ISSN (Online) : 2454 -7190 **ISSN (Print) 0973-8975**
Copyright reserved © J.Mech.Cont.& Math. Sci., Vol.-14, No.2, March-April (2019) pp 335-353

Identity-Based Directed Signature Scheme without Bilinear Pairings

[1]R. R. V. Krishna Rao, [2]N. B. Gayathri, [3]P. Vasudeva Reddy

[1]Department of Engineering Mathematics, Andhra University, Visakhapatnam, India;

[2]Department of Engineering Mathematics, Andhra University, Visakhapatnam, India;

[3]Department of Engineering Mathematics, Andhra University, Visakhapatnam, India;

[1]rrvkrisharao@gmail.com, [2]gayatricrypto@gmail.com, [3]vasucrypto@andhrauniversity.edu.in

***** Corresponding Author: P. Vasudeva Reddy

https://doi.org/10.26782/jmcms.2019.04.00027

Abstract

The most important contribution of modern cryptography is the invention of digital signatures. Digital signature schemes have been extended to meet the specific requirements for real world applications. A directed signature scheme is a kind of signature scheme intended to protect the privacy of the signature verifier. In directed signature schemes, a signer signs the document/message for a designated verifier so that only the designated verifier can verify the validity of the signature and others cannot do. Thus the restriction of verification is controlled by the signer. Such directed signature schemes are applicable in many situations where the signed message is sensitive to the receiver such as signature on medical records, tax information etc. However all the existing directed signature schemes in ID based setting uses bilinear pairings over elliptic curves. Due to the heavy computational cost of pairing operations, these existing ID based directed signature schemes are not much efficient in practice. In order to improve the efficiency, in this paper, we present an efficient Identity-based directed signature scheme without pairings. The proposed scheme is proven secure under the assumption of elliptic curve discrete logarithm problem is hard. In addition, this scheme improves the efficiency than the existing directed signature schemes in terms of computational cost.

Keywords : Digital signature; Directed Signature; Elliptic Curve Discrete Logarithm Problem; Identity-based Framework; Random Oracle Model.

I. Introduction

Digital signature is a cryptographic primitive which provides data integrity, authentication and non-repudiation in digital communications. The concept of Public Key Cryptography (PKC) was introduced by Diffie and Hellman [XXII] in 1976, in which each user has a pair of public and private key. The authentication of these public keys relies on the certificate issued by Certificate Authority (CA). In multi user environment, the authentication, revocation, storage of public keys leads to large computation and communication costs and hence managing these public keys is a big problem. To avoid such difficulties in PKC, in 1984, Shamir [I] introduced Identity based Public Key Cryptography (ID-PKC). In this system, the user's unique information/identity is the public key and the private key is generated by the KGC using this identity. To satisfy different applications, many ID-based signature schemes such as Proxy signature, Blind signature, Multi signature, Group signature, Ring signature etc have been proposed. One of such variants is Directed signature.

In directed signature schemes, a signer sign the document/message intended to a designated verifier so that only the designated verifier can verify the validity of the signature and others cannot do. In case of trouble or if necessary the validity can be verified by any third party with the help of the aid provided by the signer or the designated verifier. Thus the restriction of verification is controlled by the signer. Such directed signature schemes are applicable in many situations where the signed message is sensitive to the receiver such as signature on medical records, tax information etc.

Related work

The first directed signature was proposed in 1992 by Lim and Lee in [III]. This scheme is based on GQ signature scheme [X]. In 2004, Sundarlal et al. [XVIII] presented a directed signature scheme on Public Key Infrastructure (PKI) setting based on the Schnorr signature scheme [IV]. In 2005, an universally convertible directed signature scheme was designed by Laguillaumie et al. [VII]. In 2006, an RSA based directed signature scheme was presented by Lu and Cao [XVII]. In 2007 E.S. Ismail et al. [VI] and in 2009, Wei et al. [XVI] presented a DLP based directed signature scheme for confidential group communication. In 2013, N. N. Ramlee et al. [XIII] designed a new directed signature scheme. All these schemes are in PKI based setting.

In 2005, Wang [XXV] proposed the first ID-based directed signature scheme. In 2008, X. Sun et al. [XXIV] proposed a directed signature scheme in ID-based setting using bilinear pairings. Zhang et al. [IX] designed a new directed signature scheme in ID-based setting without random oracles in 2009. In the same year, B.U. Rao et al. [II] presented a new and efficient directed signature in ID-based setting using bilinear pairings over elliptic curves. In 2012, J. Ku et al. [VIII] proposed an efficient ID-based directed signature scheme on hyper elliptic curves.

All the existing directed signature schemes in ID-based setting are designed using bilinear pairings over elliptic curves. But the computation of bilinear pairing is very costly. Hence, the schemes without bilinear pairing under general hash function would be more desirable. Elliptic Curve Cryptography (ECC) provides the same level of security as other systems with smaller key size along with efficiency. Recently, in this direction, Gayathri et al. [XI] proposed a novel pairing-free directed signature scheme in certificateless based setting and proved its security in random oracle model under the assumption that elliptic curve discrete logarithm problem is intractable. With this motivation, in this paper we design a new directed signature scheme without pairings in ID-based setting.

Our contribution

In this paper, we present an efficient Identity Based Directed Signature (ID-DS) scheme without using bilinear pairings over elliptic curves. To the best of our knowledge this is the first scheme in identity based setting addressing about directedness in pairing free environment. We formally prove the security of the proposed scheme in random oracle paradigm under the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP).

Organization

The rest of the part of this paper is arranged as follows. In Section II we presented some preliminaries on elliptic curves and some complexity assumptions. In Section III we presented the syntax/frame work and security model for our ID-DS scheme. In Section IV we presented our proposed ID-DS scheme with its security analysis. Section V deal with efficiency analysis of the proposed scheme. Finally, Section VI concludes the paper.

