Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2014, Article ID 792063, 6 pages
http://dx.doi.org/10.1155/2014/792063

Research Article

Pairing-Free Certificateless Signature with Security Proof

Wenhao Liu, Qi Xie, Shengbao Wang, Lidong Han, and Bin Hu

Hangzhou Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, Hangzhou, Zhejiang 311121, China

Correspondence should be addressed to Qi Xie; qixie@126.com

Received 30 May 2014; Revised 6 November 2014; Accepted 6 November 2014; Published 26 November 2014

Academic Editor: Tzonelih Hwang

Since certificateless public key cryptosystem can solve the complex certificate management problem in the traditional public key cryptosystem and the key escrow problem in identity-based cryptosystem and the pairing computation is slower than scalar multiplication over the elliptic curve, how to design certificateless signature (CLS) scheme without bilinear pairings is a challenge. In this paper, we first propose a new pairing-free CLS scheme, and then the security proof is presented in the random oracle model (ROM) under the discrete logarithm assumption. The proposed scheme is more efficient than the previous CLS schemes in terms of computation and communication costs and is more suitable for the applications of low-bandwidth environments.

1. Introduction

In 2003, Al-Riyami and Paterson [1] first introduced the concept of certificateless public key cryptosystem (CL-PKC). The basic idea of CL-PKC is to construct the user's public/private key pair by combining a master key of the key generation center (KGC) with a random secret value generated by the user. Hence, the KGC is unable to compute the user's private key, and each user has one additional random public key and this public key does not need to be certified by a trusted third party in CL-PKC. Thus, CL-PKC not only eliminates the certificates in PKC but also solves the key escrow problem in identity-based public key cryptosystem (ID-PKC). As a classical signature scheme, it should provide existential unforgeability, which ensures that the adversary cannot forge a valid signature. In the formal security model of CLS scheme, two types of adversaries should be considered. Type I adversary is allowed to replace user's public key; however, it cannot access the master key of KGC, while Type II adversary is allowed to know the master key of KGC but cannot replace the target user's public key.

The first CLS scheme was proposed by Al-Riyami and Paterson [1]. Following their works, Huang et al. [2] pointed out that Al-Riyami et al.'s scheme is insecure against Type I adversary. Later, Yum and Lee [3] presented a generic construction of CLS scheme. Hu et al. [4] demonstrated that their scheme is also insecure against Type I adversary. Gorantla and Saxena [5] proposed an efficient CLS scheme, but Cao et al. [6] showed that their scheme is insecure against Type I adversary. Since then, many CLS schemes [7–15] have been proposed. However, only few of them, for example, [10–12], are secure against these two types of adversaries; others are vulnerable to the key replacement attack.

With the rapid development of wireless network technology, more and more users use their mobile devices to deal with the transactions. However, almost all of the above-mentioned schemes cannot be used in the low-bandwidth communication and low-storage and less computation environments. Therefore, many researchers tried to design short CLS schemes. In 2007, Huang et al. [16] proposed the first CLSS scheme. After that, Du and Wen [17] and Choi et al. [18] proposed CLSS schemes, respectively. Unfortunately, all of them are vulnerable to the key replacement attack launched by a Type I adversary; what is more, they still need bilinear pairing computations. As we know, the computation cost of a pairing is approximately 20 times higher than that of the scalar multiplication over an elliptic curve group [19, 20]. Therefore, how to design a pairing-free CLSS scheme is an attractive topic.

Recently, He et al. [21] proposed an efficient certificateless signature scheme without pairings; Tian and Huang [22] and Tsai et al. [23] pointed out that the scheme cannot withstand a Type II adversary's attack. Tsai et al. [23] also proposed an improved new scheme in order to enhance security. Gong and Li [24] pointed out the new scheme is insecure against the super adversary in the random oracle model, and they proposed a real CLS scheme and demonstrated that their scheme is secure against the supper adversary. Yeh et al. [25] proposed a secure certificateless signature scheme without pairings in 2014. In this paper, we propose a new CLS scheme without bilinear pairings, and is provable secure in the random oracle model (ROM) under the discrete logarithm assumption.

