Multimed Tools Appl (2015) 74:6365–6377
DOI 10.1007/s11042-015-2470-3

A strongly secure pairing-free certificateless authenticated key agreement protocol suitable for smart media and mobile environments

Hang Tu · Neeraj Kumar · Jongsung Kim · Jungtaek Seo

Received: 16 December 2014 / Accepted: 14 January 2015 / Published online: 27 February 2015

Abstract The authenticated key agreement (AKA) protocol is an important cryptographic mechanism, which allows two users to establish a session key for future communication. Recently, the certificateless public key cryptography received wide attention since it could solve the certificate management problem in the traditional public key cryptography and solve the key escrow problem in the identity-based public key cryptography. In this paper, we present a strongly secure certificateless authenticated key agreement (CLAKA) protocol without pairing suitable for smart media and mobile environments, which is provably secure in the extended Canetti–Krawczyk (eCK) model and is secure as long as each party has at least one uncompromised secret. Compared with previous CLAKA protocols, our protocol has advantages over them in security or efficiency.

Keywords Certificateless cryptography · Authenticated key agreement · Provable security · Bilinear pairings · Elliptic curve

1 Introduction

To solve the certificate management problem in the traditional public key cryptography, Shamir [17] proposed the concept of identity-based public key cryptography (ID-based PKC) in 1984.

H. Tu
School of Computer, Wuhan University, Wuhan, China

N. Kumar
Department of Computer Science and Engineering, Thapar University, Patiala, India
e-mail: neeraj.kumar@thapar.edu

J. Kim (✉)
Department of Mathematics and Department of Financial Information Security (BK21 Plus Future Financial Information Security Specialist Education Group), Kookmin University, Seoul, Republic of Korea
e-mail: jongsung.k@gmail.com

J. Seo
National Security Research Institute (NSRI), Daejeon, Republic of Korea
e-mail: seojt@ensec.re.kr

There is no certificate is required in ID-based PKC since the user's public key is his identity such as name, e-mail address, telephone number et al. However, the user's private key is generated by key generation centre (KGC) in the ID-based PKC. Then the ID-based PKC has to face with the key escrow problem, i.e. the KGC knows user's private key. To solve the problem, Al-Riyami et al. [1] proposed the concept of the certificateless public key cryptosystem (CLPKC). In CLPKC, a user's private key is comprised of partial secret and a secret value, which are generated by the KGC and the user separately. Then, the CLPKC could solve the key escrow problem in the ID-based PKC. Since Al-Riyami et al.'s work, many cryptographic mechanisms in certificateless setting have been proposed.

Authenticated key agreement (AKA) protocol is a cryptographic mechanism, through which two users can generate a shared session key over an open network. As an important protocol of the CLPKC, certificateless authenticated key agreement (CLAKA) protocol has been studied widely. After the pioneering work of Al-Riyami et al., many CLAKA protocols [14, 15, 18–21, 23] using pairings have been proposed to satisfy different applications. From the theoretical analysis [7] and experimental results [6] we know that the pairing operation is a very complicated operation. To improve efficiency, several pairing-free CLAKA protocols [8–10, 12] have been proposed. The authors also demonstrated that their protocols are provably secure in formal models. Bellare et al. [3] proposed the first formal model for AKA protocols. After that, several extended models have been proposed [2, 4, 5]. Among them, the Canetti-Krawczyk (CK) model [5] is considered as the most promising one. To capture more desirable security properties, LaMacchia et al. [13] presented a more strong security model—the extended Canetti-Krawczyk (eCK) model for AKA protocols. In 2009, Lippold et al. [14] proposed the eCK model for CLAKA protocols. They also proposed the first CLAKA protocols using paring, which is provably secure in the eCK model for CLAKA protocols. Yang et al. [22] found that these pairing-free CLAKA protocols [8–10, 12] are not secure in the eCK model. To improve security, Yang et al. proposed a new pairing-free CLAKA protocol and demonstrated it is provably secure in the eCK model. In Yang et al.'s protocol, nine elliptic curve scalar multiplication operations are needed to generate a session key. To improve performance, He et al. [11] proposed an efficient protocol. However, He et al.'s protocol is not secure in Lippold et al.'s model since the adversary could compute the session key if he get the initiator's partial private key and the ephemeral private key and the responder's secret value and the ephemeral private key. In this paper, we present a strongly secure CLAKA without pairing, which is provably secure in the eCK model and is secure as long as each party has at least one uncompromised secret. Compared with previous ID-based AKA protocols, our protocol has advantages over them in security or efficiency.

The remainder of this paper is organized as follows. Section 2 describes some preliminaries. In Section 3, we propose our CLAKA protocol. The security analysis of the proposed protocol is presented in Section 4. In Section 5, performance analysis is presented. Finally, in Section 6 we conclude the result.