II. Preliminaries

In this section we briefly describe the fundamental concepts on elliptic curve and the complexity assumption, on which the proposed scheme is designed and achieves the desired security.

Elliptic Curve Group

Let the symbol E/F_P denote an elliptic curve E over a prime finite field F_P, defined by an equation $y^2 = (x^3 + ax + b),\ a,b \in F_P$ and with the discriminant $\Delta = 4a^3 + 27b^2 \neq 0.$ The points on E/F_P together with an extra point 'O' called the point at infinity form a group $G = \{(x,y) : x,y \in F_P, E(x,y) = 0\} \cup \{O\}.$ Now G forms an additive group with point addition. For further details, please refer [XII, XXI, XIV].

Elliptic Curve Discrete Logarithm Problem (ECDLP):

Let G be an additive group over elliptic curve with prime order $q.$ Let P be a generator of G then the ECDLP is defined as follows. For a random instance $Q \in G$ and $Q = xP$ where $x \in Z_q^*,$ compute x from P and $Q.$ We consider the advantage of an algorithm to solve the ECDLP as $ADV_n ECDLP = \Pr[A(P,Q) = x].$

ECDLP assumption

For every PPT algorithm, the $ADV_n ECDLP$ is negligible.

Notations and their meanings that are used in this paper are presented in the following Table 1.

Table 1: Notations and their meanings

Notation	Meaning
n, s	Security parameter and Master secret key of the system generated by Private Key Generator (PKG).
τ	System Parameters.
z_q^*	The group with elements 1,2…q-1 under addition modulo q.
G	Additive cyclic group of prime order q.
H_1, H_2, H_3	Cryptographic one way hash functions.
ID_s, ID_v	Signer's identity and designated verifiers identity respectively.
D_i	Private key of the identity.
ADV	Adversary.
ξ	An algorithm to solve ECDL problem by using adversary.
D	A distinguisher to distinguish a valid signature on an adaptively chosen

	message by the attacker from one randomly drawn from the signature space.
σ_s	Signature on a message.

Notations and Acronyms

The following Table 2 presents the acronyms that are used throughout this paper.

Table 2: Acronyms and explanation.

Acronyms	Explanation
DS	Directed Signature
ECDLP	Elliptic Curve Discrete Logarithm Problem
PKC	Public Key Cryptography
ID-DS	Identity Based Directed Signature
ECC	Elliptic Curve Cryptography
PPT	Probabilistic Polynomial Time
PKG	Private Key Generator
ROM	Random Oracle Model
IFP	Integer Factorization Problem
DLP	Discrete Logarithm Problem
CDHP	Computational Diffie Hellman Problem
DBDHP	Decisional Bilinear Diffie Hellman Problem

III. Syntax and Security Model of the Proposed PF-ID-DS-MR Scheme

This section presents the syntax and security model for our pairing free ID-DS scheme.

Syntax of ID-DS Scheme

A formal model of the proposed ID-DS scheme consists of five components whose functionalities are described as follows.

- **Setup:** This algorithm takes $1^n, n \in Z^+$ as the security parameter and outputs a master public/secret key pair and publicly known system parameters *params*.
- **Extract:** For a given *params*, master secret key, *ID* as input, PKG run this algorithm and generates private key D_i.
- **Signature Generation:** To sign a message $m \in \{0,1\}^*$ for a designated user with identity ID_v this algorithm takes *params*, ID_s, D_s, ID_v and message $m \in \{0,1\}^*$ as input and outputs a signature σ_s.

- **Direct Verification (D. Verify):** This algorithm is run by the designated verifier with inputs signature σ_s on a message *m, params,* σ_s, ID_s, ID_v. It outputs '*accept*' if σ_s is valid; or '*reject*', otherwise.
- **Public Verification (P. Verify):** To verify a signature σ_s on a message *m,* this algorithm takes *params,* σ_s, ID_s, ID_v and an ***Aid*** provided by the signer ID_s or the designated verifier ID_v as input, and outputs '*accept*' if σ_s is valid; or '*reject*', otherwise.

Security Model of ID-DS Scheme

We consider the security notions such as unforgeability and invisibility of an ID-DS scheme.

Definition 1: Unforgeability: An ID-DS scheme is said to be existentially unforgeable under adaptive chosen message and identity attack, if no PPT algorithm has a non-negligible advantage in the following game. Game I is played between a challenger and adversary.

Game I: This game is executed between the challenger ξ and an adversary ADV as follows.

Initialization Phase: The challenger ξ runs Setup algorithm gives *params* to ADV and keeps the Master secret key secret.

Queries Phase: In this phase, ADV issues the following queries.

Key Extraction Oracle: On receiving a query from ADV, the challenger ξ computes D_i by taking ID_i as input and gives this to ADV.

Sign Oracle: On receiving a query from adversary ADV with (ID_s, ID_v, m), signing oracle returns a valid signature σ_s signed by the user ID_s, by taking ID_s, ID_v with message $m \in \{0,1\}^*$ as input.

D.Verify Oracle: On receiving a query from adversary ADV with $(ID_s, ID_v, m, \sigma_s)$, ξ checks the validity of the signature by extracting ID_v's private key D_v. It outputs 1 if the signature is valid. Otherwise returns 0.

P.Verify Oracle: When ADV issues a query on $(ID_s, ID_v, m, \sigma_s)$, ξ checks the validity of the signature and returns $\perp$ to ADV if σ_s is invalid. Otherwise, ξ produces an **Aid** in the name of the signer ID_s or the designated verifier ID_v, then forwards **Aid** to ADV.

