Journal of Engineering Science and Technology Review 6 (2) (2013) 35-40

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

A Secure and Efficient Certificateless Short Signature Scheme

Lin Cheng[*] and Qiaoyan Wen

Laboratory of Networking and Switch Technology,Beijing University of Posts and Telecommunications, Beijing 100876, China

Received 15 May 2013; Accepted 25 July 2013

Abstract

Certificateless public key cryptography combines advantage of traditional public key cryptography and identity-based public key cryptography as it avoids usage of certificates and resolves the key escrow problem. In 2007, Huang et al. classified adversaries against certificateless signatures according to their attack power into normal, strong and super adversaries (ordered by their attack power). In this paper, we propose a new certificateless short signature scheme and prove that it is secure against both of the super type I and the super type II adversaries. Our new scheme not only achieves the strongest security level but also has the shortest signature length (one group element). Compared with the other short certificateless signature schemes which have a similar security level, our new scheme has less operation cost.

Keywords: Cryptography, Short signature, Certificateless signature, Bilinear parings

1. Introduction

In traditional public key cryptography, a certification authority (CA) issues a certificate to achieve authentication of the user's public key. Identity-based cryptography proposed by Shamir [1] intended to conquer the problem of certificate management in traditional public key cryptography. In identity-based cryptography, the user's public key is derived directly from its name, email-address or other identity information, but it requires a trusted third party called Key Generation Center (KGC) generate the user's private key. Hence, we are confronted with the key escrow problem. At 2003, Al-Riyami and Paterson [2] introduced certificateless public key cryptography, which resolves the inherent key escrow problem in identity-based cryptography, without requiring certificates as used in traditional public key cryptography. In certificateless public key cryptography, the user's public key is independently generated by the user, and the user's private key is a combination partial private key computed by KGC and some user-chosen secret value, in such a way that the key escrow problem can be eliminated without requiring certificates. In certificateless signature, there exist two different Types of attackers. Huang et al.[3] classified the adversaries against certificateless signatures according to their attack power into normal adversary, strong adversary and super adversary (ordered their attack power). As illustrated in [4, 5], the strong Type I adversary might stand in some particular situations. So, a secure certificateless signature scheme should resist the attack of strong Type I/II adversary at least. In [3], the first certificateless short signature scheme (the signature length of one group element) was proposed, which is secure against the normal Type I and super II adversaries. Shim [4] showed that the certificateless short signatures scheme in [3] is insecure against the strong Type I adversary. Subsequently, Du and Wen [5] presented another short CLS scheme, which was provably secure against the strong Type I adversary and normal Type II adversaries. Choi et al. [6] showed that Du and Wen's certificateless short signature scheme [5] is insecure against the strong Type I adversary and proposed a short CLS scheme which is provably secure against the super Type I and Type II adversaries. Another provably secure short CLS scheme was proposed by Tso et al. [7], however, their defined adversary model is not as powerful as that in Huang et al. [3] and their scheme can not resist the attacks from a strong Type I adversary who can obtain the valid signatures for the replaced public key if he can supply the secret value corresponding to the replaced public key. To the best of our knowledge, Choi et al.'s scheme [6] is the first certificateless short signature scheme which satisfies both the strongest security level and the shortest signature length (one group element). However, their short CLS scheme [6] was proved to be insecure even against the strong Type I adversaries in [8]. In this paper, we propose a new short CLS scheme and prove it is secure against both of the super type I and the super type II adversaries. Compared with the Choi et al.'s scheme [6], our new scheme has less operation cost.

2. Preliminaries

In this section, we introduce the related complexity assumption and security model.

2.1 Computational Diffie-Hellman (CDH)

Given a generator P of an additive cyclic group G_1, and (aP, bP) for some unknown $a, b \in Z_q^*$. The CDH problem is

[*] E-mail address: stonewoods302@163.com
ISSN: 1791-2377

to compute abP. Let C be a probabilistic polynomial time algorithm. We define C 's advantage in solving the CDH problem by $Adv(C) = Pr[B(P, aP, bP) = abP]$.

The CDH assumption states that for every probabilistic polynomial time algorithm C , $Adv(C)$ is negligible.

2.2 Security model

In the security model defined in [3, 6], each super adversary $A \in \{A_I, A_{II}\}$ may issue the following queries.

