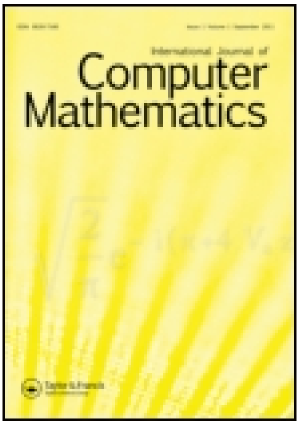


International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography

SK Hafizul Islam[a] & G.P. Biswas[a]

[a] Department of Computer Science and Engineering, Indian School of Mines, Dhanbad 826004, Jharkhand, India

Accepted author version posted online: 26 Feb 2013.Published online: 10 Apr 2013.

To cite this article: SK Hafizul Islam & G.P. Biswas (2013) Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography, International Journal of Computer Mathematics, 90:11, 2244-2258, DOI: 10.1080/00207160.2013.776674

To link to this article: http://dx.doi.org/10.1080/00207160.2013.776674

International Journal of Computer Mathematics, 2013
Vol. 90, No. 11, 2244–2258, http://dx.doi.org/10.1080/00207160.2013.776674

Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography

SK Hafizul Islam* and G.P. Biswas

Department of Computer Science and Engineering, Indian School of Mines, Dhanbad 826004, Jharkhand, India

(*Received 2 April 2012; revised version received 13 June 2012; second revision received 2 September 2012; third revision received 3 December 2012; accepted 11 February 2013*)

A provably secure certificateless digital signature scheme using elliptic curve cryptography is presented in this paper. Since the certificateless public key cryptosystem removes the complex certificate management procedure and the private key escrow problem of traditional public key cryptography (PKC) and identity-based cryptosystem (IBC), respectively, and as a result, the proposed scheme is more efficient than IBC- and PKC-based signatures. Besides, the bilinear pairing and map-to-point hash function are time-consuming operations, and thus the signatures without these two operations are more attractive in real applications and the present work has been carried out in this direction. Based on the elliptic curve discrete logarithm assumption, it is shown that the proposed scheme is unforgeable under the adaptive chosen message and identity attacks in the random oracle model against variety of adversaries. Finally, our signature scheme is compared with a number of competitive schemes and the satisfactory performance has been achieved.

Keywords: certificateless cryptosystem; elliptic curve cryptography; digital signature; random oracle model; provable security

2010 AMS Subject Classifications: 94A60; 14H52; 68P25; 97Fxx; 97P20

1. Introduction

The digital signature is the most fundamental tools used in public key cryptography (PKC)/public key infrastructure (PKI) in order to accomplish the message integrity, authenticity and non-repudiation, when the messages are exchanged over any public channel. The notion of PKC was first proposed by Diffie and Hellman [11] in 1976, based on which several digital signature schemes have been proposed [12,25,27,30–32]. Out of all, the RSA signature is one of them and its security mainly depends on the two prime numbers used and if these numbers are not sufficiently large like 1024-bit or more, the RSA signature is easily breakable. On the other hand, the ElGamal signature is unforgeable based on the decisional Diffie–Hellman assumption in large multiplicative group. The digital signature schemes [12,25,27,30–32] based on PKI need complex public key certificate management process in order to authenticate the public key, which decreases the applicability in real environments. Note that the generation, management, delivery, revocation, etc. of certificate need to bear high computing cost and the huge storage space. To defeat these troubles, Shamir [33] introduces the idea of identity-based cryptosystem (IBC) in

*Corresponding author. Email: hafi786@gmail.com, hafizul.ism@gmail.com

1984, which eliminates the use of public key certificate as needed in PKI-based cryptosystems. In this scheme, a user's public key is computed from the publicly known identity of the user such as email identity, passport number, social security number, etc. and a trusted third party, called private key generator (PKG) is responsible to generate the corresponding private key of the user by binding user's identity and PKG's private key. However, the IBC schemes have an inherent problem known as private key escrow problem since PKG have the knowledge about the private keys of all users in the system and thus, the user secrecy can be compromised if PKG is not trusted.

In order to manage these problems, the certificateless public key cryptography (CL-PKC), which captures the advantages of both the cryptosystems PKI and IBC, was proposed by Al-Riyami and Paterson [1]. In this setting, like PKI the user chooses a secret key completely unknown to PKG and similar to IBC, and the PKG computes the identity-based long-term private key of the user, and these two secrets are combined together to form the full private key of the user. Thus, the needs of public key certificate in PKI and the private key escrow problem exists in IBC are abolished in the CL-PKC cryptosystems. In CL-PKC, the identity-based public key is easily computable from the identity of the user and the PKI-based public key is easily accessible from the directory of the corresponding user. In the following, we briefly address some of the efficient certificateless digital signature (CL-DS) schemes and their shortcomings.

1.1 *Related works*

Al-Riyami and Paterson first establish the notion of CL-DS scheme in 2003, later on, Huang *et al.* [21] demonstrated that the scheme is susceptible against the public key replacement attack, and they then proposed a new CL-DS scheme as a remedy of it. After the pioneer work of Al-Riyami and Paterson, several CL-DS schemes [9,14,20,26,35–37,40,41] and different security models for signature have been proposed in recent years. In 2006, Zhang *et al.* [40] construct a more efficient security model than the model presented in [21]. Huang *et al.* [20] independently define three types of adversaries, called normal adversary, strong adversary and super adversary according to their attack powers. They also revisited the security models of CL-DS schemes and proposed two schemes that were proven to be secured in the random oracle model against all potential adversaries. In 2004, Yum and Lee [37] also proposed a new CL-DS model and claimed strong security against all adversaries; however, Hu *et al.* [19] show that the security assumptions adopted in [37] are insufficient to protect the key replacement attack. Subsequently, Hu *et al.* proposed a modernized scheme and its security was established in a new and simplified security model. In 2005, Gorantla and Saxena [14] designed a pairing-based CL-DS scheme and claimed that the scheme can provide the desirable security and computation efficiency; later on, Cao *et al.* [8] illustrate the vulnerability of the scheme against the public key replacement attack.

In 2004, Yap *et al.* [36] proposed a pairing-based CL-DS scheme based on the intractability of the computational Diffie–Hellman problem (CDHP) in the random oracle model. The scheme is computationally efficient since the signing phase is free from bilinear pairing operation and the verification phase requires only two bilinear pairing computation. However, Zhang and Feng [38] analysed that the public key replacement attack could be mounted on Yap *et al.*'s scheme. To cope with the problem of Yap *et al.*'s scheme, Hu *et al.* proposed an improved CL-DS scheme. In 2007, Choi *et al.* [9] construct an efficient CL-DS scheme using elliptic curve bilinear pairing and then analysed its security in the random oracle model based on the CDHP and mICDHP (modified inverse CDHP) assumptions. Recently, two CL-DS schemes were proposed in [35,41] and the authors claimed that the schemes were intended to provide the strong security in the random oracle model against adaptive chosen message and identity adversaries. Unfortunately, the papers in [15,39] show that the scheme presented in [35] is insecure against universal forgeries under known message attack.

1.2 *Motivations and contributions*

In the previous section, some discussions have been made on a number of earlier CL-DS schemes and their security and computing cost. It is found that most of the schemes are insecure against different security attacks and they used costly operations such as elliptic curve bilinear pairing and map-to-point (MTP) hash function [6,24,28]. Despite the usefulness of bilinear pairing and MTP function in cryptography, these operations increase computation costs in the schemes employed, since the time needed to execute one bilinear pairing operation is approximately two to three times more than an elliptic curve scalar point multiplication (ECPM) requires [7,18,34] and the computation cost of MTP hash function is more than an ECPM [4,5]. In addition, the implementation of these schemes is also difficult as a bilinear pairing operation needs a super-singular elliptic curve group with large element size and the MTP hash function is usually implemented as a probabilistic algorithm [7]. Therefore, the security and computation cost deficiencies of the previous schemes inspired us to construct an efficient CL-DS scheme having low computation cost and strong security in the random oracle model.

In this paper, we proposed an efficient CL-DS scheme using elliptic curve cryptography (ECC), whose security is analysed in the random oracle model [3] against the various adaptive chosen message and identity adversaries with different attack powers. Compared with competitive schemes, the proposed scheme provides the desired efficiency in computation while maintaining the strong security. The proposed CL-DS scheme has the following three contributions. Firstly, the scheme does not require bilinear pairing operation and MTP hash function and thus, it requires less effort for realization. Secondly, the proposed scheme is provably secure in the random oracle model based on elliptic curve discrete logarithm problem (ECDLP) assumption, so it provides strong security against various adversaries. Thirdly, the computation cost of our scheme is lower than other existing scheme and thus, it may be implemented in low power, low computation cost and low storage space devices such as mobile phones, smartcards, PDAs (personal digital assistants), etc.