The rest of this paper is organized as follows. In Section 2, we present the preliminaries including the elliptic curve, bilinear pairings, some hard problems, and complexity assumptions. A new CLS scheme and the security proof are presented in Sections 3 and 4, respectively. After that, we show the performance comparison among our scheme and other related schemes in Section 5. Finally, we conclude the paper in Section 6.

2. Preliminaries

In this section, we present some definitions and assumptions which are needed in the rest of the paper.

2.1. Elliptic Curve. Let the symbol E/F_p be an elliptic curve E over a prime finite field F_p; an equation $y^2 = x^3 + ax + b$, $a, b \in F_p$ with the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$. The point on E/F_p together with an extra point O, called the point at infinity, forms a group $G = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \cup \{O\}$. We define $tP = P + P + \cdots + P$ (t times) as scalar multiplication. Let q be the order of G.

2.2. Bilinear Pairings. Let G be a cyclic additive group of prime order q and let G_2 be a cyclic multiplicative group with the same order q; P is a generator. A bilinear pairing is a map $e : G \times G \rightarrow G_2$ with the following properties.

(1) Bilinearity: if $P, Q, R \in G$, then $e(P + Q, R) = e(P, R)e(Q, R)$.

(2) Nondegeneracy: there exists a $P \in G$ such that $e(P, P) \neq I_{G_2}$, where I_{G_2} is the identity element of G_2.

(3) Computability: there exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G$.

2.3. Hard Problems and Complexity Assumptions. Discrete logarithm problem (DLP): let P be a generator of group G. Given a tuple $(P, xP) \in G$ for $x \in Z_q^*$, it is hard to compute x.

DL Assumption. There exists no algorithm running in expected polynomial time, which can solve the DLP with nonnegligible probability.

2.4. Security Model. The security model of CLS can be referred to [3, 7, 8]. There are two types of adversaries for a CLS scheme, a *Type I* adversary A_1 and a *Type II* adversary A_2. A_1 represents an attacker who is not allowed access to the master key of KGC but he may replace public keys. A_2 represents an attacker who is allowed access to the master key of KGC but he cannot replace public keys.

In general, we use two games to define the existential unforgeability of a CLS scheme against a *Type I* adversary A_1 and a *Type II* adversary A_2.

Game 1. A challenger C takes a security parameter k and generates a master private key s and public parameter *params* and then sends *params* to A_1 and keeps s secret. A_1 executes the game according to the following steps.

Create(ID). On input an identity $\mathrm{ID} \in \{0,1\}^*$, if ID has already been created, nothing is to be carried out. Otherwise, C generates the public/private key pair $(\mathrm{PK_{ID}}, \mathrm{SK_{ID}})$.

Public-Key(ID). On input an identity ID, C outputs the public key $\mathrm{PK_{ID}}$ to A_1.

Partial-Private-Key-Extract. On input an identity ID, C outputs the partial key D_{ID}.

Set-Secret-Key. On input a user's identity ID, C outputs the private key $\mathrm{SK_{ID}} = (x_{\mathrm{ID}}, D_{\mathrm{ID}})$.

Public-Key-Replacement(ID,PK'_{ID}). For a participant whose identity is ID_i, A_1 chooses a new public key $\mathrm{PK'_{ID}}$ and then sets $\mathrm{PK'_{ID}}$ as the new public key of this participant. C will record this replacement, which will be used later.

Sign(ID,m). On input $(\mathrm{ID}, m, \mathrm{PK_{ID}})$, C uses the private key $(x_{\mathrm{ID}}, D_{\mathrm{ID}})$ to compute the signature σ and returns it to A_1. If the public key $\mathrm{PK_{ID}}$ has been replaced by A_1, then C cannot find $(x_{\mathrm{ID}}, D_{\mathrm{ID}})$; the answer of the signing oracle may be incorrect. In this situation, we assume that C submits a secret value r corresponding to the replaced public key $\mathrm{PK_{ID}}$ to the signing oracle.