2 Preliminaries

2.1 Background of elliptic curve group

Let the symbol E/F_p denote an elliptic curve E over a prime finite field F_p, defined by an equation

$$y^2 = x^3 + ax + b \quad , \quad a, b \in F_p \tag{1}$$

and with the discriminant

$$\Delta = 4a^3 + 27b^2 \neq 0. \tag{2}$$

The points on E/F_p together with an extra point O called the point at infinity form a group

$$G = \{(x,y) : x,y \in F_p, E(x,y) = 0\} \cup \{O\} \tag{3}$$

G is a cyclic additive group in the point addition "+" defined as follows: Let $P,Q \in G$, l be the line containing P and Q (tangent line to E/F_p if $P = Q$), and R, the third point of intersection of l with E/F_p. Let l' be the line connecting R and O. Then P "+" Q is the point such that l' intersects E/F_p at R and O. Scalar multiplication over E/F_p can be computed as follows:

$$tP = P + P + \ldots + P(t\ times) \tag{4}$$

Let the order of G be n. The following problems are commonly used in the security analysis of many cryptographic protocols.

Computational Diffie-Hellman (CDH) problem Given a generator P of G and (aP,bP) for unknown $a,b \in_R Z_n^*$, the task of CDH problem is to compute abP.

2.2 Security model for CLAKA protocols

Lippold et al. proposed the eCK model for CLAKA protocols based on Swanson's work. We will give an introduction to Lippold et al.'s model. The details of the eCK model for CLAKA can be found in [14].

Let $U=\{U_1,U_2,\ldots,U_n\}$ be a set of parties. The protocol may be run between any two of these parties. For each party there exists an identity based public key that can be derived from its identity. There is a KGC that generates a party's partial private key according to his identity. Additionally, a party generates their own secret values and public keys. The adversary $\mathcal{A}$ is modeled as a probabilistic polynomial-time Turing machine and has full control of the communication network over which protocol messages are exchanged. Let $\prod_{i,j}^{x}$ denote x th protocol session which is run between the party U_i (the initiator) and the partner party U_j (the responder). We say that a session $sk_{i,j}^{x}$ enters an *accepted* state when it computes a session key $sk_{i,j}^{x}$. The session $sk_{i,j}^{x}$ is assigned a partner identity $pid=(ID_i,ID_j)$. Let *comms* denote the transcript of the messages exchanged between the initiator and the responder during the session. Two sessions $sk_{i,j}^{x}$ and $\prod_{j,i}^{y}$ are called matching if they have the same *comms* and *pid*. The game runs in two phases. During the first phase of the game, the adversary $\mathcal{A}$ is allowed to issue the following queries in any order:

> *Send*($\prod_{i,j}^{x}$,m): The adversary sends the message m to party i in session $\prod_{i,j}^{u}$ on behalf of party j and gets response from i according to the protocol specification. In the case of one-round protocols, party i behaves as follows:

- $m=\lambda$: Party i generates an ephemeral value and responds with an outgoing message only.
- $m\neq\lambda$ If party i is a responder, it generates an ephemeral value for the session and responds with an outgoing message m and a decision indicating acceptance or rejection of the

session. If party i is an initiator, it responds with a decision indicating accepting or rejecting the session.

RevealMasterKey: The adversary is given the master secret key.
RevealSessionKey($\prod_{i,j}^{u}$): If the session has not accepted, it returns $\perp$, otherwise it reveals the accepted session key.
RevealPartialPrivateKey(i): The adversary is given party i's partial private key.
RevealSecretValue(i): If party i has been asked the *ReplacePublicKey* query, it returns $\perp$. Otherwise, the adversary is given party i's secret value.
ReplacePublicKey(i,pk): The adversary replaces party i's public key with the value chosen by himself.
RevealEphemeralKey($\prod_{i,j}^{x}$): The adversary is given the ephemeral secret key used in $\prod_{i,j}^{x}$.

Once the adversary $\mathcal{A}$ decides that the first phase is over, it starts the second phase by choosing a fresh session $\prod_{i,j}^{x}$ and issuing a *Test*($\prod_{i,j}^{x}$) query, where the fresh session and test query are defined as follows:

Definition 1 (**Fresh session**) [14]. A session $\prod_{i,j}^{x}$ is fresh if (1) $\prod_{i,j}^{x}$ has accepted; (2) $\prod_{i,j}^{x}$ is unopened (not being issued the *RevealSessionKey* query); (3) the session state at neither party participating in this session is fully corrupted; (4) there is no opened session $\prod_{j,i}^{y}$ which has a matching conversation to $\prod_{i,j}^{x}$.

Test($\prod_{i,j}^{x}$): The input session $\prod_{i,j}^{x}$ must be fresh. A bit $b \in \{0,1\}$ is randomly chosen. If b=0, the adversary is given the session key, otherwise it randomly samples a session key from the distribution of valid session keys and returns it to the adversary.

After the *Test*($\prod_{i,j}^{x}$) query has been issued, the adversary can continue querying except that the test session $\prod_{i,j}^{x}$ should remain fresh. At the end of the game, the adversary $\mathcal{A}$ outputs a guess bit b'. $\mathcal{A}$ wins if and only if $b'=b$. $\mathcal{A}$'s advantage to win the above game, denoted by $Adv_{\mathcal{A}}(k)$, is defined as: $Adv_{\mathcal{A}}(k) = \left|\Pr[b' = b] - \frac{1}{2}\right|$, where k is a security parameter.