1.3 *Organization of the paper*

The remainder of the paper is organized in the following way. Section 2 describes some preliminaries of the elliptic curve and some of its computational problems. In Section 3, the formal definition and the attack model of CL-DS scheme are given. The proposed CL-DS scheme is presented in Section 4 and Section 5 analyses the correctness and security of the proposed scheme in the random oracle model. The comparison of our scheme with others is discussed in Section 6 and finally, Section 7 concludes the paper.

2. Preliminaries

This section briefly introduces the basic idea of the elliptic curve group, the group properties and some of the mathematical hard problems that are proven to be intractable by a polynomial time-bounded algorithm.

2.1 *Elliptic curve cryptography*

The ECC was proposed by Miller [28] and Koblitz [24], and its security is based on the difficulty of solving the ECDLP. Any cryptosystem based on ECC provides high security with small key size, for example, a 160-bit ECC is considered to be as secured as 1024-bit RSA key [16]. Let F_{q}

be the field of integers of modulo a large prime number q. A non-singular elliptic curve $E_{\mathrm{q}}(a, b)$ over F_{q} is defined by the following equation:

$$y^2 \bmod q = (x^3 + ax + b) \bmod q, \tag{1}$$

where $a, b, x, y \in F_q$ and $\Delta = (4a^3 + 27b^2) \bmod q \neq 0$. A point $P(x, y)$ is an elliptic curve point if it satisfies Equation (1), and the point $Q(x, -y)$ is called the negative of P, i.e. $Q = -P$. Let $P(x_1, y_1)$ and $Q(x_2, y_2)(P \neq Q)$ be two points in Equation (1), the line l (tangent line to Equation (1) if $P = Q$) joining the points P and Q intersects the curve (1) at $-R(x_3, -y_3)$ and the reflection of $-R$ with respect to x-axis is the point $R(x_3, y_3)$, i.e. $P + Q = R$. The points $E_q(a, b)$ together with a point O (called point at infinity) form an additive cyclic group G_{q}, that is, $G_{\mathrm{q}} = \{(x, y)$: $a, b, x, y \in F_{\mathrm{q}}$ and $(x, y) \in E_q(a, b)\} \cup \{O\}$ of prime order q. The scalar point multiplication on the group G_{q} can be computed as follows: $kP = P + P + \cdots + P$ (k times). A point P has order n if n is the smallest positive integer such that $nP = O$.

2.2 *Computational problems*

In this section, some computational problems are introduced, which are computationally intractable by a polynomial time-bounded algorithm.

Definition 1 *A function $\varepsilon(k)$ is said to be negligible if, for every $c > 0$, there exists k_0 such that $\varepsilon(k) \leq 1/k^c$ for every $k \geq k_0$.*

Definition 2 *Elliptic curve discrete logarithm problem (ECDLP): Given a random instance $(P, Q) \in G_q$, find a number $a \in_R Z_q^*$ such that $Q = aP$. The probability that a polynomial time-bounded algorithm B can solve the ECDLP problem is defined as $Adv_{B,G_q}^{ECDLP}(k) = \Pr[B(P, Q) = a : a \in_R Z_q^*, Q = aP]$.*

Definition 3 *Elliptic curve discrete logarithm assumption: For every probabilistic polynomial time-bounded algorithm $B, Adv_{B,G_q}^{ECDLP}(k)$ is negligible, i.e. $Adv_{B,G_q}^{ECDLP}(k) \leq \varepsilon$ for some negligible function ε.*

3. Formal model of pairing-free certificateless digital signature scheme

3.1 *Definition of pairing-free certificateless digital signature scheme*

This section describes the formal definition [9,20,21,40,41] of a pairing-free CL-DS scheme. In a CL-DS scheme, three entities are involved, namely a trusted third party PKG who is responsible to generate all the system parameters and partial private keys of all users related with the system, a signer who generates the signature on a given message and a verifier who verifies the message–signature pair. An efficient CL-DS scheme consists of the following algorithms and the description of each is given below:

- *Setup*: The PKG runs this algorithm, which takes the security parameter $k \in Z^+$ as inputs and outputs the system's parameter Ω and a master secret key, *msk*, of PKG. Note that the system's parameter Ω is known to all the users whereas the *msk* is kept secret by PKG.
- *Set-Secret-Value*: This algorithm is run by every user in the system separately to generate a secret value for oneself. It takes the system's parameter Ω and an identity ID_i of the user i as inputs and then returns the secret key x_i as output to ID_i.

- *Partial-Private-Key-Extract*: The PKG runs this algorithm to generate the partial private key of the users. The inputs of this algorithm are the system's parameter Ω, master key *msk* and an identity ID_i of the user i, and then it outputs the partial private key D_i of ID_i.
- *Set-Private-Key*: In order to generate the full private key, every user in the system executes this algorithm that needs the system's parameter Ω, partial private key D_i and the secret value x_i of ID_i as inputs and it outputs the full private key sk_i for ID_i.
- *Set-Public-Key*: This algorithm is also run by every user independently in order to compute one's full public key intended for him. It takes the system's parameter Ω and the secret value x_i of ID_i as inputs and then returns the full public key pk_i to ID_i as output.
- *CL-DS-Sign*: The signer ID_i runs this algorithm to obtain a signature on a given message. This algorithm takes system's parameter Ω, full private key sk_i of ID_i and a message $m_i \in \{0, 1\}^*$ as inputs and then outputs a signature δ_i for the same message m_i.
- *CL-DS-Verify*: This algorithm is executed by the verifier in order to verify the message–signature pair generated by the signer. This algorithm takes system's parameter Ω, full public pk_i of ID_i and a message–signature pair (m_i, δ_i) as inputs. It outputs true if the signature δ_i is valid, otherwise outputs false.

3.2 *Attack model of certificateless digital signature scheme*

In this subsection, we describe the attack model of the CL-DS scheme. The stronger security notion [9,20,41] of a CL-DS scheme is the existential unforgeability against the adaptive chosen message and identity attacks (EUF-CMA), which means that an adversary chooses any message randomly from the message space and then asks the signer to sign the message. In any CL-PKC, two types of adversaries [1] with different attack powers can be found, namely the Type I adversary A_{I} and the Type II adversary A_{II}. The adversary A_{I} acts as a malicious user while the adversary A_{II} acts as a malicious PKG.

- *Type I adversary* A_{I}: The adversary A_{I} cannot access the PKG's master secret key, but he can replace the public key of any user with a value chosen by him.
- *Type II adversary* A_{II}: The adversary A_{II} can access the PKG's master secret key, but he cannot change the public key of any user.

The adaptive chosen message and identity attacks caused by the adversary $A \in \{A_{\mathrm{I}}, A_{\mathrm{II}}\}$ are considered as powerful attacks in CL-PKC cryptosystems. The unforgeability of a CL-DS scheme against EUF-CMA is defined by the following challenge–response games: *Game 1* and *Game 2* played between a challenger C and the adversary $A \in \{A_{\mathrm{I}}, A_{\mathrm{II}}\}$.

Game 1 The adversary A_{I} and the challenger C interactively play this game in order to breach the security of CL-DS scheme under EUF-CMA.

- *Setup*: The challenger C takes a security parameter $k \in \mathrm{Z}^+$ and runs this algorithm to generate PKG's master key *msk* and system's parameter Ω. Then C sends Ω to A_{I} while keeping *msk* secret.
- *Hash queries*: The adversary A_{I} can request the hash value for any input.
- *Set-Secret-Value queries*: The adversary A_{I} can ask for the secret value of the user ID_i, C then computes the corresponding secret value x_i and sends it to A_{I}.
- *Partial-Private-Key-Extract queries*: The adversary A_{I} can ask for the partial private key of ID_i and in response, C computes the corresponding D_i and sends it to A_{I}.
- *Set-Public-Key queries*: The adversary A_{I} can ask for the full public key of ID_i, C then computes the corresponding pk_i and sends it to A_{I}.