At the end, A_1 outputs a signature σ on the message m corresponding to the public key $\mathrm{PK_{ID^*}}$ for an identity ID^*, which is the challenge identity. A_1 wins the game if the following conditions hold.

(1) $\mathrm{Verify}(params, \mathrm{ID}, m, \mathrm{PK_{ID^*}}, \sigma) = 1$.

(2) (ID^*, m) has never been submitted to the oracle *Sign*.

(3) ID^* has never been submitted to the *Partial-Private-Key-Extract* query or *Set-Secret-Key* query.

If A_1 has the advantage at least ε in the above game, runs in time at most t, and makes at most q_C *Create*(ID) queries, q_S *Sign* queries, and q_H *hash* queries, respectively, then A_1 is said to be an $(\varepsilon, t, q_C, q_S, q_H)$-forger. If there exists no such forger, then a signature scheme is said to be $(\varepsilon, t, q_C, q_S, q_H)$-secure against Type I adversary.

Game 2. A challenger C is playing the game with Type II adversary A_2.

A challenger C takes a security parameter k, generates a master private key s and public parameter *params*, and then sends *params* and s to A_2. C answers *Create*(ID), *Public-Key*(ID), *Set-Secret-Key*, *Partial-Private-Key-Extract,* and *Sign*(ID, m) queries from A_2, like does in *Game 1.*

At the end, A_2 outputs a signature σ on the message m corresponding to the public key PK_{ID^*} for an identity ID^*, which is the challenge identity. A_2 wins the game if the following conditions hold.

(1) $\text{Verify}(params, ID, m, PK_{ID^*}, \sigma) = 1$.

(2) (ID^*, m) has never been submitted to the oracle *Sign*.

(3) ID^* has never been submitted to the *Set-Secret-Key* query.

If A_2 has the advantage at least ε in the above game, runs in time at most t, and makes at most q_C *Create*(ID) queries, q_S *Sign* queries, and q_H *hash* queries, respectively, then A_2 is said to be an $(\varepsilon, t, q_C, q_S, q_H)$-forger. If there exists no such forger, then a signature scheme is said to be $(\varepsilon, t, q_C, q_S, q_H)$-secure against Type II adversary.

3. The Proposed CLS Scheme

In this section, we propose a new CLS scheme, which consists of seven algorithms: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Key*, *Set-Private-Key*, *Set-Public-Key*, *Sign*, and *Verify*. The details are described as follows.

Setup. On input a security parameter k, and return system parameters and master key:

(1) KGC generates a cyclic additive group G and a cyclic multiplicative group G_2 with the same order q;

(2) KGC chooses a generator $P \in G$ and two cryptographic secure hash functions $H_1 : \{0,1\}^* \times G \rightarrow Z_q^*$ and $H_2 : \{0,1\}^* \rightarrow Z_q^*$;

(3) KGC picks a random master private key $x \in Z_q^*$ and computes $P_{pub} = xP$;

(4) KGC publishes system parameters $params = \{e, P, G, G_2, H_1, H_2, P_{pub}\}$ and keeps x secret.

Partial-Private-Key-Extract. On input *params*, system master key x and a random value r_A, user A's identity ID_A, compute $R_A = r_A P$, $Q_A = H_1(ID_A, R_A)$, and $D_A = r_A + xQ_A$, output D_A through a secret channel, and publish R_A.

Set-Secret-Key. On input the security parameter k and a user's identity ID_A, compute $X_A = x_A P$ and set x_A as his secret value.

Set-Private-Key. On input x_A, D_A and a user's identity ID_A, output the user's private key (x_A, D_A).

Set-Public-Key. On input a user's identity ID_A, output the user's public key (X_A, Q_A).

Sign. On input *params*, ID_A, (x_A, D_A), and message m, perform the following steps.

(1) Choose a random number $a \in Z_q^*$ to compute $T_A = aP$.

(2) Compute $h = H_2(ID_A \| R_A \| m \| T_A \| X_A)$.

(3) Compute $s = a/(hx_A + D_A)$.

(4) Return $\sigma = (s, h)$ as the signature on the message m.