There are two kinds of adversaries in the CLAKA protocol, i.e. the Type I adversary $\mathcal{A}1$ and the Type II adversary $\mathcal{A}2$. The adversary $\mathcal{A}1$ is not able to access the master key but he could replace public keys at his will. The adversary $\mathcal{A}2$ represents a malicious KGC who generates partial private keys of users. $\mathcal{A}2$ could access to the master key, but he is not able to replace public keys.

Definition 2 (**Strong Type I secure key agreement protocol**) [14]. A CLAKA protocol is Strong Type I secure if every probabilistic, polynomial-time adversary $\mathcal{A}$ has negligible advantage in winning the above game subject to the following constraints:

- $\mathcal{A}$ may corrupt at most two out of three types of secrets per party involved in the test session,
- $\mathcal{A}$ is allowed to replace public keys of any party; however, this counts as the corruption of one secret,
- $\mathcal{A}$ may not reveal the secret value of any identity for which it has replaced the certificateless public key,
- $\mathcal{A}$ is allowed to ask session key reveal queries even for session keys computed by identities where $\mathcal{A}$ replaced the identity's public key,
- $\mathcal{A}$ is allowed to replace public keys of any party after the test query has been issued.

Definition 3 (Strong Type II secure key agreement protocol) [14]. A CLAKA protocol is Strong Type II secure if every probabilistic, polynomial-time adversary $\mathscr{A}$ has negligible advantage in winning the above game subject to the following constraints:

- $\mathscr{A}$ is given the master secret key s at the start of the game,
- $\mathscr{A}$ may corrupt at most one additional type of secret per party participating in the test query,
- $\mathscr{A}$ is allowed to replace public keys of any party; however, this counts as the corruption of one secret,
- $\mathscr{A}$ may not reveal the secret value of any identity for which it has replaced the public key,
- $\mathscr{A}$ is allowed to ask session key reveal queries even for session keys computed by identities where he replaced the identity's public key,

$\mathscr{A}$ is allowed to replace public key of any party after the test query has been issued.

3 Our protocol

In this section, we propose a new CLAKA protocol which is secure against the Type I/II adversary. Our protocol consists of five polynomial time algorithms, i.e. *Setup*, *PartialPrivateKeyExtract*, *SetSecretValue*, *SetPublicKey* **and** *Key−Agreement*. The detail of these algorithms is described as follows.

Setup: Given security parameter k, KGC does the following steps to generate the system parameters and the mast key.

(1) KGC chooses a k -bit prime p, generates an elliptic curve E over finite field F_p, generates a group G of elliptic curve points on E with prime order n and determines a generator P of G.
(2) KGC chooses the master key $mk=s\in Z_n^*$ and computes the master public key $P_{pub}=s\cdot P$.
(3) KGC chooses three cryptographic secure hash functions $H_1:\{0,1\}^*\times G\rightarrow Z_n^*$ $H_2:\{0,1\}^*\times G\times G\times G\rightarrow Z_n^*$ and $H_3:\{0,1\}^*\times\{0,1\}^*\times G\times G\rightarrow\{0,1\}^k$.
(4) KGC publishes $params=\{F_p,E_p,G,P,P_{pub},H_1,H_2,H_3\}$ as system parameters and secretly keeps the master key s.

PartialPrivateKeyExtract: Given *params*, *mk*, and identity *ID* of a user, KGC generates a random number $r_{ID}\in Z_n^*$, computes $R_{ID}=r_{ID}\cdot P$, $h_{ID}=H_1(ID,R_{ID})$ and $s_{ID}=r_{ID}+h_{ID}s \bmod n$. Then KGC returns the partial private key $D_{ID}=(s_{ID},R_{ID})$ to the user.

The user can validate D_{ID} by checking whether the equation $s_{ID}\cdot P=R_{ID}+h_{ID}\cdot P_{pub}$ holds. The partial private key is valid if the equation holds and vice versa.

SetSecretValue: Given *params*, the user with identity *ID* picks a random number $x_{ID}\in Z_n^*$, computes $P_{ID}=x_{ID}\cdot P$ and sets x_{ID} as his secret value.

SetPublicKey: Given *params* and x_{ID}, the user with identity *ID* computes $P_{ID}=x_{ID}P$ and sets P_{ID} as his public key.

Key−Agreement: Assume that an entity A with identity ID_A has partial private key D_A, secret value x_A and public key P_A and an entity B with identity ID_B has partial private key D_B, secret value x_B and public key P_B. If they want to establish a session key, as shown in Fig. 1, the following steps will be executed.