Forgery Phase: Finally, ADV outputs a forged tuple $(ID_s^*, ID_v^*, m^*, \sigma_s^*)$ and wins the game if

(i) σ_s^* is a valid signature.

(ii) (ID_s^*, ID_v^*, m^*) has never been queried to the Key Extraction Oracle and (ID_s^*, ID_v^*, m^*) has never been queried to the Sign Oracle.

Definition 2: Invisibility: An ID-DS scheme is said to have the property of invisibility under chosen message and identity attack, if there no PPT distinguisher D has a non-negligible advantage in the following game II.

Game II: This game is executed between the challenger ξ and a distinguisher D as follows.

Initialization Phase: Same as in Game I.

Phase 1: D performs a series of queries to the challenger as mentioned in Game I. The Challenger responds to these queries as in Game I.

Challenge: After Phase I is over, D submits ID_s, ID_v, and a message m to the challenger under the following conditions: ID_v^* has not been submitted to Key Extraction Oracle for D. The Challenger then generates a random bit $b \in \{0,1\}$ and produces a signature σ_s^* as in the Sign Oracle if $b = 1$. Otherwise, it picks a random σ_s^* from the signature space. In both cases σ_s^* is forwarded to D.

Phase 2: D again makes a series of queries as in Phase 1, subjected to the following conditions:

D cannot run Key Extraction Oracle on ID_v^*; and D.Verifiy or P.Verifiy Oracle on $(ID_s^*, ID_v^*, m^*, \sigma_s^*)$.

Guess: Finally D outputs a bit $b' \in \{0,1\}$. D succeeds if $b = b'$.

IV. Proposed ID-DS Scheme without Bilinear Pairings

In this section we propose our efficient ID-DS scheme and we prove its security.

Proposed ID-DS Scheme

As discussed in section III, the proposed ID-DS scheme consists of the following algorithms.

- **Setup:** Given a security parameter $n \in Z^+$, PKG does the following.

  (i) PKG chooses (q, P, G) according to n, where q is a prime, G is additive cyclic elliptic curve group, P is the generator of G.

  (ii) PKG selects a random $s \in Z_q^*$ as the master secret key and sets master public key as $P_{Pub} = sP$.

  (iii) Choose four cryptographic hash functions $H_1, H_2, H_3 : \{0,1\}^* \to Z_q^*$. PKG publishes the system parameters as $\tau = \{q, G, P, P_{Pub}, H_1, H_2, H_3\}$ and keeps s secret.

- **Extract:** PKG runs this algorithm by taking ID_i and system parameters τ as input. PKG chooses a random number $r_i \in Z_q^*$, and computes $R_i = r_i P$, $h_{1i} = H_1(ID_i, R_i, P_{Pub})$ and $d_i = r_i + sh_{1i} \bmod q$. PKG sends $D_i = (d_i, R_i)$ to the user securely. User keeps d_i as his private key and publishes R_i. The user can validate D_i by checking whether the equation $d_i P = R_i + h_{1i} P_{Pub}$ holds or not.

- **Signature Generation:** To generate a valid directed signature on a given message $m \in \{0,1\}^*$, signer takes designated verifier identity ID_v, R_v and signer's identity ID_s with signing key D_s as input along with τ, m and does as follows.

  (i) The signer chooses $t_1, t_2 \in Z_q^*$ and computes $U_s = t_1 P, V_s = t_2 P$,
  $W_s = U_s + R_v + h_{1v} P_{Pub}$ and $h_2 = H_2(m, ID_s, ID_v, U_s, R_s)$ and
  $h_3 = H_3(m, ID_s, ID_v, U_s, R_s, h_2)$.

  (ii) The signer computes $k_s = h_2 d_s + h_3 t_2 \bmod q$.
  Now $\sigma_s = (W_s, V_s, k_s)$ is the signature on a message m. The signer sends this signature to the designated verifier.

- ***D. Verification:*** Given a signature σ_s, a signer's identity ID_s, a verifier's identity ID_v and a message m, the designated verifier can verify the signature as follows.
  (i) Compute
  $$Y_v = W_s - d_v P = (U_s + R_v + h_{1v} P_{Pub}) - (r_v + h_{1v} s)P = U_s + R_v + h_{1v} P_{Pub} - R_v - h_{1v} P_{Pub} = U_s.$$
  (ii) Compute $h_2 = H_2(m, ID_s, ID_v, Y_v, R_s)$ and $h_3 = H_3(m, ID_s, ID_v, Y_v, R_s, h_2)$.
  (iii) Checks whether the equation $(k_s P - (R_s + h_{1s} P_{Pub})h_2)h_3^{-1} = V_s$ holds or not. If the equation holds, verifier accepts the signature $\sigma_s = (W_s, V_s, k_s)$ and outputs 1; rejects and outputs 0 otherwise.
- **P. Verification:** Given a purported signature $\sigma_s = (W_s, V_s, k_s)$ on a message m for the signer identity ID_s and the verifier identity ID_v, it works as follows.
  (i) Either ID_s or ID_v computes $Aid = U_s = Y_v$, and then sends to the third party (TP).
  (ii) TP computes $h_2 = H_2(m, ID_s, ID_v, Y_v, R_s)$ and $h_3 = H_3(m, ID_s, ID_v, Y_v, R_s, h_2)$.
  (iii) Checks whether the equation $(k_s P - (R_s + h_{1s} P_{Pub})h_2)h_3^{-1} = V_s$ holds or not. If the equation holds, verifier accepts the signature $\sigma_s = (W_s, V_s, k_s)$ and outputs 1; rejects and outputs 0 otherwise.