Verify. On input *params*, ID_A, T_A, R_A, (X_A, Q_A), m, and σ, compute $Q_A = H_1(ID_A, R_A)$ and $h = H_2(ID_A \| R_A \| m \| T_A \| X_A)$. Check whether $H_2(ID_A \| R_A \| m \| s(hX_A + R_A + Q_A P_{pub}) \| X_A) = h$ holds or not. If the equation holds, output 1, otherwise 0.

We can easily see that the following equation is correct:

$$\begin{aligned} & s\left(hX_A + R_A + Q_A P_{pub}\right) \\ &\quad = a\left(hx_A + r_A + xQ_A\right)^{-1}\left(hx_A + r_A P + Q_A xP\right) \\ &\quad = aP = T_A. \end{aligned} \tag{1}$$

Therefore, our CLS scheme is correct.

4. Security Analysis

In this section, we will show that the proposed scheme is secure in the random oracle model under the discrete logarithm assumption.

Theorem 1. *If Type I and Type II adversaries can forge a CLS scheme in probabilistic polynomial time t with nonnegligible probability ε, then the discrete logarithm problem can be solved with nonnegligible probability $\epsilon' \geq \varepsilon/(q_1^2 q_2)$, where q_i $(i = 1, 2)$ are denoted by the times of accessing H_i $(i = 1, 2)$ oracles, respectively.*

Proof. Firstly, we consider *Type I* attack.

Suppose that there exists a *Type I* adversary A_1 which has a nonnegligible probability ε in attacking our CLS scheme; we construct a challenger C that uses A_1 to solve the DLP. Challenger C receives a DL instance $y = vP$ for randomly chosen $v \in Z_q^*$ and $P \in G$ and wants to compute v. C runs A_1 as a subroutine and simulates its attack situations. C sets $P_{pub} = y$, where v is the master key, which is unknown to C, and returns system parameters to A_1. C maintains initially empty lists L_C, L_{H_1}, L_{H_2}, L_D, L_{SK}, L_{PK}, L_S, and L_V in order to simulate the oracle queries of A_1 as follows.

Create(ID_i). C maintains list L_C of tuple $(ID, R_i, P_i, D_i, x_i, h_i)$. C responds with $(ID, R_i, P_i, D_i, x_i, h_i)$ if ID_i is on L_C. Otherwise, it chooses three random values $a_i, b_i, x_i \in Z_q^*$, sets $R_i = a_i P - b_i P_{pub}$, $D_i = a_i$, $h_i = H_1(ID_i, R_i)$, $h_i = b_i$, and $PK_i = x_i P$, responds with $(ID, R_i, PK_i, D_i, x_i, h_i)$, and inserts (ID, R_i, h_i) into L_{H_1}. It is known that (ID, R_i, D_i, h_i) satisfies the equation $D_i P = R_i + h_i P_{pub}$ in the *Partial-Private-Key-Extract* algorithm.

H_1 Queries. Suppose that A_1 makes at most q_1 queries to the oracle H_1. C chooses a random number $j \in [1, q_1]$. When it makes a H_1 query on ID_i where $1 \leq i \leq q_1$, if $i = j$ (we let $ID_i = ID^*$ at this point), C randomly chooses $u \in Z_q^*$ and returns u and then adds $\langle ID_i, R_i \rangle$ to L_{H_1}. Otherwise, C picks a random number $h_i \in Z_q^*$, returns h_i to A_1, and adds $\langle ID_i, R_i, h_i \rangle$ to L_{H_1}.

H_2 Queries. Suppose that A_2 makes at most q_2 queries to the oracle H_2. If the list L_{H_2} contains $\langle m, R_i, X_i, T_i, \mathrm{ID}_i, h_i\rangle$, C returns h_i. Otherwise, C picks a random number $h_i \in Z_q^*$, returns h_i, and adds $\langle m, R_i, X_i, T_i, \mathrm{ID}_i, h_i\rangle$ to L_{H_2}.