Extract-Partial-Private-Key (ID). When A supplies an identity ID, challenger C computes the corresponding partial private key D_{ID} for this identity and returns D_{ID} to A .

Extract-Public-Key (ID). When A supplies an identity ID, challenger C returns the corresponding public key Pk_{ID} to A .

Extract-Secret-Value (ID). When A supplies an identity ID, challenger returns x_{ID} to A . Note that, the secret value x_{ID} is used to generate the original public key of ID. If the public key associated with ID has been replaced earlier, A cannot receive any response.

Replace-Public-Key (ID, PK'_{ID}). When A supplies an identity ID and a new valid public key value PK'_{ID}, challenger C replaces the current public key with PK'_{ID}.

Super-Sign (ID, m). When A supplies an identity ID and a message m, challenger C responds with a signature δ such that $1 \leftarrow Verify(params, ID, PK'_{ID}, m, \delta)$, where PK'_{ID} is the current public key corresponding to ID and it may be replaced by the Replace-Public-Key query.

Game I. This game is performed between a challenger C and a Type I adversary A_I for a CLS scheme as follows.

Initialization. Challenger C runs algorithm **Setup** to generate a master secret key msk, and public system parameters $params$. C then gives params to A_I and keeps msk secret. Note that A_I does not know the master key msk.

Queries. In this phase, A_I adaptively performs a polynomially bounded number of oracle queries :Extract-Partial-Private-Key , Extract-Secret-Value, Request-Public-Key, Replace-Public-Key and Super-Sign.

Output. Eventually, A_I outputs (ID^*, m^*, δ^*), where ID^* is the identity of a target user, m^* is a message, and δ^* is a signature for m^*. A_I wins the game if

(1) Extract-Partial-Private-Key (ID^*) query has never been queried.

(2) Super-Sign (ID^*, m^*) query has never been queried.

(3) $1 \leftarrow Verify(params, ID^*, PK_{ID^*}, m^*, \delta^*)$, where PK_{ID^*} which may be replaced by A_I is the current public key of ID^*.

Game II. This game is performed between a challenger C and a Type I adversary A_I for a CLS scheme as follows.

Initialization. Challenger C runs algorithm **Setup** to generate a master secret key msk, and public system parameters $params$. C then gives $params$ and msk to A_{II}.

Queries. In this phase, A_{II} can adaptively issue Extract-Secret-Value, Request-Public-Key and Replace-Public-Key queries to C . In addition, he can also issue only one type of the following queries: Normal-Sign, Strong-Sign, and Super-Sign.

Output. Eventually, A_{II} outputs (ID^*, m^*, δ^*), where ID^* is the identity of a target user, m^* is a message, and δ^* is a signature for m^*. A_{II} wins the game if

(1) Extract-Partial-Private-Key (ID^*) query has never been queried.

(2) Super-Sign (ID^*, m^*) query has never been queried.

(3) $1 \leftarrow Verify(params, ID^*, PK_{ID^*}, m^*, \delta^*)$, where PK_{ID^*} is the original public key of ID^*.

Definition 1. We say that a CLS scheme is existentially unforgeable, if no polynomially bounded adversaries A (A_I and A_{II}) have non-negligible advantage of winning the above game.

3. New Short Certificateless Signature Scheme

In this section, we propose a new short CLS scheme which is secure against the super Type I/II adversary.

3.1 Construction

Setup. Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of prime order q and P is a generator of G_1. Given a bilinear pairing $e: G_1 \times G_1 \to G_2$ and three distinct hash functions H_1, H_2 and H_3 : $H_1: \{0,1\}^* \to Z_q^*$, $H_2: \{0,1\}^* \to G_1$, $H_3: \{0,1\}^* \to G_1$. The KGC selects $s \in Z_q^*$ uniformly as master-key and sets $P_{pub} = sP$.The public parameters list $params = \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3\}$. The master secret key $msk = s$.

Partial-Private-Key-Extract: On input $params$, master key s, $ID \in \{0,1\}^*$, KGC carries out the following for generating a partial private d_{ID} for a user with identity ID.