Proof of correctness of the proposed scheme:

The correctness of the presented scheme can be verified as follows.

$$\begin{aligned}
&(k_s P - (R_s + h_{1s} P_{Pub} + X_s)h_2)h_3^{-1} \\
&= \left((h_2 d_s + h_3 t_2)P - (R_s + h_{1s} P_{Pub})h_2\right)h_3^{-1} \\
&= \left((h_2(r_s + s h_{1s}) + h_3 t_2)P - (R_s + h_{1s} P_{Pub})h_2\right)h_3^{-1} \\
&= \left(h_2(R_s + h_{1s} P_{Pub}) + h_3 t_2 P - (R_s + h_{1s} P_{Pub})h_2\right)h_3^{-1} \\
&= \left(h_3 t_2 P\right)h_3^{-1} \\
&= V_s.
\end{aligned}$$

Security of our ID-DS Scheme

In this section we prove the security of the proposed ID-DS scheme in the random oracle model under the assumption that the ECDLP is intractable.

Theorem 1: The proposed PF ID-DS scheme is secure and unforgeable in the ROM under the hardness of ECDLP. If an adversary can break the unforgeability of the

proposed ID-DS scheme, then there is an algorithm which can solve the ECDL problem.

Proof:

Let ξ be an ECDL challenger and is given a random instance $(Q = sP)$ of the ECDL problem in G for a randomly chosen $s \in Z_q^*$. Its goal is to compute s. Let ADV is an adversary who interacts with ξ by performing oracle queries as modeled in [XXIV]. Now we prove that ξ can solve the ECDLP using ADV. During the simulation process ξ needs to guess the target identity of ADV. Without loss of generality, ξ takes ID^* as target identity of ADV on a message m^*.

- **Initialization Phase:** Algorithm ξ sets $P_{Pub} = Q = sP$ and runs **Setup** to generate τ. ξ then gives τ and P_{Pub} to ADV.
- **Query Phase:** In this phase, ADV performs the oracle simulation and ξ responds to these oracles as follows.

  Queries on oracle $H_1\left(H_1(ID_i, R_i, P_{Pub})\right)$*:* A list $\mathcal{L}_1$, with records of the form $(ID_i, R_i, P_{Pub}, l_{1i})$, is maintained by ξ. After receiving a query on $H_1(ID_i, R_i, P_{Pub})$, if there is a record $(ID_i, R_i, P_{Pub}, l_{1i})$ in $\mathcal{L}_1$, ξ returns l_{1i}. Otherwise, ξ picks a random l_{1i} and adds to $\mathcal{L}_1$. Finally, ξ returns l_{1i}.

  Some time ADV can query for the public key component corresponding to identity ID_i as ADV wants to know the actual R_i corresponding to ID_i. ξ does the following.

(i) If $ID_i = ID^*$, ξ sets $R_i = sP = P_{Pub}$ where s is unknown to ξ and P_{Pub} is the ECDL problem that ξ wants to solve. ξ stores the record $(ID_i, R_i, \perp, P_{Pub}, l_{1i})$ to $\mathcal{L}_1$, and returns R_i to ADV.

(ii) If $ID_i \neq ID^*$, choose $r_i \in Z_q^*$ and set $R_i = r_i P - l_{1i} P_{Pub}$ and stores the record $(ID_i, R_i, r_i, P_{Pub}, l_{1i})$ to $\mathcal{L}_1$, and returns R_i to ADV.

Queries on oracle H_2 $\left(H_2(m, ID_s, ID_v, U_s, R_s)\right)$*:* When ADV submits a query on $(m, ID_s, ID_v, U_s, R_s)$, ξ searches the list $\mathcal{L}_2$. If a record $(m, ID_s, ID_v, U_s, R_s, l_{2i})$ exists on $\mathcal{L}_2$, ξ returns l_{2i}. otherwise, ξ picks a random $l_{2i} \in Z_q^*$ and returns l_{2i}. ξ adds $(m, ID_s, ID_v, U_s, R_s, l_{2i})$ to $\mathcal{L}_2$.

Queries on oracle H_3 $\left(H_3(m, ID_s, ID_v, U_s, R_s, l_{2i})\right)$*:* When ADV submits a query on $(m, ID_s, ID_v, U_s, R_s, l_{2i})$, ξ searches the list $\mathcal{L}_3$ and returns $(m, ID_s, ID_v, U_s, R_s, l_{2i}, l_{3i})$ if it already exists. Otherwise, ξ picks a random $l_{3i} \in Z_q^*$, and returns l_{3i} and ξ adds $(m, ID_s, ID_v, U_s, R_s, l_{2i}, l_{3i})$ to $\mathcal{L}_3$.

Key Extraction Oracle $\left((KExtID_i)\right)$*:* When ADV makes this query on identity ID_i, ξ does the following. If $ID_i = ID^*$, ξ aborts. Otherwise (if $ID_i \neq ID^*$), ξ sets $d_i = r_i$ and returns d_i to ADV.

Signing Oracle: When ξ receives a query on (ID_s, m), with a verifier ID_v, ξ first makes queries on H_1, H_2, H_3 oracles and recovers the records $(ID_i, R_i, P_{Pub}, l_{1i})$, $(m, ID_s, ID_v, U_s, R_s, l_{2i})$, $(m, ID_s, ID_v, U_s, R_s, l_{2i}, l_{3i})$ from $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ respectively. ξ generates two random numbers $r_{1i}, r_{2i} \in Z_q^*$ and sets $k_i = r_{1i}, V_i = \left(r_{1i}P - (R_i + l_{1i}P_{Pub})l_{2i}\right)l_{3i}^{-1}$, $U_i = r_{2i}X_v$ and $W_i = r_{2i}P$. ξ returns $\sigma_i = (W_i, V_i, k_i)$ to ADV. Note that $\sigma_i = (W_i, V_i, k_i)$ generated in this way satisfies the verification eq

$$(k_i P - (R_i + h_{1i}P_{Pub})h_{2i})h_{3i}^{-1} = V_i. \quad (1)$$

D. Verify Oracle $\left(DV(ID_i)\right)$**:** ADV submits (ID_s, ID_v, m) and $\sigma_i = (W_i, V_i, k_i)$ to ξ. It first recovers $(ID_v, R_v, P_{Pub}, l_{1i})$ from $\mathcal{L}_1$ list and continues as follow.