- *Public-Key-Replacement queries*: The adversary A_{I} can choose a new public key P_i' for ID_i and asks to replace the old public key P_i by P_i'. Then A_{I} sets P_i' as new public key of ID_i and C records it.
- *CL-DS-Sign queries*: The adversary A_{I} can ask for a signature on the chosen message m_i with signer's identity ID_i. Then C executes *CL-DS-Sign* algorithm and outputs a message–signature pair (m_i, δ_i) and returns it to A_{I}.
- *CL-DS-Verify queries*: The adversary A_{I} can ask for the verification of a message–signature pair (m_i, δ_i) with signer's identity ID_i, C then runs *CL-DS-Verify* algorithm and outputs true if (m_i, δ_i) is valid. Otherwise, outputs false.
- *Forgery*: At the end of *Game 1*, A_{I} outputs a forged message–signature pair (m_i^*, δ_i^*) with signer's identity ID_i. The adversary A_{I} can win the *Game 1* if the following holds:
  (a) The signature δ_i^* is valid signature of the message m_i^* with respect to ID_i.
  (b) The message m_i^* has never been submitted to the *CL-DS-Sign* oracle with signer's identity ID_i.
  (c) The identity ID_i has never been submitted to the oracle *Partial-Private-Key-Extract* and *Set-Secret-Value-Extract*.

DEFINITION 4 *The probability that a polynomial time-bounded adversary A_{I} can break the security of the CL-DS scheme under the adaptive chosen message and identity attacks is defined as $Adv_{CL-DS,A_I}^{ECF-CMA}(k)$. The CL-DS signature scheme is secure against polynomial time-bounded adversary A_{I}, if the advantage to win Game 1 defined above is negligible, i.e. $Adv_{CL-DS,A_I}^{ECF-CMA}(k) \leq \varepsilon$ for some negligible function ε.*

Game 2 This game is executed between the probabilistic polynomial time-bounded EUF-CMA adversary A_{II} and the challenger C. This game can be defined as follows:

- *Setup*: The challenger C runs this algorithm, which takes the security parameter $k \in \mathrm{Z}^+$ as inputs and generates PKG's secret key *msk* and system's parameter Ω, C then sends Ω to A_{II} while keeping the *msk* secret.
- *Hash queries*: The adversary A_{II} can request the hash value for any input.
- *Set-Secret-Value queries*: The adversary A_{II} can ask for the secret value of the user ID_i, C then computes the corresponding secret value x_i and sends it to A_{II}.
- *Partial-Private-Key-Extract queries*: The adversary A_{II} can ask for the partial private key of ID_i, C then computes the corresponding D_i and sends it to A_{II}.
- *Set-Public-Key queries*: The adversary A_{II} can ask for the full public key ID_i, C then computes the corresponding pk_i and sends it to A_{II}.
- *CL-DS-Sign queries*: The adversary A_{II} can ask for a signature of the chosen message m_i with signer's identity ID_i, and C executes *CL-DS-Sign* algorithm to output a message–signature pair (m_i, δ_i) and then returns it to A_{II}.
- *CL-DS-Verify queries*: The adversary A_{II} can ask for the verification of a message–signature pair (m_i, δ_i) with signer's identity ID_i, C then runs *CL-DS-Verify* algorithm and outputs true if (m_i, δ_i) is valid. Otherwise, outputs false.
- *Forgery*: At the end of *Game 2*, A_{II} outputs a forged (m_i^*, δ_i^*) with signer's identity ID_i. The adversary A_{II} can win *Game 2* if the following holds:
  (a) The signature δ_i^* is valid signature of the message m_i^* with signer's identity ID_i.
  (b) The message m_i^* has never been submitted to the *CL-DS-Sign* oracle with signer's identity ID_i.
  (c) The identity ID_i has never been submitted to the oracles *Set-Secret-Value* and *Public-Key-Replacement.*

Definition 5 *The advantage $Adv_{CL-DS,A_{II}}^{ECF-CMA}(k)$ is defined as the probability that a polynomial time-bounded adaptive chosen message and identity adversary A_{II} wins the Game 2. The CL-DS scheme is secure against A_{II} adversary if $Adv_{CL-DS,A_{II}}^{ECF-CMA}(k)$ is negligible, i.e. $Adv_{CL-DS,A_{II}}^{ECF-CMA}(k) \leq \varepsilon$ for some negligible function ε.*

Definition 6 *The CL-DS scheme is existentially unforgeable against adaptive chosen message and identity attacks if it is secure against the adversaries A_I and A_{II}, i.e. if both $Adv_{CL-DS,A_I}^{ECF-CMA}(k) \leq \varepsilon$ and $Adv_{CL-DS,A_{II}}^{ECF-CMA}(k) \leq \varepsilon$ hold for some negligible function ε.*

4. Proposed pairing-free certificateless digital signature scheme

We motivated from the pairing-free protocols proposed in [2,7,17,22,23] and Schnorr's signature scheme [32], and proposed a pairing-free CL-DS scheme using ECC in this section. As stated, the scheme provides strong security in the random oracle model [3] based on ECDLP assumption and has lower computation costs since it is free from bilinear pairing and MTP function. Our CL-DS scheme executes the following algorithms to generate a signature on a given message.

4.1 *Setup*

This algorithm takes a security parameter $k \in Z^+$ as input, and returns list of system's parameter Ω and PKG's master private key *msk*. Given k, PKG performs in the following way:

(a) Choose a *k-bit* prime number q and determine the tuple $\{F_q, E/F_q, G_q, P\}$, where the point P is the generator of G_q.
(b) Choose the master key $x \in_R Z_q^*$ and compute the system public key $P_{\text{pub}} = xP$.
(c) Choose three cryptographic secure hash functions $H_0 : \{0,1\}^* \times G_q^2 \rightarrow Z_q^*$, $H_1 : \{0,1\}^* \times \{0,1\}^* \times G_q^2 \rightarrow Z_q^*$ and $H_2 : \{0,1\}^* \times \{0,1\}^* \times G_q^2 \rightarrow Z_q^*$.
(d) Publish $\Omega = \{F_q, E/F_q, G_q, P, P_{\text{pub}}, H_0, H_1, H_2\}$ as the system's parameter and keep the master key x secret.

4.2 *Set-Secret-Value*

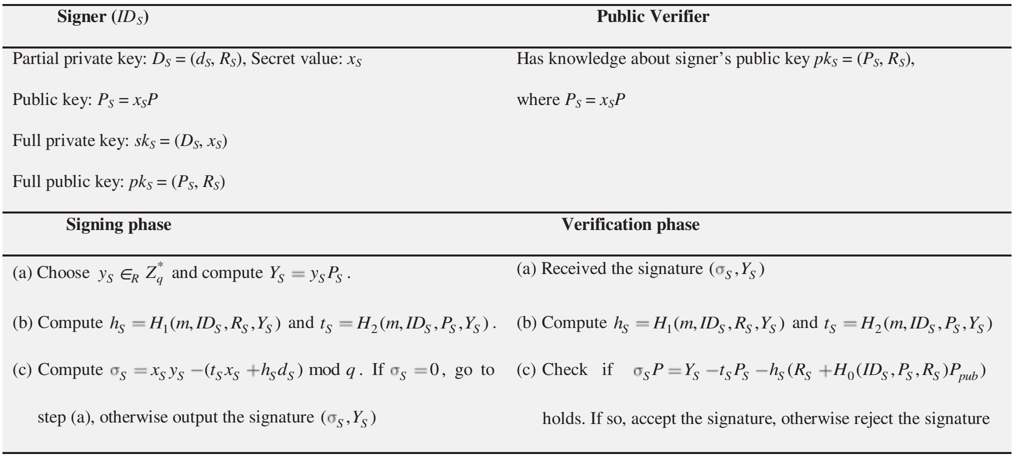
The user $\text{ID}_i \in \{0,1\}^*$ picks a number $x_i \in_R Z_q^*$ and sets x_i as his secret value and generates the corresponding public key as $P_i = x_iP$.

4.3 *Partial-Private-Key-Extract*

This algorithm takes PKG's master secret key, identity of a user and the system's parameter as input and returns the identity-based private key of the user. In order to obtain the partial private key, the user ID_i sends (ID_i, P_i) to the PKG and then PKG does as follows:

(a) PKG chooses a number $r_i \in_R Z_q^*$ and computes $R_i = r_iP$.
(b) PKG computes $d_i = r_i + xH_0(\text{ID}_i, R_i, P_i) \mod q$.

The partial private key of the user ID_i is the tuple $D_i = (d_i, R_i)$, from which the user can validate his private key by checking whether the equation $d_iP = R_i + H_0(ID_i, R_i, P_i)P_{\text{pub}}$ holds. The private key D_i is valid if the above equation holds and vice-versa.