Request-Public-Key(ID_i) Queries. Suppose that A_2 makes this query to the *Public-Key-Request* oracle. C looks up the list. C randomly chooses $\mathrm{PK}_i = bP$ and then adds $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i = \perp\rangle$ to $L_{H_{\mathrm{PK}}}$. Otherwise, C picks a random number $r_i \in Z_q^*$, computes $X_i = x_iP$, and adds $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i\rangle$ to L_{PK}.

Partial-Private-Key-Extract(ID_i) Queries. When A_1 makes this query, C does the following steps.

If $\mathrm{ID}_i = \mathrm{ID}^*$, C terminates the session. Otherwise, C looks up L_{H_1} for the tuple $\langle \mathrm{ID}_i, R_i, D_i\rangle$. If there exists such a tuple, C returns D_i to A_1. Otherwise, C makes *Replace-Public-Key queries* on itself and returns D_i as the response.

Extract-Secret-Key(ID_i) Queries. C picks a random $x_i \in Z_q^*$, computes $X_i = x_iP$, returns x_i to A_1, and adds $\langle \mathrm{ID}_i, X_i, x_i\rangle$ to L_{SK}.

Replace-Public-Key(ID_i, PK_i') Queries. If the list L_{PK} contains $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i\rangle$, C sets $\mathrm{PK}_i = \mathrm{PK}_i'$, $x_i = \perp$. Otherwise, C makes *Extract-Secret-Key*(ID_i) *query* on ID_i and then sets $\mathrm{PK}_i = \mathrm{PK}_i'$, $x_i = \perp$, and returns $\langle \mathrm{PK}_i, \perp\rangle$ to A_1.

Sign(m, ID_i) Queries. If the list L_{H_1} contains $\langle \mathrm{ID}_i, Q_i, r_i\rangle$ and the list L_{PK} contains $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i\rangle$, C does the following.

If $\mathrm{ID}_i = \mathrm{ID}^*$, then C picks three random numbers a, $x_i, r_i \in Z_q^*$, computes $T = aP$, $R = r_iP$, $X = x_iP$, $h = H_2(T\|X\|R\|\mathrm{ID}_i\|m)$, $D_i = r_i + vH_1(\mathrm{ID}_i, R) = r_i + vh_i$, and $s = a/(hx_i + r_i + vh_i)$, returns $\langle h, s\rangle$ to A_1, and adds $\langle m, \mathrm{PK}_i, \mathrm{ID}_i, r_i, r_iP\rangle$ and $\langle m, \mathrm{PK}_i, \mathrm{ID}_i, h_i\rangle$ to L_{H_1} and L_{H_2}, respectively. The public key PK_i may be replaced by A_1. The following equation holds because the signature is valid:

$$
\begin{aligned}
&s\left(hX_i + r_iP + h_1y\right)\\
&\quad = a\left(hx_i + r_i + vh_1\right)^{-1}\left(hx_iP + r_iP + h_1vP\right) = aP = T.
\end{aligned} \tag{2}
$$

C computes $(aP - shx_iP - sr_iP)/sh_1 = vP$.

Note that C can solve the DL problem because he knows (a, s, h, x_i, r_i, P). Thus, we have $\epsilon' \geq \varepsilon/(q_1^2q_2)$.

Then, we consider *Type II* attack.

Suppose that there exists a *Type II* adversary A_2 which has a nonnegligible probability ε in attacking our CLS scheme; we construct a challenger C that uses A_2 to solve the DLP. Challenger C receives a DL instance (vP, P) for randomly chosen $v \in Z_q^*$ and $P \in G$ and wants to compute v. C runs A_2 as a subroutine and simulates its attack situation. C sets $y = vP$, where v is the master key, and returns system parameters and v to A_2. C maintains initially empty lists L_C, L_{H_1}, L_{H_2}, L_D, L_{SK}, L_{PK}, L_S, and L_V in order to simulate the oracle queries of A_2 as follows.