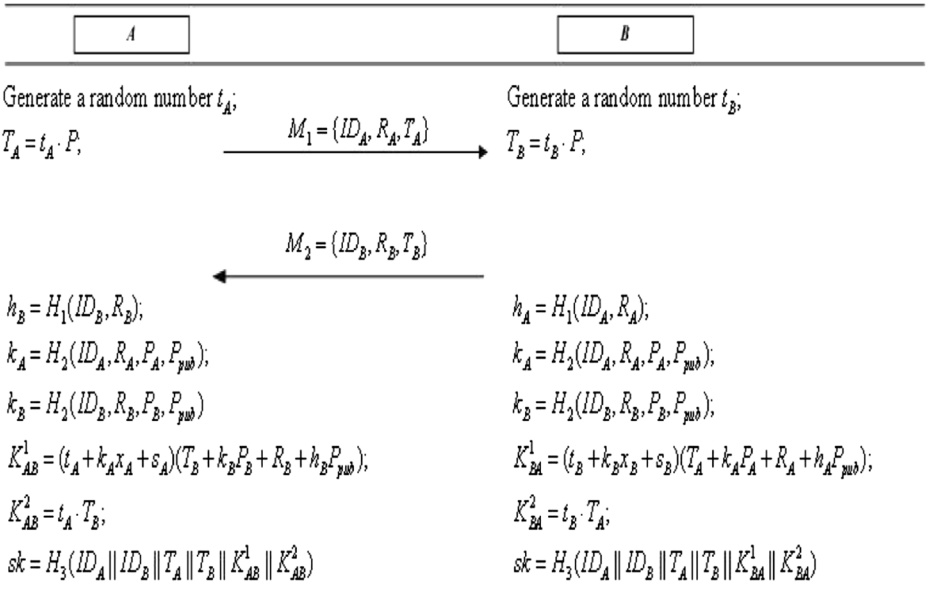

Fig. 1 Key agreement of our protocol

1) A chooses a random number $t_A \in Z_n^*$ and computes $T_A = t_A \cdot P$, then A sends $M_1 = \{ID_A, R_A, T_A\}$ to B.
2) After receiving M_1, B chooses a random number $t_B \in Z_n^*$ and computes $T_B = t_B \cdot P$, $h_A = H_1(ID_A, R_A)$, $k_A = H_2(ID_A, R_A, P_A, P_{pub})$, $k_B = H_2(ID_B, R_B, P_B, P_{pub})$, $K_{BA}^1 = (t_B + k_B x_B + s_B)(T_A + k_A P_A + R_A + h_A P_{pub})$, $K_{BA}^2 = t_B \cdot T_A$, and the session key $sk = H_3(ID_A \| ID_B \| T_A \| T_B \| K_{BA}^1 \| K_{BA}^2)$. At the last, B sends $M_2 = \{ID_B, R_B, T_B\}$ to A.
3) Upon receiving M_2, A computes $h_B = H_1(ID_B, R_B)$, $k_A = H_2(ID_A, R_A, P_A, P_{pub})$, $k_B = H_2(ID_B, R_B, P_B, P_{pub})$, $K_{AB}^1 = (t_A + k_A x_A + s_A)(T_B + k_B P_B + R_B + h_B P_{pub})$, $K_{AB}^2 = t_A \cdot T_B$ and the session key $sk = H_3(ID_A \| ID_B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$.

$$\begin{aligned} K_{AB}^1 &= (t_A + k_A x_A + s_A)\left(T_B + k_B P_B + R_B + h_B P_{pub}\right) \\ &= (t_A + k_A x_A + s_A)(t_B + k_B x_B + s_B)P \\ &= (t_B + k_A x_B + s_B)(t_A + k_B x_A + s_A)P \\ &= (t_B + k_B x_B + s_B)\left(T_A + k_A P_A + R_A + h_B P_{pub}\right) = K_{BA}^1 \end{aligned} \tag{5}$$

and

$$K_{AB}^2 = t_A t_B P = t_B t_A P = K_{BA}^2 \tag{6}$$

Thus, the correctness of the protocol is proved.

4 Security analysis

In this section, we will show our protocol is provably secure in the eCK model. We treat H_1, H_2 and H_3 as three random oracles [3]. For the security, the following theorems are provided.

Theorem 1 If there exists an adversary that has an advantage against our CLAKA protocol $Adv_{\mathcal{A}}(k)$, the challenger $\mathcal{C}$ can use this adversary to solve the CDH problem. We show that the success probability of any adversary against the protocol is limited by $Adv_{\mathcal{A}}(k) \le 9Adv_{\mathcal{C}}^{CDH}(k)$, where $Adv_{\mathcal{C}}^{CDH}(k)$ is the advantage that the challenger gets in solving the CDH problem given security parameter k using the adversary.

Proof From the correctness analysis our protocol, we know that matching sessions compute the same session keys. Like Lippold et al. [14] did, we also do not distinguish two types of adversaries. We will use the similar method proposed by Lippold et al. to show the proposed CLAKA protocol is provably secure in the eCK model.

We assume that an adversary $\mathcal{A}$ against our protocol has a non-negligible advantage $Adv_{\mathcal{A}}(k)$ in winning the game outlined in Section 2.2, where k is the security parameter. Let n_0 and n_1 denote the maximum number of sessions that any one party may have and the maximum number of distinctive honest parties that $\mathcal{A}$ activates separately. Since H_3 is modeled as a random oracle, then $\mathcal{A}$ has only three possible ways to distinguish the tested session key from a random string.