Signer (ID_S)	**Public Verifier**
Partial private key: $D_S = (d_S, R_S)$, Secret value: x_S	Has knowledge about signer's public key $pk_S = (P_S, R_S)$,
Public key: $P_S = x_SP$	where $P_S = x_SP$
Full private key: $sk_S = (D_S, x_S)$	
Full public key: $pk_S = (P_S, R_S)$	
Signing phase	**Verification phase**
(a) Choose $y_S \in_R Z_q^*$ and compute $Y_S = y_SP_S$.	(a) Received the signature (σ_S, Y_S)
(b) Compute $h_S = H_1(m, ID_S, R_S, Y_S)$ and $t_S = H_2(m, ID_S, P_S, Y_S)$.	(b) Compute $h_S = H_1(m, ID_S, R_S, Y_S)$ and $t_S = H_2(m, ID_S, P_S, Y_S)$
(c) Compute $\sigma_S = x_Sy_S - (t_Sx_S + h_Sd_S) \bmod q$. If $\sigma_S = 0$, go to step (a), otherwise output the signature (σ_S, Y_S)	(c) Check if $\sigma_SP = Y_S - t_SP_S - h_S(R_S + H_0(ID_S, P_S, R_S)P_{pub})$ holds. If so, accept the signature, otherwise reject the signature

Figure 1. The proposed pairing-free CL-DS scheme.

4.4 *Set-Private-Key*

The user ID_i takes the pair $\mathrm{sk}_i = (D_i, x_i)$ as his full private key, where $D_i = (d_i, R_i)$.

4.5 *Set-Public-Key*

The user ID_i takes $\mathrm{pk}_i = (P_i, R_i)$ as his full public key.

4.6 *CL-DS-Sign*

To sign a message $m \in \{0, 1\}^*$, the signer ID_S with full private key $\mathrm{sk}_\mathrm{S} = (D_\mathrm{S}, x_\mathrm{S})$ computes the signature as follows:

(a) Choose a number $y_\mathrm{S} \in_R Z_q^*$ and compute $Y_\mathrm{S} = y_\mathrm{S}P_\mathrm{S}$.
(b) Compute $h_\mathrm{S} = H_1(m, \mathrm{ID}_\mathrm{S}, R_\mathrm{S}, Y_\mathrm{S})$ and $t_\mathrm{S} = H_2(m, \mathrm{ID}_\mathrm{S}, P_\mathrm{S}, Y_\mathrm{S})$.
(c) Compute $\sigma_\mathrm{S} = x_\mathrm{S}y_\mathrm{S} - (t_\mathrm{S}x_\mathrm{S} + h_\mathrm{S}d_\mathrm{S}) \mod q$. If $\sigma_\mathrm{S} = 0$, go to step (a), otherwise output the signature $(\sigma_\mathrm{S}, Y_\mathrm{S})$ for the message m and send it to the verifier for verification.

4.7 *CL-DS-Verify*

To verify a signature $(\sigma_\mathrm{S}, Y_\mathrm{S})$ for a message m, the verifier uses the full public key $\mathrm{pk}_\mathrm{S} = (P_\mathrm{S}, R_\mathrm{S})$ of the signer ID_S and then performs the following steps:

(a) Compute $h_\mathrm{S} = H_1(m, \mathrm{ID}_\mathrm{S}, R_\mathrm{S}, Y_\mathrm{S})$ and $t_\mathrm{S} = H_2(m, \mathrm{ID}_\mathrm{S}, P_\mathrm{S}, Y_\mathrm{S})$.
(b) Check whether the equation $\sigma_\mathrm{S}P = Y_\mathrm{S} - t_\mathrm{S}P_\mathrm{S} - h_\mathrm{S}(R_\mathrm{S} + H_0(\mathrm{ID}_\mathrm{S}, P_\mathrm{S}, R_\mathrm{S})P_\mathrm{pub})$ holds. If so, the verifier accepts the signature $(\sigma_\mathrm{S}, Y_\mathrm{S})$, otherwise rejects it.

The proposed scheme is further illustrated in Figure 1.

5. Analysis of the proposed certificateless digital signature scheme

In this section, we analyse the proposed pairing-free CL-DS scheme rigorously from the aspects of security. The proposed scheme is secure against various adversaries with different capabilities under the adaptive chosen message and identity attacks based on the ECDLP assumption in the elliptic curve group. In the following, the correctness of our scheme is described first and then some theorems related to the security of the proposed scheme are presented.

5.1 *Correctness*

It can be proved that the proposed pairing-free CL-DS scheme is correct by the following computations.

Since $h_S = H_1(m, \mathrm{ID}_S, R_S, Y_S)$ and $t_S = H_2(m, \mathrm{ID}_S, P_S, Y_S)$, we obtain

$$\begin{aligned}\sigma_S P &= [x_S y_S - (t_S x_S + h_S d_S)]P \\ &= x_S y_S P - (t_S x_S + h_S d_S)P \\ &= y_S P_S - (t_S x_S P + h_S d_S P) \quad [\because P_S = x_S P] \\ &= Y_S - t_S P_S - h_S(r_S + x H_0(\mathrm{ID}_S, P_S, R_S)P) \quad [\because Y_S = y_S P_S] \\ &= Y_S - t_S P_S - h_S(r_S P + H_0(\mathrm{ID}_S, P_S, R_S)xP) \quad [\because R_S = r_S P] \\ &= Y_S - t_S P_S - h_S(R_S + H_0(\mathrm{ID}_S, P_S, R_S)P_{\mathrm{pub}}) \quad [\because P_{\mathrm{pub}} = xP].\end{aligned}$$

This justifies the correctness of the proposed CL-DS scheme.

5.2 *Security analysis*

The rigorous security analysis of the proposed CL-DS scheme in the random oracle model is presented in this subsection. According to [1], two types of adversary A_{I} and A_{II} are present in any certificateless cryptosystem, the former adversary is nothing but a dishonest user who has the ability to replace the public key of any user with the value of his choice and the later adversary is a malicious PKG who can access the master key, but cannot replace the public key of any user. The existential unforgeability of the proposed CL-DS scheme is based on the intractability of ECDLP problem in the elliptic curve group against the adversaries under the adaptive chosen message and identity attacks. For proving the strong security of the proposed scheme, the following theorems are presented.

THEOREM 1 *The proposed pairing-free CL-DS scheme is existential unforgeable under the adaptive chosen message and identity attacks against the adversary A_I in the random oracle model provided the ECDLP problem is intractable by any polynomial time-bounded algorithm in the elliptic curve group.*

Proof Assume that the proposed CL-DS scheme can be forged under the adaptive chosen message and identity attacks by a polynomial time-bounded adversary A_{I}, then it is possible to construct an algorithm C, which helps A_{I} to solve an instance of ECDLP, i.e. A_{I} outputs a from the input tuple $(P, Q = aP)$, where $a \in_R Z_q^*$. To solve the ECDLP, C sets the private/public key pair of PKG as $(x = a, P_{\mathrm{pub}} = aP)$, where P is the generator of the group G_q and the hash functions $H_i (i = 0, 1, 2)$ are treated as random oracle. The algorithm C sends the system's parameter $\Omega = \{F_q, E/F_q, G_q, P, P_{\mathrm{pub}}, H_0, H_1, H_2\}$ to the adversary A_{I}. To response quickly and to avoid the inconsistency, C maintains the following lists:

(a) H_0 *list* $L_{H_0}^{\mathrm{list}}$: This is an initial-empty list, which includes the tuple of the form $(\mathrm{ID}_i, R_i, P_i, l_i)$.

(b) H_1 *list*$L_{H_1}^{\mathrm{list}}$: This is an initial-empty list, which includes the tuple of the form $(\mathrm{ID}_i, m_i, R_i, Y_i, h_i)$.

(c) *H_2 list*$L^{\text{list}}_{H_2}$: This is an initial-empty list that includes the tuple of the form $(\mathrm{ID}_i, m_i, P_i, Y_i, t_i)$.

(d) *List* L^{list}_{C}: This is an initial-empty list that contains the tuple of the form $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$.

