protocol has better performance in term of computational cost than our 3PAKA protocol and Chen et al.'s protocol. However, Chen et al.'s protocol and Tan's protocol are vulnerable to the stolen-verifier attack and the key compromise impersonation attack separately. Our 3PAKA protocol could overcome security weakness in previous protocols at the cost of increasing computational cost slightly. Therefore, our 3PAKA protocol is more suitable for practical applications.

Table 1. Performance comparison.

	Chen et al.'s protocol	Tan's protocol	Our 3PAKA protocol
Communicational cost	2720bits	2816bits	2112bits
Computational cost of A	$4T_m + 4T_h$	$2T_m + 3T_h + 2T_s$	$3T_m + 3T_h$
Computational cost of B	$4T_m + 4T_h$	$2T_m + 3T_h + 2T_s$	$3T_m + 3T_h$
Computational cost of S	$2T_m + 6T_h$	$3T_h + 2T_s$	$2T_m + 6T_h$

Conclusion

Due to overcoming weaknesses in the 2PAKA protocol, the 3PAKA protocol attracted wide attentions from all over the world. Many 3PAKA protocols have been proposed for practical applications in last several years. In this paper, we analyze the security of a novel 3PAKA protocol based on the elliptic curve cryptography and point out that it cannot withstand the key compromise impersonation attack. To enhance security, this paper proposes a new 3PAKA protocol based on the elliptic curve cryptography. A security analysis show the proposed 3PAKA protocol could overcome weakness in previous schemes and is provably secure in the random oracle model. A performance analysis shows that the proposed 3PAKA protocol has better communication cost and increases the computation cost slightly. Therefore, the proposed 3PAKA protocol is more practical than previous schemes.

Acknowledgements

The authors thank the editors and the anonymous reviewers for their valuable comments. This research was supported by the National Natural Science Foundation of China (Nos.61202447, 61472074), the Natural Science Foundation of Hebei Province of China (No. F2013501066), the Northeastern University at Qinhuangdao Science and Technology Support Program (No. xnk201307), the Fundamental Research Funds for the Central Universities（No.N130423008）.

Reference

CHANG, C.; CHANG, Y. A novel three-party encrypted key exchange protocol. **Computer Standards and Interfaces**, v. 26, n. 5, p. 472-476, 2004.

CHANG, T.; HWANG, M.; YANG, W. A communication-efficient three-party password authenticated key exchange protocol. **Information Sciences**, v. 181, n. 1, p. 217-226, 2011.

CHEN, H.; CHEN, T.; LEE, W.; CHANG, C. Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks. **Computer Standards and Interfaces**, v. 30, n. 1-2, p. 95-99, 2008a.

CHEN, Y.; HAN, W. Efficient identity-based authenticated multiple key exchange protocol. **Acta Scientiarum. Technology**, v. 35, n. 4, p. 629-636, 2013.

CHEN, Y.; LEE, W.; CHEN, H. A round-and-computation-efficient three party authenticated key agreement protocol. **Journal of Systems and Software**, v. 81, n. 9, p. 1581-1590, 2008b.

DING, X.; MA, C. The three-party password authenticated key exchange protocol with stronger security. **Journal of Computer**, v. 33, n. 1, p. 111-118, 2010.

FU, Z.; SUN, X.; LIU, Q.; ZHOU, L.; SHU, J. Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. **IEICE Transactions on Communications**, v. 98, n. 1, p. 190-200, 2015.

GUO, P.; WANG, J.; LI, B.; LEE, S. A variable threshold-value authentication architecture for wireless mesh networks. **Journal of Internet Technology**, v. 15, n. 6, p. 929-936, 2014.

HE, D.; CHEN, J.; HU, J. A pairing-free certificateless authenticated key agreement protocol. **International Journal of Communication Systems**, v. 25, n. 2, p. 221-230, 2012.

HE, D.; HUANG, B.; CHEN, J. New certificateless short signature scheme. **IET Information Security**, v. 7, n. 2, p. 113-117, 2013.

HE, D.; ZHANG, Y.; CHEN, J. Cryptanalysis and improvement of an anonymous authentication protocol for wireless access networks. **Wireless Personal Communications**, v. 74, n. 2, p. 229-243, 2014.

HE, D.; KUMAR, N.; CHEN, J.; LEE, C.; CHILAMKURTI, N.; YEO, S. Robust anonymous authentication protocol for healthcare applications using wireless medical sensor networks. **Multimedia Systems**, v. 21, n. 1, p. 49-60, 2015a.

HE, D.; ZEADALLY, S. Authentication protocol for ambient assisted living system. **IEEE Communications Magazine**, v. 35, n. 1, p. 71-77, 2015b.

LO, N.; YEH, K. Cryptanalysis of two three-party encrypted key exchange protocols. **Computer Standards and Interfaces**, v. 31, n. 6, p. 1167-1174, 2009.

MENEZES, A.; OORSCHOT, P.; VANSTONE, S. **Handbook of Applied Cryptograph**. NewYork: CRC Press, 1997.

SHEN, J.; TAN, H.; WANG, J.; WANG, J.; LEE, S. A novel routing protocol providing good transmission reliability in underwater sensor networks. **Journal of Internet Technology**, v. 16, n. 1, p. 171-178, 2015.

TAN, Z. An enhanced three-party authenticated key agreement protocol using elliptic curve cryptography for mobile commerce environments. **Journal of Communications**, v. 5, n. 6, p. 436-443, 2010.

TAN, Z. A communication and computation-efficient three-party authenticated key agreement protocol. **Security and Communication Networks**, v. 6, n. 7, p. 854-863, 2013.

YANG, J.; CAO, T. Provably secure three-party password authenticated key exchange protocol in the standard model. **Journal of Systems and Software**, v. 85, n. 2, p. 340-350, 2012.

YANG, J.; CHANG, C. An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments. **Journal of Systems and Software**, v. 82, n. 9, p. 1497-1502, 2009.

Received on October 10, 2014.
Accepted on March 2, 2015.