Choose at random $r \in Z_q^*$, compute $R_{ID} = rP$, $h_1 = H_1(R_{ID}, ID)$ and $d_{ID} = r + h_1 s \bmod q$. Return (R_{ID}, d_{ID}) to the user. The user can check its correctness by checking whether $d_{ID}P = R_{ID} + h_1 P_{pub}$.

Set-Secret-Value. The user selects a random value $x_{ID} \in Z_q^*$ as his secret key.

Set-Public-Key. The user computes $Y_{ID} = x_{ID}P$, then sets his public key $PK_{ID} = (R_{ID}, Y_{ID})$.

CL-Sign. On inputs $params$, a message $m \in \{0,1\}^*$, signer's identity ID and his partial private d_{ID} and secret key x_{ID}, the signer computes $\delta = d_{ID} H_2(m, ID, R_{ID}, Y_{ID}) + x_{ID} H_3(m, ID, R_{ID}, Y_{ID})$.

CL-Verify. Given $params$, PK_{ID}, message m, signer's identity ID and signature δ, the verifier computes

$$h_1 = H_1(R_{ID}, ID), h_2 = H_2(m, ID, R_{ID}, Y_{ID}), h_3 = H_3(m, ID, R_{ID}, Y_{ID}).$$

Accept the signature if the following equation holds: $e(\delta, P) = e(R_{ID} + h_1 P_{pub}, h_2) e(Y_{ID}, h_3)$.

3.2 Proof of security

Theorem 1. The proposed certificateless signature scheme is existential unforgeable against a super adversary A_I under the CDH assumption.

Proof. Suppose there exists a super Type I adversary A_I which has advantage ε in attacking our short CLS scheme. We want to build an algorithm C that uses A_I to solve the CDH problem. Suppose that C is given (P, aP, bP) as an instance of the CDH problem. Its goal is to compute abP. C will run A_I as a subroutine and act as A_I's challenger. We describe the simulation as follows.

Initialization. C sets $P_{pub} = aP$ and provides A_I with $\{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3\}$ as public parameters, where H_1, H_2, H_3 are random oracles controlled by C.

Queries. In the query phase, C responds A_I 's queries as follows:

H_1 **query**: C maintains a H_1 list of tuples (ID_i, R_{ID_i}, t_{1i}). When A_I makes H_1 query on (ID_i, R_{ID_i}), C looks up the H_1 list and does the following:

1. If H_1 list contains (ID_i, R_{ID_i}, t_{1i}), C returns t_{1i} to A_I.

2. Otherwise, C picks $t_{1i} \in Z_p^*$ at random, adds (ID_i, R_{ID_i}, t_{1i}) to H_1 list and returns t_{1i} to A_I.

H_2 **query**: C maintains a H_2 list of tuples $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$. When A_I makes H_2 query on $(m_i, ID_i, R_{ID_i}, Y_{ID_i})$, C loos up the H_2 list and does the following:

1. If H_2 list contains $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$, C returns T_{2i} to A_I.

2. Otherwise, C picks $t_{2i} \in Z_p^*$ at random, computes $T_{2i} = t_{2i} bP$ and adds $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$ to H_2 list and returns T_{2i} to A_I.

H_3 **query**: C maintains a H_3 list of tuples $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$. When A_I makes H_3 query on $(m_i, ID_i, R_{ID_i}, Y_{ID_i})$, C looks up the H_3 list and does the following:

1. If H_3 list contains $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$, C returns T_{3i} to A_I.

2. Otherwise, C picks $t_{3i} \in Z_p^*$ at random, computes $T_{3i} = t_{3i} P$ and adds $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$ to H_3 list and returns T_{3i} to A_I.

Extract-Partial-Private-Key (ID_i) **query**: C maintains a partial key list of tuples $(ID_i, R_{ID_i}, d_{ID_i})$. Suppose A_I makes at most q_{pp} queries to the partial private key extraction oracle. First, C chooses $j \in [1, q_{pp}]$ randomly. When A_I makes a partial key extraction query on ID_i.

1. If $i = j$ (we let $ID_i = ID^*$ at this point), then C outputs "failure" and halts because it is unable to coherently answer the query.