- *Create(ID_i)*: On receiving a *Create(ID_i)* query, C picks the number a_i, b_i, $x_i \in_R Z_q^*$, then sets $H_0(\mathrm{ID}_i, R_i, P_i) \leftarrow -b_i$, $R_i \leftarrow b_i P_{\text{pub}} + a_i P$ and computes the public key as $P_i = x_i P$. Note that $D_i = (d_i, R_i)$ with $d_i = a_i$ satisfies the equation $d_i P = R_i + H_0(\mathrm{ID}_i, R_i, P_i) P_{\text{pub}}$. Now, C responds in the following way:
  (a) If $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$, C then outputs the partial private key, full private key and full public key of the user ID_i as $D_i = (\perp, R_i)$, $\mathrm{sk}_i = (\perp, x_i)$ and $\mathrm{pk}_i = (P_i, R_i)$, respectively.
  (b) If $\mathrm{ID}_i \neq \mathrm{ID}_{\mathrm{S}}$, C outputs the partial private key, full private key and full public key of ID_i as $D_i = (d_i, R_i)$, $\mathrm{sk}_i = (D_i, x_i)$ and $\mathrm{pk}_i = (P_i, R_i)$, respectively.
  Finally, C inserts the tuples $(\mathrm{ID}_i, R_i, P_i, b_i)$ and $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$ to the list $L^{\text{list}}_{H_0}$ and L^{list}_{C}, respectively.
- *Hash queries to H_0*: Suppose A_{I} submits a H_0 query with ID_i. Then C searches the list $L^{\text{list}}_{H_0}$ and responds with the previous value l_i if there is a tuple $(\mathrm{ID}_i, R_i, P_i, l_i)$, otherwise, choose a number $l_i \in_R Z_q^*$ and outputs it as the answer and then, inserts the tuple $(\mathrm{ID}_i, R_i, P_i, l_i)$ to $L^{\text{list}}_{H_0}$.
- *Partial-Private-Key-Extract queries*: To obtain the partial private key, adversary A_{I} executes this query on ID_i, C then responds as follows:
  (a) If $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$, C stops the execution.
  (b) Else C searches the list L^{list}_{C} for the tuple $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$ and returns D_i as the answer.
- *Set-Secret-Value queries*: If A_{I} asks a *Set-Secret-Value* query for ID_i, C then answers as follows:
  (a) If $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$, then C aborts the simulation.
  (b) If $\mathrm{ID}_i \neq \mathrm{ID}_{\mathrm{S}}$, C looks for the tuple $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$ in L^{list}_{C} and outputs x_i if such tuple is found, otherwise C asks a *Create(ID_i)* query to obtain the tuple $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$ and then replies with x_i.
- *Set-Public-Key queries*: For this query, C searches the list L^{list}_{C} and answers in the following way:
  (a) If a tuple $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$ is found in L^{list}_{C}, C then outputs $\mathrm{pk}_i = (R_i, P_i)$.
  (b) Else C executes a *Create(ID_i)* query to obtain the tuple $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$, and then outputs $\mathrm{pk}_i = (R_i, P_i)$.
- *Public-Key-Replacement queries*: If A_{I} wants to replace the public key pk_i of ID_i with a value pk'_i of his choice, then C looks into the list L^{list}_{C} for $(\mathrm{ID}_i, D_i, \mathrm{sk}_i, \mathrm{pk}_i)$ and then updates pk_i with pk'_i. Now, C sets $d_i = \perp$ and $\mathrm{sk}_i = \perp$.
- *Hash queries to H_1*: If A_{I} executes a H_1 query on $(\mathrm{ID}_i, m_i, R_i, Y_i)$, C searches the list $L^{\text{list}}_{H_1}$ and returns h_i if the tuple $(\mathrm{ID}_i, m_i, R_i, Y_i, h_i)$ is found, otherwise chooses a number $h_i \in_R Z_q^*$ and returns it as the answer and adds the tuple $(\mathrm{ID}_i, m_i, R_i, Y_i, h_i)$ in $L^{\text{list}}_{H_1}$.
- *Hash queries to H_2*: Suppose A_{I} submits a H_2 query for $(\mathrm{ID}_i, m_i, P_i, Y_i)$, C searches the list $L^{\text{list}}_{H_2}$ and returns t_i if a tuple $(\mathrm{ID}_i, m_i, P_i, Y_i, t_i)$ is found in $L^{\text{list}}_{H_2}$, otherwise selects a $t_i \in_R Z_q^*$ and returns it as the answer and adds the tuple $(\mathrm{ID}_i, m_i, P_i, Y_i, t_i)$ in $L^{\text{list}}_{H_2}$.
- *CL-DS-Sign queries*: It is assume that A_{I} never repeats *CL-DS-Sign* query. Suppose that A_{I} submits a *CL-DS-Sign* query for a message m_i with the signer's identity ID_i. If $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$, C then aborts the simulation, else C access the lists $L^{\text{list}}_{H_1}$, $L^{\text{list}}_{H_2}$ and L^{list}_{C}, then generates the signature as follows:
  (a) Choose a number $y_i \in_R Z_q^*$ and compute $Y_i = y_i P_i$.
  (b) Compute $\sigma_i = x_i y_i - (t_i x_i + h_i d_i) \mod q$.
  (c) Output the signature (σ_i, Y_i) on the message m_i.
- *CL-DS-Verify queries*: If A_{I} asks a *CL-DS-Verify* query on (σ_i, Y_i) for the message m_i with signer's identity ID_i, C then aborts the verification if $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$ holds. Otherwise, C verifies the signature (σ_i, Y_i) by using the *CL-DS-Verify* algorithm.

- *Forgery*: Finally, A_{I} returns a valid signature (σ_i, Y_i) on the message m with h_i for the signer ID_i. If $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$, C aborts the execution of this game. Otherwise, C searches the list $L_{H_1}^{\mathrm{list}}$ and outputs another valid signature $(\sigma_{\mathrm{S}}^*, Y_{\mathrm{S}})$ with different h_{S}^* such that $h_{\mathrm{S}} \neq h_{\mathrm{S}}^*$ on the same message m as done in forking lemma [29]. Thus, we can write $\sigma_{\mathrm{S}}P = Y_{\mathrm{S}} - t_{\mathrm{S}}P_{\mathrm{S}} - h_{\mathrm{S}}(R_{\mathrm{S}} + l_{\mathrm{S}}P_{\mathrm{pub}})$ and $\sigma_{\mathrm{S}}^*P = Y_{\mathrm{S}} - t_{\mathrm{S}}P_{\mathrm{S}} - h_{\mathrm{S}}^*(R_{\mathrm{S}} + l_{\mathrm{S}}P_{\mathrm{pub}})$. Let, $R_{\mathrm{S}} = b_{\mathrm{S}}P_{\mathrm{pub}} + a_{\mathrm{S}}P$ and $P_{\mathrm{pub}} = aP$, and then subtracting the above two equations, we get

$$
\begin{aligned}
\sigma_{\mathrm{S}}^*P - \sigma_{\mathrm{S}}P &= h_{\mathrm{S}}(R_{\mathrm{S}} + l_{\mathrm{S}}P_{\mathrm{pub}}) - h_{\mathrm{S}}^*(R_{\mathrm{S}} + l_{\mathrm{S}}P_{\mathrm{pub}})\\
&\Rightarrow (\sigma_{\mathrm{S}}^* - \sigma_{\mathrm{S}})P = h_{\mathrm{S}}(b_{\mathrm{S}}P_{\mathrm{pub}} + a_{\mathrm{S}}P + l_{\mathrm{S}}P_{\mathrm{pub}}) - h_{\mathrm{S}}^*(b_{\mathrm{S}}P_{\mathrm{pub}} + a_{\mathrm{S}}P + l_{\mathrm{S}}P_{\mathrm{pub}})\\
&\Rightarrow (\sigma_{\mathrm{S}}^* - \sigma_{\mathrm{S}})P = (h_{\mathrm{S}} - h_{\mathrm{S}}^*)(b_{\mathrm{S}} + l_{\mathrm{S}})aP + (h_{\mathrm{S}} - h_{\mathrm{S}}^*)a_{\mathrm{S}}P\\
&\Rightarrow (\sigma_{\mathrm{S}}^* - \sigma_{\mathrm{S}})P - (h_{\mathrm{S}} - h_{\mathrm{S}}^*)a_{\mathrm{S}}P = (h_{\mathrm{S}} - h_{\mathrm{S}}^*)(b_{\mathrm{S}} + l_{\mathrm{S}})aP\\
&\Rightarrow [(\sigma_{\mathrm{S}}^* - \sigma_{\mathrm{S}}) - (h_{\mathrm{S}} - h_{\mathrm{S}}^*)a_{\mathrm{S}}]P = (h_{\mathrm{S}} - h_{\mathrm{S}}^*)(b_{\mathrm{S}} + l_{\mathrm{S}})aP\\
&\Rightarrow a = [(\sigma_{\mathrm{S}}^* - \sigma_{\mathrm{S}}) - (h_{\mathrm{S}} - h_{\mathrm{S}}^*)a_{\mathrm{S}}]/[(h_{\mathrm{S}} - h_{\mathrm{S}}^*)(b_{\mathrm{S}} + l_{\mathrm{S}})].
\end{aligned}
$$

Therefore, A_{I} solves the ECDLP problem as $a = (\sigma_{\mathrm{S}}^* - \sigma_{\mathrm{S}}) - (h_{\mathrm{S}} - h_{\mathrm{S}}^*)a_{\mathrm{S}}/(h_{\mathrm{S}} - h_{\mathrm{S}}^*)(b_{\mathrm{S}} + l_{\mathrm{S}})$ using the algorithm C for given a random instance $(P, P_{\mathrm{pub}} = aP) \in G_q$. ■

Now we will analyse the probability of success and the running time of the algorithm $C(\varepsilon', t')$ in security breaking of the proposed CL-DS scheme. Assume that A_{I} can ask at most q_{H_i} times $H_i (i = 0, 1, 2)$ queries, q_C times *Create* queries, q_S times *CL-DS-Sign* queries and q_V times *CL-DS-Verify* queries. Also, we assume that A_{I} never repeats $H_i (i = 0, 1, 2)$ query with the same inputs.

