International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8964-8971


# An Efficient Key Escrow-Free Identity-Based Signature Scheme

**Subhas Chandra Sahana**
*Assistant Professor, Department of Information Technology*
*North Eastern Hill University, Shillong, Meghalaya, India.*
*Orcid Id: 0000-0003-2018-8322*

**Bubu Bhuyan**
*Associate Professor, Department of Information Technology*
*North Eastern Hill University, Shillong, Meghalaya, India.*

**Manik Lal Das**
*Associate Professor, Dhirubhai Ambani Institute of Information and*
*Communication Technology (DA-IICT), Gandhinagar- 382007, Gujarat, India.*

**Abstract**

There are mainly two drawbacks of identity-based cryptosystem. First one is that it suffers from key escrow problem and the second one is that it uses a secure channel at the stage of private key issuance by the Private Key Generator (PKG). In this paper, we propose a key escrow-free identity-based signature scheme without using secure channel in the process of private key issuance stage. The bilinear pairing is used for the construction of the proposed scheme. The scheme is secure against adaptive chosen message attack and given *ID* attack under the assumption that the computation Diffie-Hellman problem is hard.

**Keywords:** Bilinear pairings, Digital signature scheme, Identity-Based Cryptosystem, Key escrow, Computational Diffie-Hellman problem, Adaptive chosen Message Attack.

## INTRODUCTION

In a traditional Public Key Cryptosystem (PKC) [1], one of the biggest advantages is that anyone can verify the authenticity of the public key using the issued certificate from a trusted certificate authority. But, this advantage comes with a lot of involved certificate management activities and computational cost in practice. To avoid this problem, Shamir [2] introduced the concept of identity-based cryptosystem (IBC) in 1984. In IBC, the main advantage is that there is no need of public key distribution in the form of certificates, as user can use his unique identity information such as name, email address etc. to provide his own public key. Thus, the user does not need any extra computation cost to verify the validity of other signer's public keys or any extra storage to store the certificate of other signer's public keys, since he/she only needs to know other signers' identities. However, the IBC has an inherent severe security problem, called key escrow problem as a malicious Private Key Generator (PKG) can decrypt a message or forge a signature on a message using user's private key which is generated only by the PKG involved in the cryptosystem.

After the pioneer work [2] by Shamir, several identity-based signature schemes [3-4] have been proposed. But till 2001, no practical identity-based scheme was proposed. In 2001, Boneh and Franklin [5] proposed a practical identity-based encryption scheme. This implementation was done using bilinear pairings. After that many identity-based signature schemes [6] were presented. Most of them [5] constructed using bilinear pairings. But, all identity-based signature schemes in [6] suffer from inherent key escrow problem. It is always an open problem for finding a solution to eliminate the key escrow. Various efforts [7-10] have been made to get a solution for solving the key escrow problem. Besides the key escrow problem, when PKG issues the private key to the user, a secure channel is used for the transmission of the private key. Das ***et al.*** [11] proposed a solution using binding-blinding technique for to eliminate the key escrow problem and avoid the secure channel requirements.

The rest of the paper is organized as follows. In section II, the mathematical back ground, required for our proposed scheme, is presented. After that, in section III, a variant identity-based signature scheme has been reviewed. The formal security model for the proposed scheme has been given in section IV. In section V, the proposed scheme has been presented. Following the proposed scheme, proof of security of the proposed scheme has been given in section VI. The efficiency of the proposed scheme with similar established scheme has been discussed in section VII. At last, we conclude our work in section VIII.

## MATHEMATICAL BACKGROUND

In this section, the basic mathematical concepts, required for implementing the scheme, presented in this paper have been discussed.

### *Bilinear Pairings*

A bilinear pairing [12, 13] is a mapping function which takes elements from two groups as input and produce an element as output of a multiplicative abelian group. The two famous example of bilinear pairing is Weil pairing and Tate pairing. At first, the bilinear pairing was not able to draw significant attention. But when, Menezes ***et al.*** [12] used the Weil pairing to attack discrete logarithm based system on a certain class of elliptic curves, ( MOV reduction Problem) after that, the researchers found that it has potential in constructing different primitives in the field of cryptography.

Let $G_1$ and $G_2$ be two cyclic groups of the same prime order $q$. We will view $G_1$ as additive group and $G_2$ as a multiplicative group. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties.

- *Bilinearity:* For all $a, b \in Z_q$ and $P, Q \in G_1$and the map $e : G_1 \times G_1 \rightarrow G_2$ satisfie $e(aP, bQ) = e(P, Q)^{ab}$.
- *Non-degeneracy:* There are $P, Q \in G_1$ such that $e(P, Q) \neq 1$.
- *Computability:* There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

### *Different Difie-Hellman Problems*

Actually, the security of an asymmetric key cryptographic scheme basically depends on the assumption on the hardness of solving a computational problem. It has been assumed that there is a randomized parameter generation algorithm. In each problem, we consider a polynomial time algorithm which accepts security parameter $1^K$ and outputs the required result.