2. Otherwise ($i \neq j$), C looks up the partial key list. If partial key list contains $(ID_i, R_{ID_i}, d_{ID_i})$, C returns (R_{ID_i}, d_{ID_i}) to A_I. Otherwise, C chooses $t_{1i}, d_i \in Z_q^*$ at random, and sets $R_{ID_i} = d_i P - t_{1i} P_{pub}$, $d_{ID_i} = d_i$, $H_1(R_{ID_i}, ID_i) = t_{1i}$. C adds (ID_i, R_{ID_i}, t_{1i}) to H_1 list and $(ID_i, R_{ID_i}, d_{ID_i})$ to partial key list, and returns (R_{ID_i}, d_{ID_i}) to A_I. Note that (R_{ID_i}, d_{ID_i}) is a validly partial private key for the identity ID_i since it satisfies the equation $d_{ID_i} P = R_{ID_i} + h_1(R_{ID_i}, ID_i) P_{pub}$.

Request-Public-Key (ID_i) **query**: C maintains a public key list of tuples $(ID_i, (R_{ID_i}, Y_{ID_i}))$. When A_I makes a public key request query on ID_i, C looks up public key list and does the following:

1. If public key list contains $(ID_i, (R_{ID_i}, Y_{ID_i}))$, C returns $PK_{ID_i} = (R_{ID_i}, Y_{ID_i})$ to A_I.

2. Otherwise, C does the following:

(a) If partial key list contains $(ID_i, R_{ID_i}, d_{ID_i})$, C picks $x_{ID_i} \in Z_p^*$ at random and computes $Y_{ID_i} = x_{ID_i} P$; adds (ID_i, x_{ID_i}) to secret value list and $(ID_i, R_{ID_i}, Y_{ID_i})$ to public key list; returns $PK_{ID_i} = (R_{ID_i}, Y_{ID_i})$ to A_I.

(b) Otherwise, C gets a partial key (R_{ID_i}, d_{ID_i}) by making partial key extraction query on ID_i; then C picks $x_{ID_i} \in Z_p^*$ at random and computes $Y_{ID_i} = x_{ID_i} P$ and adds $(ID_i, R_{ID_i}, d_{ID_i})$ to partial key list and (ID_i, x_{ID_i}) to secret value list and $(ID_i, R_{ID_i}, Y_{ID_i})$ to public key list; finally C returns $PK_{ID_i} = (R_{ID_i}, Y_{ID_i})$ to A_I.

Replace-Public-Key (ID_i, PK'_{ID_i}) **query**: When A_I makes this query on ID_i, if public key list contains PK'_{ID_i}, C sets $PK_{ID_i} = PK'_{ID_i}$. Otherwise, C makes a secret value query on ID_i, C then sets $PK_{ID_i} = PK'_{ID_i}$.

Extract-Secret-Value (ID_i) **query**: C maintains a secret value list of tuples (ID_i, x_{ID_i}). When A_I makes this query on ID_i, C looks up secret value list and does the following.

1. If the secret value list contains ID_i, x_{ID_i}, C returns x_{ID_i}.

2. Otherwise, C picks $r_{ID_i}, x_{ID_i} \in Z_p^*$ at random and computes $R_{ID_i} = r_{ID_i}P, Y_{ID_i} = x_{ID_i}P$; adds (ID_i, x_{ID_i}) to secret value list and $(ID_i, R_{ID_i}, Y_{ID_i})$ to public key list. C then returns x_{ID_i}.

Super-Sign (m_i, ID_i) **query**: When A_I makes this query on (ID_i, m_i), C performs as follows:

1. If $ID_i = ID^*$, C picks two random values $t_{1i}, t_{3i} \in Z_p^*$, sets $h_2 = -R_{ID} - t_{1i}aP$ and $\delta = t_{3i}Y_{ID_i}$ (C halts and outputs " failure " if H_2 turns out to have already been defined for $(m_i, ID_i, R_{ID_i}, Y_{ID_i})$) . C then returns δ and adds (ID_i, R_{ID_i}, t_{1i}) , $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$ to H_1 list, H_3 list, respectively.

2. Otherwise, C picks two random values $t_{2i}, t_{3i} \in Z_p^*$ and computes $\delta = d_{ID_i}t_{2i}bP + t_{3i}Y_{ID}$. C then returns δ and adds $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$, $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$ to H_2 list, H_3 list, respectively.