(a) The execution of *Create* query succeeds with probability $(1 - q_{H_0}/q)$. Therefore, the execution of *Create* oracle is successful q_C times with the probability $(1 - q_{H_0}/q)^{q_C} \geq 1 - q_{H_0}q_C/q$.
(b) The execution of H_1 hash query gets success with probability $(1 - q_{H_1}/q)$. Thus, the execution of H_1 hash function is successful q_{H_1} times with the probability $(1 - q_{H_1}/q)^{q_{H_1}} \geq 1 - q_{H_1}^2/q$.
(c) Similarly, the simulation of H_2 hash function is successful q_{H_2} times with the probability $(1 - q_{H_2}/q)^{q_{H_2}} \geq 1 - q_{H_2}^2/q$.
(d) The execution of *CL-DS-Sign* query succeeds with probability $q_S/(1 - 1/q) \geq q_S(1 + 1/q)$.
(e) $\mathrm{ID}_i = \mathrm{ID}_{\mathrm{S}}$ holds with probability $1/q_C$.

Thus, the probability of success of A_{I} to win the *Game 1* is $\varepsilon' \geq q_S(1 - (q_{H_0}q_C/q))(1 - (q_{H_1}^2/q))(1 - (q_{H2}^2/q))(1 + (1/q))(1/q_C)\varepsilon$ and the running time is bounded by $t' < \{t + (3q_C + q_S + 4q_V)t_{\mathrm{EM}}\}$, where t_{EM} denotes the running time of executing an ECPM in group G_q. Since the probability and running time contradict the intractability of ECDLP by a polynomial time-bounded algorithm, the proposed CL-DL scheme is unforgeable in the random oracle model against the adversary A_{I} under the adaptive chosen message and identity attacks.

THEOREM 2 *In the random oracle model, if there is an adaptive chosen message and identity adversary A_{II} who can break the proposed pairing-free CL-DS scheme in polynomial time, then there exists an algorithm C that can solve the ECDLP problem in the elliptic curve group G_q.*

Proof Assume that the proposed CL-DS scheme can be forged under the adaptive chosen message and identity attacks by a polynomial time-bounded adversary A_{II}, then it is possible to construct an algorithm C that helps A_{II} to solve an instance of ECDLP, i.e. A_{II} outputs a from the input $(P, Q = aP)$, where $a \in_R Z_q^*$. To solve the ECDLP, C picks a number $\lambda \in_R Z_q^*$, sets $P_{\mathrm{pub}} = \lambda P$ and sends $\Omega = \{F_q, E/F_q, G_q, P, P_0 = \lambda P, H_0, H_1, H_2\}$ with $x = \lambda$ to A_{II}. Similar to Theorem 1, let the lists $L_{H_0}^{\mathrm{list}}$, $L_{H_1}^{\mathrm{list}}$, $L_{H_2}^{\mathrm{list}}$ and L_C^{list} are maintained by C.

- *Create(ID$_i$)*: On receiving a *Create(ID$_i$)* query, C responses in the following way:

(a) If $\text{ID}_i = \text{ID}_\text{S}$, C picks the number $a_i, l_i \in_R Z_q^*$, sets $H_0(\text{ID}_i, R_i, P_i) \leftarrow l_i$ and computes $R_i = a_i P$, $d_i = a_i + l_i\lambda$ and the public key as $P_i = aP$. The partial private key, full private key and full public key of ID_i are $D_i = (d_i, R_i)$, $\text{sk}_i = (D_i, \perp)$ and $\text{pk}_i = (P_i, R_i)$, respectively.

(b) If $\text{ID}_i \neq \text{ID}_\text{S}, C$ chooses $a_i, x_i, l_i \in_R Z_q^*$ and sets $H_0(\text{ID}_i, R_i, P_i) \leftarrow l_i$, computes $R_i = a_i P$, $d_i = a_i + l_i\lambda$ and the public key as $P_i = x_i P$. The partial private key, full private key and full public key of ID_i are $D_i = (d_i, R_i)$, $\text{sk}_i = (D_i, x_i)$ and $\text{pk}_i = (P_i, R_i)$, respectively.

Finally, C inserts the tuples $(\text{ID}_i, R_i, P_i, l_i)$ and $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ to the list $L_{H_0}^{\text{list}}$ and L_C^{list}, respectively.

- *Hash queries to* H_i *(i=0, 1, 2):* These queries are same as done in Theorem 1, so we omit them here.
- *Partial-Private-Key-Extract queries*: On receiving a query, C looks into the list L_C^{list} and returns D_i if a tuple $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ exists. Otherwise, C recalls *Create(ID$_i$)* query to obtain $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ and then returns D_i as the answer.
- *Set-Secret-Value queries*: If the adversary A_{II} asks a *Set-Secret-Value* query for ID_i, C then answers as follows:
  (a) If $\text{ID}_i = \text{ID}_\text{S}, C$ aborts the simulation.
  (b) If $\text{ID}_i \neq \text{ID}_\text{S}, C$ looks for the tuple $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ in the list L_C^{list} and outputs x_i if such tuple is available, otherwise C asks a *Create(ID$_i$)* query to obtain the tuple $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ and then outputs x_i.
- *Set-Public-Key queries*: For a query of this type, C searches the list L_C^{list} and replies with $\text{pk}_i = (R_i, P_i)$ if a tuple of the form $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ is found. Otherwise, C executes a *Create(ID$_i$)* query to obtain the tuple $(\text{ID}_i, D_i, \text{sk}_i, \text{pk}_i)$ and then returns $\text{pk}_i = (R_i, P_i)$.
- *CL-DS-Sign queries*: Suppose that A_{II} asks for signature on a message m_i chosen by himself with the signer's identity ID_i. If $\text{ID}_i = \text{ID}_\text{S}$, C aborts the simulation, else C access the lists $L_{H_1}^{\text{list}}$, $L_{H_2}^{\text{list}}$ and L_C^{list}, and then generates the signature as follows:
  (a) Choose a number $y_i \in_R Z_q^*$ and compute $Y_i = y_i P_i$.
  (b) Compute $\sigma_i = x_i y_i - (t_i x_i + h_i d_i) \mod q$.
  (c) Output the signature (σ_i, Y_i) on the message m_i.
- *CL-DS-Verify queries*: If the adversary A_{II} asks a *CL-DS-Verify* query on (σ_i, Y_i) for the message m_i with the signer's identity ID_i, C then stops the verification process if $\text{ID}_i = \text{ID}_\text{S}$ holds. Otherwise, C verifies the signature (σ_i, Y_i) using the *CL-DS-Verify* algorithm.
- *Forgery*: Eventually, A_{II} returns a valid signature $(\sigma_\text{S}, Y_\text{S})$ on the message m with the hash value t_S for the signer ID_i If $\text{ID}_i = \text{ID}_\text{S}$, C then stops the process. Otherwise, C searches the list $L_{H_2}^{\text{list}}$ and outputs another valid signature $(\sigma_\text{S}^*, Y_\text{S})$ with a hash value t_S^* such that $t_\text{S} \neq t_\text{S}^*$ on the same message m as done in forking lemma [29]. Then, C can write $\sigma_\text{S} P = Y_\text{S} - t_\text{S} P_\text{S} - h_\text{S}(R_\text{S} + l_\text{S} P_{\text{pub}})$ and $\sigma_\text{S}^* P = Y_\text{S} - t_\text{S}^* P_\text{S} - h_\text{S}(R_\text{S} + l_\text{S} P_{\text{pub}})$, and after subtracting, C obtains

$$\sigma_\text{S}^* P - \sigma_\text{S} P = (t_\text{S} - t_\text{S}^*) P_\text{S}$$

$$\Rightarrow (\sigma_\text{S}^* - \sigma_\text{S}) P = (t_\text{S} - t_\text{S}^*) aP$$

$$\Rightarrow a = (\sigma_\text{S}^* - \sigma_\text{S})/(t_\text{S} - t_\text{S}^*).$$

Therefore, for given a random instance $(P, aP) \in G_q$, C outputs $a = (\sigma_\text{S}^* - \sigma_\text{S})/(t_\text{S} - t_\text{S}^*)$ as the solution of the ECDLP. According to Theorem 1, the adversary A_{II} wins the *Game 2* with probability $\varepsilon' \geq q_S\,(1 - (q_{H_0} q_C/q))(1 - (q_{H_1}^2/q))(1 - (q_{H2}^2/q))(1 + (1/q))(1/q_C)\varepsilon$ and the running time is $t' < \{t + (3q_C + q_S + 4q_V)t_{\text{EM}}\}$. Thus, the proposed CL-DS scheme is secure in the random oracle model against the adaptive chosen message and identity adversary A_{II}. ■

Table 1. Definition and conversion of various operational time units.