**Discrete Logarithm Problem (DLP).** Suppose, $p$ and $q$ be two prime numbers such that $q$ divides $p - 1$. Let $g$ be a random element with order $q$ in $Z_p^*$. Suppose $y$ be a random element generated by $g$. For any probabilistic polynomial time algorithm $A$, if $g^x = y \, mod \, p$, then the probability $Pr[A(p, q, g, y) = x]$ is ''negligible in $k$.''

**Elliptic Curve Discrete Logarithm Problem (ECDLP).** Given an elliptic curve $E(F_q)$, points $P$ and $Q(= xP)$ for $P, Q \in E(F_q)$, the ECDLP is similarly formulated when trying to determine the integer $x$.

**Decisional Diffie-Hellman Problem (DDHP):** Given $(P, aP, bP, cP)$ for all $a; b; c \in Z_q$, determine whether $c = ab \, mod \, q$. The advantage of any probabilistic polynomial-time algorithm $A$ in solving DDHP in $G_1$ is defined as $Adv_{A,G_1}^{DDH} = [Pr[A(P, aP, bP, cP) = 1] - Pr[P(P, aP, bP, abP) = 1]$ for all $a; b; c \in Z_q$. Here the algorithm $A$ will have four parameters (elliptic curve points) $P, aP, bP,$ and $cP$ in $G_1$ and will have to answer 1 or 0 (i.e., true or false) whether $c \equiv ab \, mod \, q$ for all $a; b; c \in Z_q$.

**Computational Diffie-Hellman Problem (CDHP):** Given $(P, aP, bP)$ for all $a; b \in Z_q$, compute $abP$. The advantage of any probabilistic polynomial-time algorithm $A$ in solving CDHP in $G_1$, is defined as $Adv_{A,G_1}^{CDHP} = Pr[A(P, aP, bP, abP) = 1$ for all $a; b \in Z_q$. It is noted that solving CDHP in $G_1$ is a hard problem, and the above advantage is negligible in $k$ for all probabilistic polynomial time algorithms.

**Gap Diffie-Hellman (GDH) group:** The group $G_1$ is a GDH group if there exists an efficient polynomial-time algorithm which solves the DDHP in $G_1$ and there is no probabilistic polynomial-time algorithm which solves the CDHP with non-negligible probability of success. The domains of bilinear pairings provide examples of GDH groups. The MOV reduction [9] provides a method to solve DDHP in $G_1$, whereas there is no known efficient algorithm to solve CDHP in $G_1$.

**Bilinear Diffie-Hellman Problem (BDHP):** Given $(P, aP, bP, cP)$ for all $a; b; c \in Z_q$, compute $e(P, P)^{abc}$. The advantage of any probabilistic polynomial-time algorithm $A$ in solving BDHP in $(G_1, G_{2,}e)$ is defined as $Adv_{A,G_1}^{BDH} =$ [Pr[A(P; aP; bP; cP) = e(P, P)$^{abc}$ ] for all $a; b; c \in Z_q$ and the advantage is negligible in $k$ for all probabilistic polynomial time algorithms.

## A VARIANT OF IDENTITY-BASED SIGNATURE SCHEME (VIDS)

The participating entities and their roles in the VIDS scheme are defined as follows:

- Private Key Generator (PKG): A trusted authority who receives the user's identity ($ID$) along with other parameters, checks validity of $ID$ and issues a partial key to the user corresponding to his=her $ID$.
- Signer: An entity who signs the message.
- Verifier: An entity who verifies the signed string and decides whether the signature is valid or not.

### *Binding-blinding Technique*

A binding-blinding technique is used to eliminate the secure channel requirement and to eliminate the key escrow problem. The idea of binding-blinding technique [11] was introduced by Das ***et al.*** They extended the original identity-based signature scheme of Hess ***et al***.[ 15] into a key escrow-free identity-based signature scheme without using secure channel same technique using binding-blinding technique. We use the same binding-blinding techniques as in [11], which is a two-step method as explained below.