- **Guessing attack**: $\mathcal{A}$ correctly guesses the session key.
- **Key-replication attack**: The adversary $\mathcal{A}$ forces a non-matching session to have the same session key with the test session. In this case, the adversary $\mathcal{A}$ can simply learn the session key by querying the non-matching session.
- **Forging attack**: Assume that $\prod_{I,J}^{X}$ is the test session. At some point in its run, the adversary $\mathcal{A}$ queries H_3 on the value $(ID_A \| ID_B \| T_A \| T_B \| K_{AB}^1 \| K_{AB}^2)$ in the test session owned by party I communicating with party J. Clearly, in this case $\mathcal{A}$ computes the values K_{AB}^1 and K_{AB}^2 itself.

From the similar analysis in [16], we know that the success probability of the guessing attack and the key-replication attack is negligible. Thus, we cannot get an advantage in winning the game against our protocol unless it queries the H_3 oracle on the session key through the forging attack. Then, using the adversary $\mathcal{A}$, we could construct a challenger $\mathcal{C}$ to solve the CDH problem. Let $Adv_{\mathcal{C}}^{CDH}(k)$ be the advantage that the challenger $\mathcal{C}$ gets in solving the CDH problem given the security parameter k using the adversary $\mathcal{A}$. Before the game starts, the challenger $\mathcal{C}$ tries to guess the test session and the strategy that the adversary $\mathcal{C}$ will adopt. $\mathcal{C}$ randomly selects two indexes $I,J \in \{1,2,\ldots,n_1\}$, where $I \neq J$. $\mathcal{C}$ also chooses $X \in \{1,2,\ldots,n_0\}$ and thus determines the test session $\prod_{I,J}^{X}$, which is correct with probability larger than $\frac{1}{n_0 n_1^2}$. $\mathcal{C}$ aborts the game whenever it finds that it has missed its guess. Otherwise, the game proceeds as usual. According to the fresh session definition, $\mathcal{C}$ has the following nine choices for $\mathcal{A}$'s strategy:

1) The adversary may neither learn the secret value of I nor the secret value of J.
2) The adversary may neither learn the ephemeral private key of I nor the ephemeral private key of J.
3) The adversary may neither learn the secret value of J nor replace the public key of J and may also not learn the partial private key of I.
4) The adversary may neither learn the ephemeral private key of J nor the secret value of I.
5) The adversary may neither learn the ephemeral private key of I nor the secret value of J.
6) The adversary may neither learn the secret value of I nor replace the secret value of I and may also not learn the partial private key of J.

7) The adversary may neither learn the ephemeral private key of J nor the partial private key of I.
8) The adversary may neither learn the ephemeral private key of I nor the partial private key of J.
9) The adversary may neither learn the partial private key of I nor of the partial private key J.

Strategy 1. In this strategy, $\mathcal{A}$ could get I's partial private key s_I and ephemeral private key t_I of the test session $\prod_{I,J}^{X}$. $\mathcal{A}$ also could get J's partial private key s_J and ephemeral private key t_J of the session $\prod_{J,I}^{Y}$ where $\prod_{J,I}^{Y}$ is $\prod_{I,J}^{X}$'s matching session. Given a CDH problem instance (P,aP,bP), $\mathcal{C}$'s task is to compute abP using the adversary $\mathcal{A}$. To achieve the goal, $\mathcal{C}$ sets the public key P_I of I to aP and the public key P_J of J to bP. $\mathcal{C}$ uses a proper pairing to check whether the queries of the adversary to the H_3 oracle are validity by checking whether the equation $e(aP,bP)=e(Q,P)$ holds, where $Q = \frac{1}{k_I k_J}\left(K_{I,J}^{1} - (t_I + s_I)\left(T_J + k_J P_J + R_J + h_J P_{pub}\right) - k_I(t_J + s_J)R_I\right)$. As soon as $\mathcal{C}$ finds such a query, $\mathcal{C}$ aborts the game and returns Q as solution of the CDH challenge. The probability that $\mathcal{C}$ is able to find a solution to the CDH challenge is

$$Adv_{\mathcal{C}}^{CDH}(k) \geq \frac{Adv_{\mathcal{A}}(k)}{9n_0 n_1^2}.$$

Strategy 2. In this strategy, $\mathcal{A}$ could get I's secret value x_I and. $\mathcal{A}$ also could get J's partial private key s_J and secret value x_J. Given a CDH problem instance (P,aP,bP), $\mathcal{C}$'s task is to compute abP using the adversary $\mathcal{A}$. To achieve the goal, $\mathcal{C}$ sets the ephemeral public key T_I of $\prod_{I,J}^{X}$ to aP and the ephemeral public key T_J of $\prod_{J,I}^{Y}$ to bP, where $\prod_{J,I}^{Y}$ is $\prod_{I,J}^{X}$'s matching session. $\mathcal{C}$ uses a proper pairing to check whether the queries of the adversary to the H_3 oracle are validity by checking whether the equation $e(aP,bP)=e(K_{I,J}^{2},P)$ holds, $\mathcal{C}$ is able to identify valid queries. As soon as $\mathcal{C}$ finds such a query, $\mathcal{C}$ aborts the game and returns $K_{I,J}^{2}$ as solution of the CDH challenge. The probability that $\mathcal{C}$ is able to find a solution to the CDH challenge is