Output: Eventually, A_I outputs a forgery signature δ^* on message m^* with respect to (ID^*, PK_{ID^*}). If $ID^* \neq ID_j$, then C outputs " failure " and stops. Otherwise, C finds out an item $(ID^*, R_{ID^*}, t_{1i}^*)$ in the H_1 list , an item $(m^*, ID^*, R_{ID^*}, Y_{ID^*}, t_{2i}^*, T_{2i})$ in the H_2 list, and an item $(m^*, ID^*, R_{ID^*}, Y_{ID^*}, t_{3i}^*, T_{3i})$ in the H_3 list. Note that the list H_1 , H_2 , H_3 must contain such entries with overwhelming probability (otherwise C outputs " failure " and stops). Note that $H_1(R_{ID^*}, ID^*) = t_{1i}^*P$, $H_2(m^*, ID^*, R_{ID^*}, Y_{ID^*}) = t_{2i}^*bP$, $H_3(m^*, ID^*, R_{ID^*}, Y_{ID^*}) = t_{3i}^*P$. If A_I succeeds in the game, then

$$\begin{aligned} e(\delta^*, P) &= e(R_{ID_i^*} + h_1^*P_{pub}, h_2^*)e(Y_{ID_i^*}, h_3^*) \\ &= e(R_{ID_i^*}, h_2^*)e(h_1^*P_{pub}, h_2^*)e(Y_{ID_i^*}, h_3^*) \\ &= e(R_{ID_i^*}, t_{2i}^*bP)e(t_{1i}^*aP, t_{2i}^*bP)e(Y_{ID_i^*}, t_{3i}^*P) \end{aligned} \tag{1}$$

Using Forking Lemma [9], after replaying A_I with the same random tape, C obtains another valid signed message (m^*, δ'^*). The message will satisfy

$$\begin{aligned} e(\delta^{*'}, P) &= e(R_{ID_i^*} + h_1^{*'}P_{pub}, h_2^{*'})e(Y_{ID_i^*}, h_3^{*'}) \\ &= e(R_{ID_i^*}, h_2^{*'})e(h_1^{*'}P_{pub}, h_2^{*'})e(Y_{ID_i^*}, h_3^{*'}) \\ &= e(R_{ID_i^*}, t_{2i}^{*'}bP)e(t_{1i}^{*'}aP, t_{2i}^{*'}bP)e(Y_{ID_i^*}, t_{3i}^{*'}P) \end{aligned} \tag{2}$$

From the Eqs. (1) and (2), C can obtain the solution of the CDH problem by computing

$$abP = \frac{t_{2i}^{*'}\delta^* - t_{2i}^*\delta^{*'} - (t_{2i}^{*'}t_{3i}^* - t_{2i}^*t_{3i}^{*'})Y_{ID_i^*}}{t_{2i}^{*'}t_{1i}^*t_{2i}^* - t_{2i}^*t_{1i}^{*'}t_{2i}^{*'}}.$$

Theorem 2. The proposed certificateless signature scheme is existential unforgeable against a super adversary A_{II} under the CDH assumption.

Proof. Suppose there exists a super Type II adversary A_{II} which has advantage ε in attacking our short CLS scheme. We want to build an algorithm C that uses A_{II} to solve the CDH problem. Suppose that C is given (P, aP, bP) as an instance of the CDH problem. Its goal is to compute abP. C will run A_{II} as a subroutine and act as A_{II}'s challenger. We describe the simulation as follows.

Initialization. C picks a random $s \in Z_q^*$ and sets $P_{pub} = sP$,where s is the master key. C gives system parameters with master key to A_{II}.

Queries. In the query phase, C responds A_{II} 's queries as follows:

H_1 **Queries**: C maintains a H_1 list of tuples (ID_i, R_{ID_i}, t_{1i}). When A_{II} makes H_1 query on (ID_i, R_{ID_i}), C looks up the H_1 list and does the following:

1. If H_1 list contains (ID_i, R_{ID_i}, t_{1i}), C returns t_{1i} to A_{II}.

2. Otherwise, C picks $t_{1i} \in Z_p^*$ at random, and adds (ID_i, R_{ID_i}, t_{1i}) to H_1 list and returns t_{1i} to A_{II}.