Create(ID_i). C maintains list L_C of tuple $(\mathrm{ID}, R_i, \mathrm{PK}_i, D_i, x_i, h_i)$. C responds with $(\mathrm{ID}, R_i, \mathrm{PK}_i, D_i, x_i, h_i)$ if ID_i is on L_C. Otherwise, if $\mathrm{ID}_i = \mathrm{ID}^*$, C chooses three random values $r_i, b_i \in Z_q^*$ and sets $R_i = r_iP$, $h_i = H_1(\mathrm{ID}_i, R_i)$, $h_i = b_i$, $D_i = r_i + b_is \bmod q$, and $P_{\mathrm{pub}} = sP$, $x_i = \perp$. If $\mathrm{ID}_i \neq \mathrm{ID}^*$, C chooses three random values $x_i, b_i, r_i \in Z_q^*$ and sets $R_i = r_iP$, $h_i = H_1(\mathrm{ID}_i, R_i)$, $h_i = b_i$, $D_i = r_i + b_is \bmod q$, $P_{\mathrm{pub}} = sP$, and $\mathrm{PK}_i = x_iP$; C responds with $(\mathrm{ID}, R_i, \mathrm{PK}_i, D_i, x_i, h_i)$ and inserts (ID, R_i, h_i) into L_{H_1}.

H_1 Queries. Suppose that A_2 makes at most q_1 queries to the oracle H_1. C chooses a random $j \in [1, q_1]$. When it makes a H_1 query on ID_i where $1 \leq i \leq q_1$, if $i = j$ (we let $\mathrm{ID}_i = \mathrm{ID}^*$ at this point), C randomly chooses $r_i \in Z_q^*$ and returns r_i and then adds $\langle \mathrm{ID}_i, R_i, r_i\rangle$ to L_{H_1}. Otherwise, C picks a random number $h_i \in Z_q^*$, returns h_i to A_1, and adds $\langle \mathrm{ID}_i, R_i, h_i\rangle$ to L_{H_1}.

H_2 Queries. Suppose that A_2 makes at most q_2 queries to the oracle H_2. If the list L_{H_2} contains $\langle m, R_i, X_i, T_i, \mathrm{ID}_i, h_i\rangle$, C returns h_i. Otherwise, C picks a random number $h_i \in Z_q^*$, returns h_i, and adds $\langle m, R_i, X_i, T_i, \mathrm{ID}_i, h_i\rangle$ to L_{H_2}.

Request-Public-Key(ID_i) Queries. Suppose that A_2 makes at most q_{PK} queries to the *Public-Key-Request* oracle. C chooses a random $j \in [1, q_{\mathrm{PK}}]$. If $i = j$ (we let $\mathrm{ID}_i = \mathrm{ID}^*$ at this point), C randomly chooses $\mathrm{PK}_i = bP$ and then adds $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i = \perp\rangle$ to $L_{H_{\mathrm{PK}}}$. Otherwise, C picks a random number $r_i \in Z_q^*$, computes $X_i = x_iP$, and adds $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i\rangle$ to L_{PK}.

Partial-Private-Key-Extract(ID_i) Queries. When A_2 makes this query, C looks up the tuple $(\mathrm{ID}_i, R_i, D_i)$; if there is the tuple, then C returns D_i to A_2. Otherwise, C makes the *Request-Public-Key*(ID_i) queries and returns D_i to A_2.

Extract-Secret-Key(ID_i) Queries. C picks a random $x_i \in Z_q^*$, computes $X_i = x_iP$, returns x_i to A_1, and adds $\langle \mathrm{ID}_i, X_i, x_i\rangle$ to L_{SK}. Otherwise, C aborts the simulation.

Sign(m, ID_i) Queries. If the list L_{H_1} contains $\langle \mathrm{ID}_i, Q_i, r_i\rangle$ and the list L_{PK} contains $\langle \mathrm{ID}_i, \mathrm{PK}_i, x_i\rangle$, C does the following.