$$Adv_{\mathcal{C}}^{CDH}(k) \geq \frac{Adv_{\mathcal{A}}(k)}{9n_0 n_1^2}.$$

Strategy 3. In this strategy, $\mathcal{A}$ could get I's secret value x_I and ephemeral private key t_I of the test session $\prod_{I,J}^{X}$. $\mathcal{A}$ also could get J's partial private key s_J and ephemeral private key t_J of the session $\prod_{J,I}^{Y}$'s , where $\prod_{J,I}^{Y}$ is $\prod_{I,J}^{X}$'s matching session. Given a CDH problem instance (P,aP,bP), $\mathcal{C}$'s task is to compute abP using the adversary $\mathcal{A}$. $\mathcal{C}$ sets I's partial public key R_I to $aP - h_I P_{pub}$ and $H_1(ID_I,R_I) \leftarrow h_I$, where h_I is a random number. $\mathcal{C}$ sets the public key P_J of J to bP. $\mathcal{C}$ uses a proper pairing to check whether the queries of the adversary to the H_3 oracle are validity by checking whether the equation $e(aP,bP)=e(Q,P)$ holds, where $Q = \frac{1}{k_J}\left(K_{I,J}^{1} - (t_I + k_I x_I)\left(T_J + k_J P_J + R_J + h_J P_{pub}\right) - (t_J + s_J)aP\right)$. As soon as $\mathcal{C}$ finds such a query, $\mathcal{C}$ aborts the game and returns Q as solution of the CDH challenge. The probability that $\mathcal{C}$ is able to find a solution to the CDH challenge is $Adv_{\mathcal{C}}^{CDH}(k) \geq \frac{Adv_{\mathcal{A}}(k)}{9n_0 n_1^2}$.

Strategy 4. The strategy is symmetric to Strategy 3, its probability of success is equal to the probability of success for Strategy 3. To save space, we will not give the detail here.

Strategy 5. In this strategy, $\mathcal{A}$ could get I's partial private key s_I and secret value x_I. $\mathcal{A}$ also could get J's partial private key s_J and ephemeral private key t_J of the session $\prod_{J,I}^{Y}$, where $\prod_{J,I}^{Y}$ is $\prod_{I,J}^{X}$'s matching session. Given a CDH problem instance (P,aP, bP), $\mathcal{C}$'s task is to compute abP using the adversary $\mathcal{A}$. To achieve the goal, $\mathcal{C}$ sets the ephemeral public key T_I of the test session $\prod_{I,J}^{X}$ to aP and the public key P_J of J to bP. $\mathcal{C}$ uses a proper pairing to check whether the queries of the adversary to the H_3 oracle are validity by checking whether the equation $e(aP,bP)=e(Q,P)$ holds, where $Q = \frac{1}{k_J}\left(K_{I,J}^{1}-(k_I x_I + s_I)\left(T_J + k_J P_J + R_J + h_J P_{pub}\right) - (t_J + s_J)T_I\right)$. As soon as $\mathcal{C}$ finds such a query, $\mathcal{C}$ aborts the game and returns Q as solution of the CDH challenge. The probability that $\mathcal{C}$ is able to find a solution to the CDH challenge is $Adv_{\mathcal{C}}^{CDH}(k) \geq \frac{Adv_{\mathcal{A}}(k)}{9n_0 n_1^2}$.

Strategy 6. The strategy is symmetric to Strategy 5, its probability of success is equal to the probability of success for Strategy 5. To save space, we will not give the detail here.

Strategy 7. In this strategy, $\mathcal{A}$ could get I's secret value x_I and ephemeral private key t_I of the test session $\prod_{I,J}^{X}$. $\mathcal{A}$ also could get J's partial private key s_J and secret value x_J. Given a CDH problem instance (P,aP,bP), $\mathcal{C}$'s task is to compute abP using the adversary $\mathcal{A}$. $\mathcal{C}$ sets I's partial public key R_I to $aP-h_I P_{pub}$ and $H_1(ID_I, R_I)\leftarrow h_I$. $\mathcal{C}$ sets the ephemeral public key T_J of $\prod_{J,I}^{Y}$ to bP. $\mathcal{C}$ uses a proper pairing to check whether the queries of the adversary to the H_3 oracle are validity by checking whether the equation $e(aP,bP)=e(Q,P)$ holds, where $Q=K_{I,J}^{1}-(t_I+k_I x_I)(T_J+k_J P_J+R_J+h_J P_{pub})-(k_J x_J+s_J)aP$. As soon as $\mathcal{C}$ finds such a query, $\mathcal{C}$ aborts the game and returns Q as solution of the CDH challenge. The probability that $\mathcal{C}$ is able to find a solution to the CDH challenge is $Adv_{\mathcal{C}}^{CDH}(k) \geq \frac{Adv_{\mathcal{A}}(k)}{9n_0 n_1^2}$.

Strategy 8. The strategy is symmetric to Strategy 7, its probability of success is equal to the probability of success for Strategy 7. To save space, we will not give the detail here.