H_2 **Queries**: C maintains a H_2 list of tuples $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$. When A_{II} makes H_2 query on $(m_i, ID_i, R_{ID_i}, Y_{ID_i})$, C looks up the H_2 list and does the following:

1. If H_2 list contains $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$, C returns $t_{2i}P$ to A_{II}.

2. Otherwise, C picks $t_{2i} \in Z_p^*$ at random, and computes $T_{2i} = t_{2i}P$ and adds $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{2i}, T_{2i})$ to H_2 list and returns T_{2i} to A_{II}.

H_3 **Queries**: C maintains a H_3 list of tuples $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$. When A_{II} makes H_3 query on $(m_i, ID_i, R_{ID_i}, Y_{ID_i})$, C looks up the H_3 list and does the following:

1. If H_3 list contains $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$, C returns T_{3i} to A_{II}.

2. Otherwise, C picks $t_{3i} \in Z_p^*$ at random, and computes $T_{3i} = t_{3i}aP$ and adds $(m_i, ID_i, R_{ID_i}, Y_{ID_i}, t_{3i}, T_{3i})$ to H_3 list and returns T_{3i} to A_{II}.

Request-Public-Key (ID_i) **query**: C maintains a public key list of tuples $(ID_i, (R_{ID_i}, Y_{ID_i}))$. Suppose A_{II} makes at most q_{pk} queries to the public key request oracle. First, C chooses $j \in [1, q_{pk}]$ randomly. When A_{II} makes a public key request query on ID_i, C looks up the public key list and does the following:

1. If public key list contains $(ID_i,(R_{ID_i},Y_{ID_i}))$, C returns $PK_{ID_i}=(R_{ID_i},Y_{ID_i})$ to A_{II}.

2. If $i=j$ (we let $ID_i=ID^*$ at this point), C sets $Y_{ID_i}=bP$ and picks r_{ID_i} at random and computes $R_{ID_i}=r_{ID_i}P$, finally C returns $PK_{ID_i}=(R_{ID_i},Y_{ID_i})$ to A_{II}.

3. Otherwise ($i\neq j$), C picks $r_{ID_i},x_{ID_i}\in Z_p^*$ at random and computes $R_{ID_i}=r_{ID_i}P,Y_{ID_i}=x_{ID_i}P$; adds (ID_i,x_{ID_i}) to secret value list and (ID_i,R_{ID_i},Y_{ID_i}) to public key list; returns $PK_{ID_i}=(R_{ID_i},Y_{ID_i})$ to A_{II}.

Replace-Public-Key (ID_i,PK'_{ID_i}) **query**: When A_{II} makes this query on ID_i, if public key list contains PK'_{ID_i}, C sets $PK_{ID_i}=PK'_{ID_i}$. Otherwise, C makes a public key query on ID_i, C then sets $PK_{ID_i}=PK'_{ID_i}$.

Extract-Secret-Value (ID_i) **query**: C maintains a secret value list of tuples $(ID_i,,x_{ID_i})$. When A_{II} makes this query on ID_i, C does the following:

1. Run the public key request taking ID_i as input to get a tuple (ID_i,R_{ID_i},Y_{ID_i}).

2. If $i\neq j$, search secret value list (ID_i,x_{ID_i}) to get x_{ID_i}, and then return $SK_{ID_i}=(d_{ID_i},x_{ID_i})$ to A_{II}.

3. Otherwise, return “ failure ” and terminate.

Super-Sign (m_i,ID_i) **query**: When A_{II} makes this query on (ID_i,m_i), C first finds (ID_i,R_{ID_i},Y_{ID_i}) from public key list, then performs as follows:

1. If $ID_i\neq ID^*$, C picks two random values $t_{2i},t_{3i}\in Z_p^*$ and computes $\delta=d_{ID_i}t_{2i}P+t_{3i}bP$. C then returns δ and adds $(m_i,ID_i,R_{ID_i},Y_{ID_i},t_{2i},T_{2i})$, $(m_i,ID_i,R_{ID_i},Y_{ID_i},t_{3i},T_{3i})$ to H_2 list, H_3 list, respectively.