(i) If $ID_v \neq ID^*$, it computes $U_i = r_v W_i$ and then recovers the entries $l_{2i} = H_2(m, ID_s, ID_v, U_i, R_i)$ and $l_{3i} = H_3(m, ID_s, ID_v, U_i, R_i, l_{2i})$ from $\mathcal{L}_2$ & $\mathcal{L}_3$ lists. If these entries does not exists, ξ selects $l_{2i}, l_{3i} \in Z_q^*$ and defines

$H_2(m, ID_s, ID_v, U_i, R_i) = l_{2i}$ and $H_3(m, ID_s, ID_v, U_i, R_i, l_{2i}) = l_{3i}$. ξ then checks the equation (1) for the validity of $\sigma_i = (W_i, V_i, k_i)$; it returns either 1(valid) or 0 (invalid) to ADV.

(ii) If $ID_v = ID^*$, ξ works on all possible entries $H_2(m, ID_s, ID_v, U_i, R_i)$ and $H_3(m, ID_s, ID_v, U_i, R_i, l_{2i})$ for some U_i. For each possible entry $H_2(m, ID_s, ID_v, U_i, R_i) = l_{2i}$ and $H_3(m, ID_s, ID_v, U_i, R_i, l_{2i}) = l_{3i}$ for some U_i, ξ checks equation (1): and returns either 1(valid) or 0 (invalid) to ADV. If the above procedure does not lead ξ to return an answer for ADV, ξ then returns 0(invalid) to ADV.

P. Verify Oracle $\left(PV(ID_i)\right)$**:** ADV. submits (ID_s, ID_v, m) and $\sigma_i = (W_i, V_i, k_i)$ to ξ. It follows the same procedure as in the simulation of *DVerify Oracle*. The only difference is; when ξ judges $\sigma_i = (W_i, V_i, k_i)$ is valid (i.e., returns 1 in the *DVerify Oracle*); it returns $Aid = U_i = r_{2i}R_v = r_v W_i = Y_i$ to ADV. When ξ judges $\sigma_i = (W_i, V_i, k_i)$ is invalid (i.e., returns 0 in the *DVerify Oracle*); it returns $\perp$ to ADV.

- **Forgery:** Finally, ADV. outputs a forged tuple $(ID_s^*, ID_v^*, m^*, \sigma_i^*)$, where $\sigma_i^* = (W_i^*, V_i^*, k_i^*)$. If $ID_i \neq ID_s^*, \xi$ stops simulation. Otherwise, let $\sigma_i^{(1)} = (W_i^{(1)}, V_i, k_i^{(1)})$ denote $\sigma_i = (W_i, V_i, k_i)$. By Forking Lemma [V], ξ repeats simulation with same random tape but different choice of H_2, H_3, ADV will output another two $\sigma_i^{(j)} = (W_i^{(j)}, V_i, k_i^{(j)})$ for $j = 2,3$, and equation (1) holds. Hence $\left(k_i^{(j)}P - (R_i + l_{1i}^{(j)}P_{Pubi})l_{2i}^{(j)}\right)l_{3i}^{-1(j)} = V_i$ for $j = 1,2,3$.

  By r_i, s, v_i, we now denote discrete logarithms of R_i, P_{Pub}, V_i respectively, that is $R_i = r_i P, P_{Pub} = sP, V_i = v_i P$. From the above equation, we get $\left(k_i^{(j)} - (r_i + l_{1i}^{(j)}s)l_{2i}^{(j)}\right)l_{3i}^{-1(j)} = v_i$ for $j = 1,2,3$.

  In these equations, only, r_i, s, v_i are unknown to ξ. ξ solves these values from the above three linear independent equations and outputs s as the solution of DLP.

Theorem 2: If a distinguisher can break the invisibility of the proposed ID-DS scheme, then there is an algorithm which can solve the ECDL problem.

Proof:

Here we present the main idea to prove the invisibility of our ID-DS scheme by giving the ECDL problem instance $(P, A = aP, z)$. The ECDLP solver ξ simulates the distinguisher D by initializing the D with $P_{Pub} = aP = A$ as the system public key. ξ answers the oracle queries of D in the same way as in Theorem1. In the challenge phase, if ID_v^* is not the target designated verifier, ξ outputs failure and terminates the simulation. Otherwise ξ chooses $e, f \in Z_q^*$ and sets $R_s^* = P_{Pub}, W_s^* = eP_{Pub}$, $V_s^* = (1 - e^{-1}(1 + l_{1s}))P$, $k_s^* = e$ and computes $Y_s^* = W_s^* - d_v P = z(k_s^* P - P - h_{1v} P)$. Now to insert the ECDLP in to challenge signature, ξ inserts $H_2(m^*, ID_s^*, ID_v^*, Y_s^*, R_s^*) = z^{-1}$, and $H_2(m^*, ID_s^*, ID_v^*, Y_s^*, R_s^*, h_2^*) = e$ into $\mathcal{L}_2$ and $\mathcal{L}_3$ respectively and sends the signature to D.as a challenge signature. Hence $\xi's$ simulation of the signature is same as the real game as long as it does not fail. D. makes a series of queries as described in [XXIV] subject to the following conditions

(i) D. cannot make Extraction queries on ID_v^*.