Notations	Definition and conversion
T_{ML}	Time complexity for executing the modular multiplication operation
T_{EM}	Time complexity for executing the ECPM, $T_{\mathrm{EM}} \approx 29\,T_{\mathrm{ML}}$ [10,22]
T_{BP}	Time complexity for executing the bilinear pairing operation, $T_{\mathrm{BP}} \approx 3\,T_{\mathrm{EM}} \approx 87\,T_{\mathrm{ML}}$ [7,18,34]
T_{EA}	Time complexity for executing the addition of two elliptic curve points, $T_{\mathrm{EA}} \approx 0.12\,T_{\mathrm{ML}}$ [10,22]
T_{MTP}	Time complexity for executing the MTP function, $T_{\mathrm{MTP}} \approx 1\,T_{\mathrm{EM}} \approx 29\,T_{\mathrm{ML}}$ [4,5]
T_{PX}	Time complexity for executing pairing-based exponentiation, $T_{\mathrm{PX}} \approx 1/2\,T_{\mathrm{BL}} \approx 43.5\,T_{\mathrm{ML}}$ [10,22]
T_{IN}	Time complexity for executing the modular inversion operation, $T_{\mathrm{IN}} \approx 11.6\,T_{\mathrm{ML}}$ [13,15]
T_{H}	Time complexity for executing the simple hash function, which is negligible [10,22]

Table 2. Security comparison of the proposed scheme with others.

	Security weaknesses		
Schemes	Against A_{I}	Against A_{II}	Security assumption
Al-Riyami and Paterson [1]	Yes	No	BDH
Zhang *et al.* [40]	No	No	CDH
Huang *et al.* [20]	No	Yes	CDH
Gorantla and Saxena [14]	Yes	No	BDH
Yap *et al.* [36]	Yes	No	CDH
Choi *et al.* [9]	Yes	No	CDH
Xu *et al.* [35]	No	Yes	CDH
Zhang and Zhang [41]	No	No	CDH
Li and Liu [26]	No	No	BDH
Proposed	No	No	ECDLP

Table 3. Efficiency comparison of the proposed scheme with others.

	Computation cost			
Schemes	Signing cost	Verification cost	Total cost	Signature length
Al-Riyami and Paterson [1]	$2T_{\mathrm{EM}} + T_{\mathrm{BP}} + T_{\mathrm{EA}}$	$T_{\mathrm{BP}} + T_{\mathrm{PX}}$	$2T_{\mathrm{EM}} + 5\,T_{\mathrm{BP}} + T_{\mathrm{PX}} + T_{\mathrm{EA}} \approx 536\,T_{\mathrm{ML}}$	$2\|G_q\|$
Zhang *et al.* [40]	$3T_{\mathrm{EM}} + 2\,T_{\mathrm{MTP}} + T_{\mathrm{EA}}$	$4T_{\mathrm{BP}} + T_{\mathrm{MTP}}$	$3T_{\mathrm{EM}} + 4\,T_{\mathrm{BP}} + 3\,T_{\mathrm{MTP}} + T_{\mathrm{EA}} \approx 551\,T_{\mathrm{ML}}$	$2\|G_q\|$
Huang *et al.* [20]	$T_{\mathrm{EM}} + T_{\mathrm{MTP}} + T_{\mathrm{EA}}$	$3T_{\mathrm{BP}} + T_{\mathrm{MTP}}$	$T_{\mathrm{EM}} + 3\,T_{\mathrm{BP}} + 2\,T_{\mathrm{MTP}} + T_{\mathrm{EA}} \approx 348\,T_{ML}$	$2\|G_q\|$
Gorantla and Saxena [14]	$3T_{\mathrm{EM}} + T_{\mathrm{EA}}$	$3T_{\mathrm{BP}} + T_{\mathrm{EM}} + T_{\mathrm{EA}}$	$2T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 2\,T_{\mathrm{MTP}} + 2\,T_{\mathrm{EA}} \approx 290\,T_{ML}$	$2\|G_q\|$
Yap *et al.* [36]	$2T_{\mathrm{EM}}$	$T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + T_{\mathrm{MTP}} + 2\,T_{\mathrm{EA}}$	$3T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + T_{\mathrm{MTP}} + 2\,T_{\mathrm{EA}} \approx 290\,T_{ML}$	$2\|G_q\|$
Choi *et al.* [9]	$2T_{\mathrm{EM}}$	$2\,T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 2\,T_{\mathrm{MTP}} + 2\,T_{\mathrm{EA}}$	$4T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 2\,T_{\mathrm{MTP}} + 2\,T_{\mathrm{EA}} \approx 348\,T_{ML}$	$2\|G_q\|$
Xu *et al.* [35]	$2T_{\mathrm{EM}}$	$3\,T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 2\,T_{\mathrm{EA}}$	$5T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 2\,T_{\mathrm{EA}} \approx 319\,T_{\mathrm{ML}}$	$2\|G_q\|$
Zhang and Zhang [41]	$3T_{\mathrm{EM}} + 2\,T_{\mathrm{EA}}$	$2T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + T_{\mathrm{MTP}} + 2\,T_{\mathrm{EA}}$	$5T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + T_{\mathrm{MTP}} + 4\,T_{\mathrm{EA}} \approx 348\,T_{\mathrm{ML}}$	$2\|G_q\|$
Li and Liu [26]	$2T_{\mathrm{EM}} + T_{\mathrm{EA}}$	$T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 2\,T_{\mathrm{EA}}$	$3T_{\mathrm{EM}} + 2\,T_{\mathrm{BP}} + 3\,T_{\mathrm{EA}} \approx 261\,T_{\mathrm{ML}}$	$2\|G_q\|$
Proposed	T_{EM}	$4T_{\mathrm{EM}} + 3\,T_{\mathrm{EA}}$	$5T_{\mathrm{EM}} + T_{\mathrm{EA}} \approx 145\,T_{\mathrm{ML}}$	$2\|G_q\|$

6. Performance analysis and comparison with others

This section analyses the proposed CL-DS scheme and compares with other relevant schemes [1,9, 14,20,26,35,36,40,41] in terms of security, signature length, hardness assumption and computation cost. For the performance comparison, we define different time complexity notations and their conversions in terms of T_{ML} as shown in Table 1. Tables 2 and 3 show the sum up of the performance of the competitive schemes including ours. Note that the bilinear paring $\hat{e} : G_q \times G_q \rightarrow G_m$ and the MTP hash function are used in the previous schemes for their implementation, where G_q is an additive elliptic curve cyclic group of prime order q and G_m is a multiplicative cyclic group of the same order q. For comparison, let us assume $|G_q| = |G_M|$, where $|x|$ indicates the bit length of $x \in G_q$. The computation cost and the signature length of the proposed scheme are calculated and found to be $145T_{\mathrm{ML}}$ and $2|G_q|$, respectively. It can be seen from comparing Tables 2 and 3 that the proposed scheme not only provides the computation efficiency, but also achieves the strong security in the random oracle model without using bilinear pairing and MTP hash function.

7. Conclusions

In recent years, the CL-PKC scheme has received much attention of many researchers as it eliminates the certificate management problems occurring in traditional PKI and the private key escrow problem of IBC schemes. In this paper, we proposed an efficient CL-DS scheme using ECC for the message integrity, non-repudiation and authentication and it is proven to be existentially unforgeable in the random oracle model under the adaptive chosen message and identity attacks against the different adversaries with different attack powers. However, it may be noted that the security of the proposed scheme is based on the intractability of ECDLP problem. The proposed scheme is easily implementable as no bilinear pairing and MTP hash function have been used. Security and computation cost comparisons of our signature scheme with other existing schemes prove to be secured and efficient. Due to the low computation cost and strong security features, the proposed scheme is applicable in the areas where the communication bandwidth, computation cost and storage space are limited.