Strategy 9. In this strategy, $\mathcal{A}$ could get I's secret value x_I and ephemeral private key t_I of the test session $\prod_{I,J}^{X}$. $\mathcal{A}$ also could get J's secret value x_J and ephemeral private key t_J of the session $\prod_{J,I}^{Y}$ where $\prod_{J,I}^{Y}$ is $\prod_{I,J}^{X}$'s matching session. Given a CDH problem instance (P,aP,bP), $\mathcal{C}$'s task is to compute abP using the adversary $\mathcal{A}$. $\mathcal{C}$ sets I's partial public key R_I to $aP-h_I P_{pub}$ and $H_1(ID_I,R_I)\leftarrow h_I$, where h_I is a random number. $\mathcal{C}$ also sets J's partial public key R_J to $bP-h_J P_{pub}$ and $H_1(ID_J,R_J)\leftarrow h_J$, where h_J is a random number. $\mathcal{C}$ uses a proper pairing to check whether the queries of the adversary to the H_3 oracle are validity by checking whether the equation $e(aP,bP)=e(Q,P)$ holds, where $Q=K_{I,J}^{1}-(t_I+k_I x_I)(T_J+k_J P_J+bP)-(t_J+k_J x_J)aP$. As soon as $\mathcal{C}$ finds such a query, $\mathcal{C}$ aborts the game and returns Q as solution of the

Table 1 Comparisons among different protocols

	Computational cost	Secure in the eCK model
Yang et al.'s protocol [22]	$9T_{mul}+2T_h$	Yes
He et al.'s protocol [11]	$5T_{mul}+3T_{add}+2T_h$	No
Our protocol	$5T_{mul}+3T_{add}+3T_h$	Yes

CDH challenge. The probability that $\mathcal{C}$ is able to find a solution to the CDH challenge is $Adv_{\mathcal{C}}^{CDH}(k) \geq \frac{Adv_{\mathcal{A}}(k)}{9n_0n_1^2}$.

5 Comparison with previous protocols

For the convenience of evaluating the computational cost, we define some notations as follows.

- T_{mul}: The time of executing a scalar multiplication operation of point.
- T_{add}: The time of executing an addition operation of point.
- T_h: The time of executing a one-way hash function.

We will compare the efficiency of our protocol with two latest CLAKA protocols without pairing, i.e. Yang et al.'s protocol [22] and He et al.'s protocol [11]. Table 1 shows the comparison among pairing-free CLAKA protocols in terms of efficiency and security model. Yang et al.'s protocol is implemented in general group. To give fair comparison, we transform it to the elliptic cure group described in Section 2.1. From Table 1, we know that Yang et al.'s protocol [22] and our protocol has advantage in security to He et al.'s protocol [3] since Yang et al.'s protocol [22] and our protocol are provably secure in the eCK model and is secure as long as each party has at least one uncompromised secret. Besides, our protocol has better performance than Yang et al.'s protocol. Moreover, our protocol just needs a more hash function operation than He et al.'s protocol. We conclude that our protocol is more suitable for practical applications.

6 Conclusion

To improve performance, many pairing-free CLAKA protocols have been proposed. In this paper, we proposed a new pairing-free CLAKA protocol and proved its security in the eCK model. The analysis shows our protocol has advantages over previous CLAKA protocols in security or efficiency.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (grant number 2013R1A1A2059864).

References

1. Al-Riyami S, Paterson KG (2003) Certificateless public key cryptography. In: Proc. of ASIACRYPT 2003, LNCS 2894, Springer-Verlag, pp 452–473
2. Bellare M, Pointcheval D, Rogaway P (2000) Authenticated key exchange secure against dictionary attacks. In: Proc.of the EUROCRYPT 2000. LNCS, Springer-Verlag, Vol. 1807, pp 139–55
3. Bellare M, Rogaway P (1993) Entity authentication and key distribution. In: Proc. of the CRYPTO 1993. LNCS, Springer-Verlag, Vol. 773, pp 232–49
4. Bellare M, Rogaway P (1995) Provably secure session key distribution: the three party case. In: Proc. of the 27th ACM symposium on the theory of computing, ACM, pp 57–66