(ii) D. cannot make a D. verify or a P. verify query on $(ID_s^*, ID_v^*, m^*, \sigma^*)$, and it outputs a bit b' as a guess of challenge bit b of ξ.

V. Efficiency Analysis

In this section we present the performance analysis of our ID-DS scheme. We compare our scheme with the existing relevant schemes [II, III, VI, IX, XI, XIII, XVII, XVIII, XXIV]. To evaluate the performance of the proposed scheme, we consider various cryptographic operations and their notations which are presented in Table 3. We consider the experimental results [XV, XIX, XX, XXIII] to achieve the comparable security with1024-bit RSA key, where the bilinear pairing (Tate pairing) is defined over the super singular elliptic curve $E / F_p : y^2 = x^3 + x$ with embedding

degree 2 and the 160-bit Solinas prime number $q = 2^{159} + 2^{17} + 1$ with 512-bit prime number p satisfying $p + 1 = 12qr$. The running time is calculated for different cryptographic operations in [XV, XIX, XXIII] using MIRACL [XX], a standard cryptographic library and implemented on a hardware platform PIV (Pentium-4) 3GHZ processor with 512-MB memory and a windows XP operating system. From these results, various cryptographic operations and their conversions are presented in Table 3.

Table 3: Conversions various cryptographic operations

Notations	Description
T_{MM}	Modular multiplication operation $1T_{MM} \approx 0.2325ms$
T_{SM}	Scalar multiplication over elliptic curves : $T_{SM} = 29T_{MM} \approx 6.38ms$
T_{BP}	Bilinear pairing: $T_{BP} = 87T_{MM} \approx 20.01ms$
T_{PEX}	Pairing–based exponentiation : $T_{PEX} = 43.5T_{MM} \approx 11.20ms$
T_{INV}	Modular inversion operation: $T_{INV} = 11.6T_{MM} \approx 2.697ms$
T_{MH}	Map to point hash function : $1T_{MH} = 29T_{MM} \approx 6.38ms$
T_{MX}	Modular exponentiation operation: $T_{MX} = 240T_{MM} \approx 55.20ms$
T_{PA}	Elliptic curve point addition : $T_{PA} = 0.12T_{MM} \approx 0.0279ms$

We now analyze and compare our ID-DS scheme with the existing directed signature schemes [II, III, VI, IX, XI, XIII, XVII, XVIII, XXIV] in terms of Signing cost, D. Verify cost, P. Verify cost and total computation costs. To generate a signature in Lim et al. scheme [III], signer needs to execute two scalar modular exponentiations, one modular multiplication and one hash function i.e. $2T_{MX} + 1T_{MM} + 1T_H$. Hence the run time to generate the signature is $\approx 110.63ms$. To verify the signature generated by the signer, a designated verifier in Lim et al. scheme needs to execute three modular exponentiations, two modular multiplications and one hash function $3T_{MX} + 2T_{MM} + 1T_H$. Hence the run time to Designated verification of the signature is $\approx 166.06ms$. To verify the signature generated by the signer, a public verifier in Lim et al. scheme needs to execute five modular exponentiations, four modular multiplications and one simple hash function i.e. $5T_{MX} + 4T_{MM} + 1T_H$. Hence the run time to Public verification of the signature is $\approx 276.93ms$. Hence the total run time for Lim et al.'s scheme is $553.62ms$. Similarly, in Lu et al. scheme [XVII], the run time to generate the signature is $110.4ms$ and for Designated verification is $166.06ms$, for Public verification is $110.4ms$. Hence the total run time for Lu et al.'s scheme is

386.86*ms*. In Ismail et al. scheme [VI], the run time to generate the signature is 58.12*ms* and for Designated verification is 165.83*ms*, for Public verification is 276.46*ms*. Hence the total run time for Ismail et al.'s scheme is 500.42*ms*.

In S. Lal et al. scheme [XVIII], the run time to generate the signature is 166.06*ms* and for Designated verification is 166.06*ms*, for Public verification is 166.06*ms*. Hence the total run time for Lal et al.'s scheme is 498.19*ms*. In N. N. Ramlee et al. scheme [XIII], the run time to generate the signature is 276.46*ms* and for Designated verification is 276.46*ms*, for Public verification is 2.92*ms*. Hence the total run time for Ramlee et al.'s scheme is 555.85*ms*. In X. Sun et al. scheme [XXIV], the run time to generate the signature is 51.93*ms* and for Designated verification is 86.42*ms*, for Public verification is 66.41*ms*. Hence the total run time for Sun et al.'s scheme is 204.76*ms*. In B.U.P. Rao et al. scheme [II], the run time to generate the signature is 83.12*ms* and for Designated verification is 77.61*ms*, for Public verification is 57.6*ms*. Hence the total run time for B.U.P. Rao et al.'s scheme is 218.33*ms*. In J. Zhang et al. scheme [IX], the run time to generate the signature is 125.49*ms* and for Designated verification is 155.22*ms*, for Public verification is 104*ms*. Hence the total run time for Zhang et al.'s scheme is 384.71*ms*. In Gayathri et al. scheme [XI], the run time to generate the signature is 19.17*ms* and for Designated verification is 34.70*ms*, for Public verification is 34.70*ms*. Hence the total run time for Gayathri et al.'s scheme is 88.59*ms*. In our proposed ID-DS scheme, the run time to generate the signature is 19.19*ms* and for Designated verification is 34.68*ms*, for Public verification is 28.27*ms*. Hence the total run time for the proposed scheme is 82.14*ms*. The comparison analysis of these schemes were presented in Table 4.