If $\mathrm{ID}_i = \mathrm{ID}^*$, then C picks three random $a, x_i, r_i \in Z_q^*$, computes $T = aP$, $X = x_iP$, $h = H_2(T\|X\|\mathrm{ID}_i\|m)$, $D_i = r_i + vH_1(\mathrm{ID}_i, R) = r_i + vh_i$, and $s = a/(hx_i + D_i) = a/(hx_i + r_i + vh_i)$, returns $\langle h, s\rangle$ to A_1, and adds $\langle m, \mathrm{PK}_i, \mathrm{ID}_i, r_i, r_iP\rangle$ and $\langle m, \mathrm{PK}_i, \mathrm{ID}_i, h_i\rangle$ to L_{H_1} and L_{H_2}, respectively. The public key PK_i may be replaced by A_1. The following equation holds because the signature is valid:

$$
\begin{aligned}
&s\left(hX_i + r_iP + h_1y\right)\\
&\quad = a\left(hx_i + r_i + vh_1\right)^{-1}\left(hx_iP + r_iP + h_1vP\right) = aP = T.
\end{aligned} \tag{3}
$$

C computes $(aP - shx_iP - sr_iP)/sh_1 = vP$.

Note that C can solve the DL problems because he knows (a, s, h, x_i, bP). Thus, we have $\epsilon' \geq \varepsilon/(q_1^2q_2)$. □

Table 1: Performance comparisons.

Schemes	Computational cost		Signature size (bytes)	Secure against Type I	Secure against Type II
	Sign	Verify			
Choi et al. [18]	3PM	2PM + 3P	64	Yes	No
He et al. [21]	1M	3M	41	Yes	No
Tsai et al. [23]	1M	4M	41	Yes	No
Gong and Li [24]	1M	4M	41	Yes	Yes
Yeh et al. [25]	1M	4M	41	Yes	Yes
Our scheme	1M	3M	41	Yes	Yes

5. Performance Comparison

In order to achieve 1024-bit RSA level security, we used the Tate pairing defined over the supersingular elliptic curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2. q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p is a 512-bit prime such as $p + 1 = 12qr$. The signature of Chen et al. [20] consists of a point of elliptic curve $E/F_p : y^2 = x^3 + x$; then, the signature size is 512/8 = 64 bytes. In order to achieve the same security level, we use the ECC group on Koblitz elliptic curve $y^2 = x^3 + x^2 + b$ defined on $F_{2^{163}}$ with $b = 163$ bit random prime. The signature size of our scheme is $\lceil 163 + 163/8 \rceil = 41$ bytes. The performance comparison among the proposed scheme and some related CLS schemes is given in Table 1. A pairing operation is denoted by P, scalar multiplication in the group G by M, a modular exponentiation in G_2 by E, and pairing-based scalar multiplication by PM. Sign and Ver denote the computational costs required for signing and verification processes of CLS scheme. According to Table 1, it is known that our scheme is more efficient than the other CLS schemes.

6. Conclusion

In this paper, we proposed a new CLS scheme without pairings and also showed that the proposed scheme is secure in the random oracle model under the DL assumption. The proposed scheme is more efficient than the previous CLS schemes in terms of computation and communication costs and is more suitable for the applications of low-bandwidth environments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the Major State Basic Research Development (973) Program of China (no. 2013CB834205), the National Natural Science Foundation of China (nos. 61070153 and 61103209), Natural Science Foundation of Zhejiang Province (no. LZ12F02005), and Education Department Foundation of Zhejiang Province (no. Y201222977).

References

[1] S. S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT 2003*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, Berlin, Germany, 2003.

[2] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *Cryptology and Network Security*, vol. 3810 of *Lecture Notes in Computer Science*, pp. 13–25, 2005.

[3] D. H. Yum and P. J. Lee, "Generic construction of certificateless signature," in *Information Security and Privacy: 9th Australasian Conference (ACISP '04)*, vol. 3108 of *Lecture Notes in Computer Science*, pp. 200–211, Springer, 2004.

[4] B. Hu, D. Wong, Z. Zhang, and X. Deng, "Key replacement attack against a generic construction of certificateless signature," in *Proceedings of the 11th Australasian Conference on Information Security and Privacy*, pp. 235–246, Melbourne, Australia, 2006.

[5] M. C. Gorantla and A. Saxena, "An efficient certificateless signature scheme," in *Computational Intelligence and Security (CIS '05)*, vol. 3802 of *Lecture Notes in Computer Science*, pp. 110–116, Springer, 3802, 2005.