5. Canetti R, Krawczyk H (2001) Analysis of key-exchange protocols and their use for building secure channels. In: Proc. of the EUROCRYPT 2001. LNCS, Springer-Verlag, Vol. 245, pp 453–74
6. Cao X, Kou W (2010) A pairing-free identity-based authenticated Key agreement scheme with minimal message exchanges. Inf Sci 180:2895–2903
7. Chen L, Cheng Z, Smart NP (2007) Identity-based key agreement protocols from pairings. Int J Inf Secur 6: 213–241
8. Geng M, Zhang F (2009) Provably secure certificateless two-party authenticated key agreement protocol without pairing. In: Proc. of International Conference on Computational Intelligence and Security, pp 208–212
9. He D, Chen Y, Chen J, Zhang R, Han W (2011) A new two-round certificateless authenticated key agreement protocol without bilinear pairings. Math Comput Model 54(11–12):3143–3152
10. He D, Chen J, Hu J (2012) A pairing-free certificateless authenticated key agreement protocol. Int J Commun Syst 25(2):221–230
11. He D, Padhye S, Chen J (2012) An efficient certificateless authenticated key agreement protocol. Comput Math Appl 64(6):1914–1926
12. Hou M, Xu Q (2009) A two-party certificateless authenticated key agreement protocol without pairing. In: Proc. of 2nd IEEE International Conference on Computer Science and Information Technology, pp 412–416
13. LaMacchia BA, Lauter K, Mityagin A (2007) Stronger security of authenticated key exchange. In: Proc. of the ProvSection 2007. LNCS, Springer-Verlag , Vol. 4784, pp 1–16
14. Lippold G, Boyd C, Nieto J (2009) Strongly secure certificateless key agreement. In: Pairing 2009, pp 206–230
15. Mandt T, Tan C (2008) Certificateless authenticated two-party key agreement protocols. In: Proc. of the ASIAN 2006, LNCS, Springer-Verlag, Vol. 4435, pp 37–44
16. Ni L, Chen G, Li J, Hao Y (2011) Strongly secure identity-based authenticated key agreement protocols. Comput Electr Eng 37:205–217
17. Shamir A (1984) Identity-based cryptosystems and signature protocols. Proc. CRYPTO1984, LNCS, Vol. 196, pp 47–53
18. Shao Z (2005) Efficient authenticated key agreement protocol using self-certifed public keys from pairings. Wuhan Univ J Nat Sci 10(1):267–270
19. Shi Y, Li J (2007) Two-party authenticated key agreement in certificateless public key cryptography. Wuhan Univ J Nat Sci 12(1):71–74
20. Swanson C (2008) Security in key agreement: Two-party certificateless protocols, Master Thesis, University of Waterloo
21. Wang S, Cao Z, Dong X (2006) Certificateless authenticated key agreement based on the MTI/CO protocol. J Inf Comput Sci 3:575–581
22. Yang G, Tan C (2011) Strongly secure certificateless key exchange without pairing. In: Proc. of 6th ACM Symposium on Information, Computer and Communications Security, pp 71–79
23. Zhang L, Zhang F, Wua Q, Domingo-Ferrer J (2010) Simulatable certificateless two-party authenticated key agreement protocol. Inf Sci 180:1020–1030

Hang Tu received his Ph.D. degree in information security from School of Computer, Wuhan University, Wuhan, China, in 2004. He is currently an associate professor of Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.

Neeraj Kumar is working as an Associate Professor in Department of Computer Science and Engineering, Thapar University, Patiala (Punjab), India. He received his Ph.D. in CSE from Shri Mata Vaishno Devi University, Katra (India) and PDF from Coventry University, Coventry, UK. He has more than 100 research publications in peer-reviewed journals and conferences including IEEE, Elsevier, and Springer. His research is focused on mobile computing, parallel/distributed computing, multiagent systems, service oriented computing, routing and security issues in wireless adhoc, sensor and mesh networks. He is leading the Mobile Computing and Distributed System Research Group. Prior to joining Thapar University, Patiala, he has worked in SMVDU, Katra, HEC Jagadhri and MMEC Mullana, Ambala, Haryana, India. He has delivered invited talks and lectures in various IEEE international conferences in India and abroad. He has organized various special sessions in international conferences in his area of expertise in India and abroad. He is TPC of various IEEE sponsored conferences in India and abroad. He is reviewer/ editorial board of various international journals. He is guest editor of special issue of 6 international journals. He is member of ACM, ACEEE and IACSIT.

Jongsung Kim received his Bachelor and Master degrees in Mathematics from Korea university, Korea in 2000 and 2002, respectively. He received double Doctoral degrees completed in November 2006 and February 2007 at the ESAT/COSIC group of Katholieke Universiteit Leuven and at Engineering in Information Security of Korea University, respectively. He had been a Research Professor of Center for Information Security Technologies (CIST) at Korea University, Korea, from March 2007 till August 2009, and an assistant professor of department of e-business at Kyungnam University, Korea, from September 2009 till February 2013. Dr. Kim has been an assistant professor of department of mathematics at Kookmin University, Korea, since March 2013. Dr. Kim has published about 60 research papers in international journals and conferences. He has been serving as chairs, program committee, or organizing committee chair for many international conferences and workshops. He is editorial board member of International Journal of Information Technology, Communications and Convergence (IJITCC), International Journal of Communication Networks and Distributed Systems (IJCNDS), InderScience and Journal of Convergence (JoC), FTRA Publishing, and Human-centric Computing and Information Sciences (HCIS), Springer. In addition, he has been serving as a Guest Editor for international journals by some publishers:

Springer, Elsevier, John Wiley, Oxford Univ. press, Inderscience. His research interests include security issues, cryptography, ubiquitous computing systems and digital forensics.

Jungtaek Seo Ph.D. received his degree in information security from the graduate school of Information Security, Korea University, in 2006. He is now a senior researcher at the Attached Institute of Electrical and Telecommunication Research Institute, Korea

Currently, as the head of Infrastructure Protection Research Department, Dr. Seo is responsible for research and development of security systems for SCADA, Smart Grid and nuclear power plants. He is a member of the technology subcommittee for Smart Grid road map of Korea, a group to lead the secure operation and construction of nation-wide Smart Grid. Dr. Seo's research interests are being expanded to cyber security in various fields including Smart Grid, I&C systems and nuclear power plants.