Table 4: Comparison of the proposed ID-DS scheme with the related schemes

Scheme	Signing Cost	D Verify cost	P Verify Cost	Total Cost	Hard Problem
Lim et al. [III]	$2T_{MX}+1T_{MM}+1T_H \approx 110.63ms$	$3T_{MX}+2T_{MM}+1T_H \approx 166.06ms$	$5T_{MX}+4T_{MM}+1T_H \approx 276.93ms$	553.62*ms*	DLP
Lu et a [XVII]	$2T_{MX}+1T_H \approx 110.4ms$	$3T_{MX}+2T_{MM}+1T_H \approx 166.06ms$	$2T_{MX}+1T_H \approx 110.4ms$	386.86*ms*	IFP
Ismail et al. [VI]	$1T_{MX}+1T_{MM}+2T_H+1T_{INV} \approx 58.12ms$	$3T_{MX}+2T_{MM}+1T_H \approx 166.06ms$	$5T_{MX}+2T_{MM}+1T_H \approx 276.46ms$	500.42*ms*	DLP
S. Lal et a [XVIII]	$3T_{MX}+2T_{MM}+1T_H \approx 166.06ms$	$3T_{MX}+2T_{MM}+1T_H \approx 166.06ms$	$3T_{MX}+2T_{MM}+1T_H \approx 166.06ms$	498.19*ms*	IFP
N.N. Ramlee et al. [XII]	$5T_{MX}+2T_{MM}+1T_H \approx 276.46ms$	$5T_{MX}+2T_{MM} \approx 276.46ms$	$1T_{MM}+1T_H+1T_{INV} \approx 2.929ms$	555.85*ms*	IFP&DLP
X. Sun et a [XXIV]	$3T_{SM}+1T_{BP}+2T_{MTPH}+1T_{PA} \approx 51.93ms$	$4T_{BP}+1T_{MTPH} \approx 86.42ms$	$3T_{BP}+1T_{MTPH} \approx 66.41ms$	204.76*ms*	CDHP& DBDHP
B.U.P. et al. [II]	$3T_{SM}+2T_{BP}+2T_{MTPH}+1T_H+1T_{PEX} \approx 83.12m$	$3T_{BP}+1T_{MTPH}+1T_H+1T_{PEX} \approx 77.61ms$	$2T_{BP}+1T_{MTPH}+1T_H+1T_{PEX} \approx 57.6ms$	218.33*ms*	CDHP& DBDHP
J.Zhang et al. [IX]	$6T_{SM}+1T_{BP}+2T_H+1T_{XOR}+6T_{PEX} \approx 125.49n$	$2T_{SM}+6T_{BP}+2T_H+1T_{XOR}+2T_{PEX} \approx 155.22ms$	$2T_{SM}+4T_{BP}+1T_H+1T_{PEX} \approx 104ms$	384.71*ms*	CDHP& DBDHP
Gayathri et al. [XI]	$3T_{SM}+1T_{PA}+2T_H \approx 19.17ms$	$5T_{SM}+2T_H+4T_{PA}+1T_{INV} \approx 34.71ms$	$5T_{SM}+2T_H+4T_{PA}+1T_{INV} \approx 34.71ms$	88.59*ms*	ECDLP
Our Scheme	$3T_{SM}+2T_H+2T_{PA} \approx 19.19ms$	$5T_{SM}+2T_H+3T_{PA}+1T_{INV} \approx 34.68ms$	$4T_{SM}+2T_H+2T_{PA}+1T_{INV} \approx 28.27ms$	82.14*ms*	ECDLP

The total computation costs of these schemes are represented through the following bar graph. Clearly Figure 1 indicates that the proposed scheme is more efficient than existing schemes. The total computation cost of our ID-DS scheme is 82.14*ms*, and is 85.16% less than Lim et al. scheme, 78.76% less than Lu et al. scheme, 83.58% less than Ismail et al. scheme, 83.51% less than Lal et al. scheme, 85.22% less than Ramlee et al. scheme, 59.88% less than sun et al. scheme, 62.37% less than B.U.P et al. scheme, 78.64% less than Zhang et al. scheme and 7.28% less than Gayathri et al. scheme.

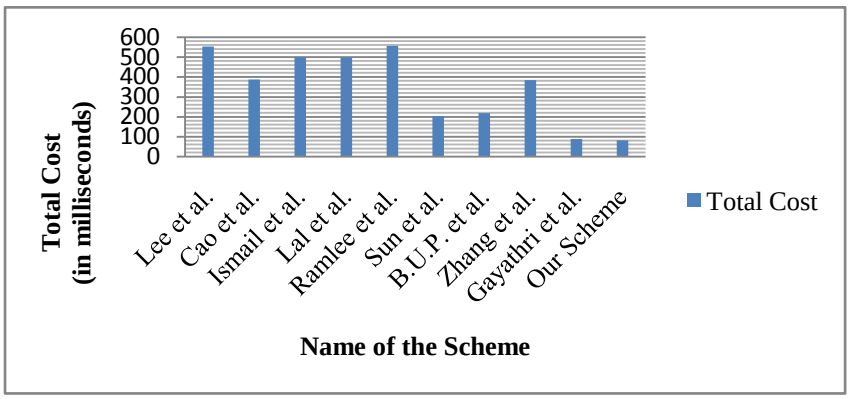

Fig. 1: Graphical representation of total computation cost.