2. Otherwise, C picks two random values $t_{2i},t_{3i}\in Z_p^*$ and computes $\delta=d_{ID_i}t_{2i}P+t_{3i}Y_{ID}$. C then returns δ and adds $(m_i,ID_i,R_{ID_i},Y_{ID_i},t_{2i},T_{2i})$, $(m_i,ID_i,R_{ID_i},Y_{ID_i},t_{3i},T_{3i})$ to H_2 list, H_3 list, respectively.

Output. Eventually, A_{II} outputs a forgery signature δ^* on message m^* with respect to (ID^*,PK_{ID^*}). If $ID^*\neq ID_j$, C outputs “ failure ” and stops. The following equation holds because the signature is valid.

$$\begin{aligned} e(\delta^*,P) &= e(R_{ID_i^*}+h_1^*P_{pub},h_2^*)e(Y_{ID_i^*},h_3^*) \\ &= e(R_{ID_i^*}+t_{1i}^*P_{pub},t_{2i}^*P)e(bP,t_{3i}^*aP) \end{aligned}$$

C can obtain the solution of the CDH problem as $abP=(t_{3i}^*)^{-1}(\delta^*-t_{2i}^*(R_{ID_i^*}+t_{1i}^*P_{pub}))$.

3.3. Performance Analysis

The existing certificateless short signature scheme (the signature length of one group element) can be provably secure against both of the super type I and the super type II adversaries is proposed by Choi et al [6]. Table 1 summarizes the comparisons of our scheme with Choi et al.’s scheme [6] in the signature and verification stages. H denotes the Hash function operation, e denotes a pairing operation, and P denotes the scalar multiplication operation.

Table 1 Efficiency comparison

Schemes	Hash	Pairing	scalar multiplication
[6]	8 H	3 e	5 P
Ours	5 H	3 e	3 P

From Table 1, we know our scheme is more efficient than Choi et al.’s scheme [6].

4. Conclusion

In this paper, we propose a new short CLS scheme and prove its security in the random oracle model under the computational Diffie-Hellman assumption. Our new scheme satisfies both the strongest security level and the shortest signature length (one group element). Compared with the short CLS scheme proposed by Choi et al. [6] which has a similar security level, our new scheme has less operation cost. Thus, our scheme can be applied in low bandwidth communication, low storage and low computation environments.

Acknowledgments

We would like to thank the anonymous reviewers for giving valuable comments. This work is supported by NSFC (Grant No. 61272057, 61202434, 61170270, 61100203, 61003286, 61121061), the Fundamental Research Funds for the Central Universities (Grant No. 2012RC0612, 2011YB01).

References

1. Shamir, A., “Identity-based cryptosystems and signature schemes”, In: Advances in Cryptology-Crypto 1984, LNCS, vol. 196, Springer-Verlag, Berlin, 1984, pp. 47–53.
2. Al-Riyami, S. and Paterson, K., “Certificateless public key cryptography”, In: Advances in Cryptology-Asiacrypt 2003, LNCS, vol.2894, Springer-Verlag, Berlin, 2003, pp. 452–473.
3. Huang, X., Mu, Y., Susilo, W., Wong, D., and Wu, W., “Certificateless signature revisited”, In ASISP 2007, vol. 4586, Springer. LNCS, vol.4586, Springer-Verlag, Berlin, 2007, pp. 308–322.
4. Shim, K. A., “Breaking the short certificateless signature scheme”. Information Sciences, 179(3), 2009, pp.303–306.

5. Du, H. and Wen, Q., "Efficient and provably-secure certificateless short signature scheme from bilinear pairings", Computer Standards and Interfaces, 31(2), 2009, pp.390–394.
6. Choi, K.Y., Park, J. H. and Lee. D. H., "A new provably secure certificateless short signature scheme", Computers & Mathematics with Applications, 61(7), 2011, pp.1760–1768.
7. Tso, R., Huang, X. and Susilo, W., "Strongly secure certificateless short signatures", The Journal of Systems and Software, 85, 2013, pp.1409–1407.
8. Chen,Y.C., Tso, R. and Horng, G., "Cryptanalysis of a provablysecure certificateless short signature scheme", In Advances in Intelligent system & Applications, SIST 21, Springer-Verlag, Berlin, 2013, pp.61-68.
9. Pointcheval, D. and Stern, J., "Security arguments for digital signatures and blind signatures", Journal of Cryptology, 13(3),2000, pp.361–369.