Acknowledgements

The authors are extremely grateful to Prof. George Loizou, Editor-in-Chief and anonymous reviewers for their insightful comments and suggestions that help us to improve this paper. The financial support of this work is provided by the Department of Science and Technology (DST), Govt. of India under the INSPIRE fellowship Ph.D. program (Reg. No. IF10247) and the Department of Information Technology (DIT), Ministry of Communication and Information Technology, Govt. of India under the 'Information Security Education and Awareness (ISEA)' program (No. MIT (2)/2006-08/189/CSE). The authors are also thankful to the Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, for providing their research support, without such co-operation this work could not have been made.

References

[1] S. Al-Riyami and K. Paterson, *Certificateless Public Key Cryptography*, Proceedings of the Asiacrypt'03, LNCS 2894, Springer-Verlag, Berlin, 2003, pp. 452–473.

[2] J. Baek, R. Safavi-Naini, and W. Susilo, *Certificateless Public Key Encryption Without Pairing*, Proceedings of the 8th Information Security Conference (ISC'05), LNCS 3650, Singapore, September 20–23, 2005, pp. 134–148.

[3] M. Ballare and P. Rogaway, *Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols*, Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS'93), Fairfax, VA, USA, November 3–5, 1993, pp. 62–73.

[4] P. Barreto, H. Kim, B. Lynn, and M. Scott, *Efficient Algorithms for Pairing-Based Cryptosystems*, Proceedings of the Advances on Cryptology (Crypto'02), LNCS 2442, Springer-Verlag, Berlin, 2002, pp. 354–368.

[5] P. Barreto, B. Lynn, and M. Scott, *On the Selection of Pairing-Friendly Groups*, Proceedings of the Selected Areas in Cryptography (SAC'03), LNCS 3006, Springer-Verlag, Berlin, 2004, pp. 17–25.

[6] D. Boneh and M.K. Franklin, *Identity-Based Encryption from the Weil Pairing*, Proceedings of the Advances in Cryptology (Crypto'01), LNCS 2139, Springer-Verlag, Berlin, 2001, pp. 213–229.

[7] X. Cao, W. Kou, and X. Du, *A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges*, Inf. Sci. 180 (2010), pp. 2895–2903.

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

[9] K. Choi, J. Park, J. Hwang, and D. Lee, *Efficient Certificateless Signature Schemes*, Proceedings of the ACNS'07, LNCS 4521, Springer-Verlag, Berlin, 2007, pp. 443–458.

[10] Y.F. Chung, K.H. Huang, F. Lai, and T.S. Chen, *ID-based digital signature scheme on the elliptic curve cryptosystem*, Comput. Stand. Interfaces 29 (2007), pp. 601–604.

[11] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Trans. Inf. Theory 22(6) (1976), pp. 644–654.

[12] T. ElGamal, *A public key cryptosystem and a signature protocol based on discrete logarithms*, IEEE Trans. Inf. Theory 31 (1985), pp. 469–472.

[13] A.W. Fan and S.X. Lu, *An improved elliptic curve digital signature algorithm*, Appl. Mech. Mater. 34–35 (2010), pp. 1024–1027.

[14] M.C. Gorantla and A. Saxena, *An Efficient Certificateless Signature Scheme*, Proceedings of the International Conference on Computational Intelligence and Security, LNAI 3802, Springer-Verlag, Berlin, 2005, pp. 110–116.

[15] H. Guozheng and H. Fan, A*ttacks Against Two Provably Secure Certificateless Signature Schemes*, Proceedings of the WASE International Conference on Information Engineering, Taiyuan, Chanxi, 2009, pp. 246–249.

[16] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag, New York, 2004.

[17] D. He, Y. Chen, J. Chen, R. Zhang, and W. Han, *A new two-round certificateless authenticated key agreement protocol without bilinear pairings*, Math. Comput. Model. 54 (2011), pp. 3143–3152.

[18] M. Hölbl, T. Welzer, and B. Brumen, *Two proposed identity-based three-party authenticated key agreement protocols from pairings*, Comput. Secur. 29(2) (2010), pp. 244–252.

[19] B. Hu, D. Wong, Z. Zhang, and X. Deng, *Key Replacement Attack Against a Generic Construction of Certificateless Signature*, Proceedings of the ACISP'06, LNCS 4058, Springer-Verlag, Berlin, 2006, pp. 235–346.

[20] X. Huang, Y. Mu, W. Susilo, D.S. Wong, and W. Wu, *Certificateles Signature Revisited*, Proceedings of the ACISP'07, LNCS 4586, Springer-Verlag, Berlin, 2007, pp. 308–322.

[21] X. Huang, W. Susilo, Y. Mu, and F. Zhang, *On the Security of a Certificateless Signature Scheme*, Proceedings of the CANS'05, LNCS 3810, Springer-Verlag, Berlin, 2005, pp. 13–25.

[22] S.H. Islam and G.P. Biswas, *A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile network*, Ann. Telecommun. 67(11) (2012), pp. 547–558.

[23] S.H. Islam and G.P. Biswas, *An Improved Pairing-Free Identity-Based Authenticated Key Agreement Protocol based on ECC*, Proceedings of the International Conference on Communication Technology and System Design (ICCTSD 2011), Coimbatore, Tamil Nadu, India. Procedia Engineering 30 (2012), pp. 499–507.

[24] N. Koblitz, *Elliptic curve cryptosystem*, J. Math. Comput. 48(177) (1987), pp. 203–209.

[25] L. Lamport, *Constructing digital signatures from a one-way function*, Technical Report CSL-98, SRI International Computer Science Laboratory, 1979.

[26] F. Li and P. Liu, *An Efficient Certificateless Signature Scheme from Bilinear Parings*, International Conference on Network Computing and Information Security, Guilin, China, 2011, pp. 35–37.

[27] R. Merkle, *A Certified Digital Signature*, Proceeding of the Advances in Cryptology (Crypto'89), LNCS 435, Spring Verlag, Berlin, 1990, pp. 218–238.

[28] V.S. Miller, *Use of Elliptic Curves in Cryptography*, Proceeding of the Advances in Cryptology (Crypto'85), Springer-Verlag, New York, 1985, pp. 417–426.

[29] D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, J. Cryptol. 13 (2000), pp. 361–396.

[30] M.O. Rabin, *Digitalized signatures as intractable as factorization*, Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[31] R.L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, Commun. ACM, 21(2) (1978), pp. 120–126.

[32] C.P. Schnorr, *Efficient Identification and Signatures for Smart Cards*, Proceeding of the Advances in Cryptology (Crypto'89), LNCS 435, Springer-Verlag, Berlin, 1990, pp. 239–251.

[33] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*, Proceeding of the Advances in Cryptology (Crypto'84), Springer-Verlag, Berlin, 1984, pp. 47–53.

[34] S.-Y. Tan, S.-H. Heng, and B.-M. Goi, *Java Implementation for Pairing-Based Cryptosystems*, Proceedings of the ICCSA'10, LNCS 6019, Springer-Verlag, Berlin, 2010, pp. 188–198.

[35] Z. Xu, X. Liu, G. Zhang, W. He, G. Dai, and W. Shu, *A Certificateless Signature Scheme for Mobile Wireless Cyber-Physical Systems*, Proceedings of the International Conference on Distributed Computing Systems Workshops (ICDCS'08), Beijing, China, 2008, pp. 489–494.

[36] W. Yap, S. Heng, and B. Goi, *An Efficient Certificateless Signature Scheme*, Proceedings of the EUC Workshops 2006, LNCS 4097, Springer-Verlag, Berlin, 2006, pp. 322–331.

[37] D. Yum and P. Lee, *Generic Construction of Certificateless Signature*, Proceedings of the ACISP'04, LNCS 3108, Springer-Verlag, Berlin, 2004, pp. 200–211.

[38] Z. Zhang and D. Feng, *Key replacement attack on a certificateless signature scheme*, Report 2006/453, Cryptology ePrint Archive, 2006.

[39] F. Zhang, S. Li, S. Miao, Y. Mu, W. Susilo, and X. Huang, *Cryptanalysis on two certificateless signature schemes*, Int. J. Comput. Commun. Control V (4) (2010), pp. 586–591.

[40] Z. Zhang, D. Wong, J. Xu, and D. Feng, *Certificateless Public-Key Signature: Security Model and Efficient Construction*, Proceedings of the ACNS 2006, LNCS 3989, Springer-Verlag, Berlin, 2006, pp. 293–308.

[41] L. Zhang and F. Zhang, *A New Provably Secure Certificateless Signature Scheme*, Proceedings of the IEEE International Conference on Communications (ICC'08), Beijing, China, 2008, pp. 1685–1689.