VI. Conclusion

When the signed message is sensitive to signature receiver, the directed signature place an important role in many applications such as signatures on medical records, tax information etc. However all the existing directed signature schemes in ID based setting uses bilinear pairings over elliptic curves. Due to the heavy computational cost of pairing operations, these existing ID based directed signature schemes are not much efficient in practice. In order to improve the efficiency, in this paper, we presented an efficient Identity-based directed signature scheme without pairings and is proven secure in the random oracle model with the assumption that the ECDLP is intractable. The performance analysis shows that the proposed directed signature scheme without pairings in ID based setting is more efficient than the existing directed signature schemes.

References

I. A. Shamir; “Identity-based Cryptosystems and Signature Schemes”, Advances in Cryptology, Crypto-84, Lecture Notes in Computer Science, Springer, vol. 196, pp.47-53, 1984

II. B. Uma Prasada Rao; P. Vasudeva Reddy; T. Gowri; "An efficient ID-Based Directed Signature Scheme from Bilinear Pairings", Available at https://eprint.iacr.org/2009/617.pdf.

III. C. H. Lim; P. J. Lee; "Directed Signatures and Applications to Threshold Cryptosystem", Workshop on Security Protocol, Cambridge, pp. 131-138, 1996

IV. C. P. Schnorr; "Efficient Identification and Signatures for Smart Cards", Advances in Cryptology-Crypto'89, Lecture Notes in Computer Science, Springer, vol. 435, pp. 239-252, 1989

V. D. Pointcheval; J. Stern; "Security Arguments for Digital Signatures and Blind Signatures", Journal of Cryptology, vol. 13, No.3, pp.361-369, 2000

VI. E. S. Ismail; Y. Abu- Hassan; "A Directed Signature Scheme Based on Discrete Logarithm Problems", Jurnal Teknologi, vol. 47(C), pp. 37-44, 2007

VII. F. Laguillaumie; P. Paillier; D. Vergnaud; "Universally Convertible Directed Signatures", Advances in Cryptology - ASIACRYPT'05, Lecture Notes in Computer Science, Springer, vol. 3788, pp. 682–701, 2005

VIII. J. Ku; D. Yun; B. Zheng; S. Wei; "An Efficient ID-Based Directed Signature Scheme from Optimal Eta Pairing", Computational Intelligence and Intelligent Systems, vol. 316, pp. 440-448, 2012

IX. J. Zhang; Y. Yang; X. Niu; "Efficient Provable Secure ID-Based Directed Signature Scheme without Random Oracle", 6th International Symposium on Neural Networks: Advances in Neural Networks-ISNN 2009, Lecture Notes in Computer Science, Springer, vol. 5553, pp.318-327, 2009

X. L. C. Guillou; J. J. Quisquater; "A "Paradoxical" Indentity-Based Signature Scheme Resulting from Zero-Knowledge", Advances in Cryptology-Crypto'88, Lecture Notes in Computer Science, Springer, vol. 403, pp. 216-231, 1988

XI. N. B. Gayathri; T. Gowri; R. R. V. Krishna Rao; P. Vasudeva Reddy; "Efficient and Secure Pairing-free Certificateless Directed Signature Scheme", Journal of King Saud University- Computer and Information Sciences, Article in press, 2018

XII. N. Koblitz; "Elliptic Curve Cryptosystems", Mathematics of Computation, vol. 48, no. 177, pp. 203-209, 1987

XIII. N. N. Ramlee; E. S. Ismail; "A New Directed Signature Scheme with Hybrid Problems", Applied Mathematical Sciences, vol. 7, No. 125, pp. 6217-6225, 2013

XIV. N. Tiwari; S. Padhye; "Provable Secure Multi-proxy Signature Scheme without Bilinear Maps", International Journal of Network Security, vol.17, no.6, pp.736-742, 2015

XV. P.S.L.M. Barreto; B. Libert; N. McCullagh; J.J. Quisquater; "Efficient and Provably Secure Identity-based Signatures and Signcryption from Bilinear Maps", Advances in Cryptology-ASIACRYPT'05, Lecture Notes in Computer Science, Springer, vol. 3788, pp. 515-532, 2005

XVI. Q. Wei; J. He; H. Shao; "Directed Signature Scheme and its Application to Group Key Initial Distribution", 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS-2009), ACM, 2009, pp. 24-26, 2009

XVII. R. Lu; Z. Cao; "A Directed Signature Scheme Based on RSA Assumption", International Journal of Network Security, vol. 2, No. 3, pp.182–421, 2006

XVIII. S. Lal; M. Kumar; "A Directed Signature Scheme and its Applications", 2004. Available at http://arxiv.org/abs/cs/0409035.

XIX. S. Y. Tan; S. H. Heng; B. M. Goi; "Java Implementation for Pairing-Based Cryptosystems", Computational Science and Its Applications (ICCSA'10), Lecture Notes in Computer Science, Springer, vol. 6019, pp. 188-198, 2010

XX. Shamus Software Ltd. Miracl Library. Available: http://certivox.org/display /EXT/MIRACL.

XXI. V. Miller; "Uses of Elliptic Curves in Cryptography", Advances in Cryptology-Crypto 85, pp. 417-426, 1985

XXII. W. Diffie; M.E. Hellman; "New Directions in Cryptography", IEEE Transactions in Information Theory, vol. 22, pp.644-654, 1976

XXIII. X. Cao; W. Kou; X. Du; "A Pairing-free Identity-based Authenticated Key Agreement Protocol with Minimal Message Exchanges", Information Sciences, vol. 180, No. 15, pp. 2895-2903, 2010

XXIV. X. Sun; J. Li; G. Chen; S. Yung; "Identity-Based Directed Signature Scheme from Bilinear Pairings", Available at https:// eprint.iacr.org/2008/305.pdf.

XXV. Y. Wang; "Directed Signature Based on Identity", Journal of Yulin College, vol. 15, No. 5, pp. 1–3, 2005.