[6] X. Cao, K. G. Paterson, and W. Kou, "An attack on a certificateless signature scheme," Cryptology ePrint Archive: Report 2006/367, 2006.

[7] W. Yap, S. Heng, and B. Goi, "An efficient certificateless signature scheme," in *Proceedings of Emerging Directions in Embedded and Ubiquitious Computing*, pp. 322–331, 2006.

[8] J. Park and B. Kang, "Security analysis of the certificateless signature scheme proposed at Sec Ubiq 2006," in *Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4809 of *Lecture Notes in Computer Science*, pp. 686–691, Springer, Berlin, Germany, 2007.

[9] L. Zhang, F. Zhang, and F. Zhang, "New efficient certificateless signature scheme," in *Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4809 of *Lecture Notes in Computer Science*, pp. 692–703, Springer, Berlin, Germany, 2007.

[10] Z. Zhang, D. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: security model and efficient construction," in *Applied Cryptography and Network Security*, vol. 3989 of *Lecture Notes in Computer Science*, pp. 293–308, Springer, Berlin, Germany, 2006.

[11] K. Y. Choi, J. H. Park, J. Y. Hwang, and D. H. Lee, "Efficient certificateless signature schemes," in *Applied Cryptography and Network Security*, vol. 4521 of *Lecture Notes in Computer Science*, pp. 443–458, Springer, New York, NY, USA, 2007.

[12] S. Duan, "Certificateless undeniable signature scheme," *Information Sciences*, vol. 178, no. 3, pp. 742–755, 2008.

[13] J. K. Liu, M. H. Au, and W. Susilo, "Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS '07)*, pp. 273–283, Singapore, March 2007.

[14] H. Xiong, Z. Qin, and F. Li, "An improved certificateless signature scheme secure in the standard model," *Fundamenta Informaticae*, vol. 88, no. 1-2, pp. 193–206, 2008.

[15] Y. Yuan, D. Li, L. Tian, and H. Zhu, "Certificateless signature scheme without random oracles," in *Advances in Information Security and Assurance: Proceedings of the 3rd International Conference and Workshops, ISA 2009, Seoul, Korea, June 25–27, 2009*, vol. 5576 of *Lecture Notes in Computer Science*, pp. 31–40, Springer, Berlin, Germany, 2009.

[16] X. Huang, W. Susilo, and D. Wong, "Certificateless signature revisited," in *Information Security and Privacy*, vol. 4586 of *Lecture Notes in Computer Science*, pp. 308–322, Springer, Berlin, Germany, 2007.

[17] H. Du and Q. Wen, "Efficient and provably-secure certificateless short signature scheme from bilinear pairings," *Computer Standards and Interfaces*, vol. 31, no. 2, pp. 390–394, 2009.

[18] K. Y. Choi, J. H. Park, and D. H. Lee, "A new provably secure certificateless short signature scheme," *Computers & Mathematics with Applications*, vol. 61, no. 7, pp. 1760–1768, 2011.

[19] K.-A. Shim, "Breaking the short certificateless signature scheme," *Information Sciences*, vol. 179, no. 3, pp. 303–306, 2009.

[20] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.

[21] D. He, J. Chen, and R. Zhang, "An efficient and provably-secure certificateless signature scheme without bilinear pairings," *International Journal of Communication Systems*, vol. 25, no. 11, pp. 1432–1442, 2012.

[22] M. Tian and L. Huang, "Cryptanalysis of a certificateless signature scheme without pairings," *International Journal of Communication Systems*, vol. 26, no. 11, pp. 1375–1381, 2013.

[23] J. L. Tsai, N. W. Lo, and T. C. Wu, "Weaknesses and improvements of an efficient certificateless signature scheme without using bilinear pairings," *International Journal of Communication Systems*, vol. 27, no. 7, pp. 1083–1090, 2014.

[24] P. Gong and P. Li, "Further improvement of a certificateless signature scheme without pairing," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2083–2091, 2014.

[25] K.-H. Yeh, K.-Y. Tsai, and C.-Y. Fan, "An efficient certificateless signature scheme without bilinear pairings," *Multimedia Tools and Applications*, 2014.