- **Step1:** In this step, two secret blinding factors were chosen by the user. After that, the user calculates the binding parameters and sends these parameters to the PKG over a public channel along with the user's $ID$. As the communication medium between user and PKG is a public channel, an unregistered identity attack is a potential threat. A dishonest party can construct his preferred binding parameters with respect to an unregistered user's $ID$ and sends the binding parameters to the PKG requesting a partial key corresponding to that $ID$. If there is no secure channel between PKG and user avoiding a unregistered $ID$ attack for any type of $ID$ is a challenging task. To avoid the unregistered user's $ID$ attack, we assume that the email-id acts as the user $ID$; however, other $ID$ could play the same role. Upon receiving the binding parameters, the PKG first send a message to the email-id and asks for a confirmation from the email-id owner. If the email-id owner confirms user request for the partial key to be used in construction of private key of the user, then the PKG proceeds to the next step.
- **Step2:** The PKG checks the validity of the binding parameters. Upon successful validation of the parameters, the PKG computes the user's partial key and sends it to the user over a public channel.

***The VIDS Model***

The VIDS model consists of the following algorithms.

- $\boldsymbol{sysPam}()$***:*** A randomized system parameter generation algorithm that generates all system parameters, denoted by $params$, which include $G_1$ (an additive cyclic group of prime order $q$), $G_2$ (a multiplicative cyclic group of same order), a generator $P$ of $G_1$, a bilinear pairing $e$: $G_1 \times G_1 \rightarrow G_2$, map-to-point $H$:$\{0,1\}^* \rightarrow G_1$, and a cryptographic hash function $H_1$:$\{0,1\}^* \rightarrow Z_q$ and $H_2$:$G_1 \rightarrow Z_q$. The $params$ are made available to all interested parties.
- $\boldsymbol{sysKey}(\boldsymbol{k})$: A randomized system key generation algorithm that takes as input a security parameter $k$ and generates PKG's private key $sk_{KGC}$ and public key $pk_{KGC}$. The $sysKey(k)$ selects a master secret $s \in Z_q$ as the PKG'sprivate key ($sk_{KGC}$), and publishes $sP$ as the PKG's public key $pk_{KGC}$.
- $\boldsymbol{bindPar}(\boldsymbol{params}, \boldsymbol{ID})$: A deterministic interactive algorithm in which a user requests a partial key from the PKG. The user chooses two secret blinding factors in $Z_q$, computes four parameters using secret blinding factors, and sends these parameters to the PKG over a public channel along with the user's $ID$.
- $\boldsymbol{parKey}(\boldsymbol{params}, \boldsymbol{bindPar}(\boldsymbol{params}, \boldsymbol{ID}), \boldsymbol{ID})$:

  A deterministic partial key generation algorithm, which takes system parameters, binding parameters, and an $ID$ as input and outputs a partial key, $D_{ID}$, and a registration-token $Reg_{ID}$ corresponding to the $ID$.
- $\boldsymbol{pvtKeyGenblinding\ factor}, D_{ID})$:A deterministic private key generation algorithm that uses a secret blinding factor from the $bindPar(params, ID)$ algorithm and the user's partial key $D_{ID}$ to generate the user's private key $sk_{ID}$, where the user's public key $pk_{ID}$ is associated withthe user $ID$.
- ***sign(params, $sk_{ID}$, m):*** A randomized signature generation algorithm that uses the signer's private key $sk_{ID}$ and a message m as input and outputs the signature $\sigma_{ID}$ on $m$.
- ***verify(params, ID, $Reg_{ID}$, m, $\sigma_{ID}$):*** A deterministic signature verification algorithm that uses the signer's $ID$, **params**, registration-token, message m, and its signature $\sigma_{ID}$ as input and outputs true if $\sigma_{ID}$ is valid. Otherwise, it returns false.

**SECURITY MODEL**

In 2010, M. L. Das [14] proposed a key escrow-free identity-based signature scheme without using Secure Channel fitted in a model called a variant identity-based signature (VIDS) model, from the blinding binding techniques. However, they presented neither a formal security model nor adversary types of their proposed model. In this section a formal security model of the VIDS model has been proposed under random oracle.

***Adversaries***

We can define two types of adversaries (type-I and type- II) of this type of VIDS model, just like certificate-less cryptosystem [9]. A type-II adversary in that cryptosystem is the malicious PKG who is armed with the master secret key but does not have the capabilities of public key replacement of the original signer and type-I adversary, a dishonest user, can replace the public key of the original signer but does not have the access to the master secret key. As in our proposed VIDS scheme, there is no facility to replace the public key of the original signer so we can treat the type-I adversary just as an dishonest user who has neither the facility to access the master secret key nor the facility replace the public key. So, according to our proposed VIDS scheme type-II adversary has more attacking power than the type I adversary. So, type-II adversary can overlap the attacking power of type-I in the sense, if type-II adversary can forge the signature of a message it is always possible to forge the signature by the type-I adversary.

So, from the above discussion, it is sufficient to define only one adversary who has the facility to get the master secret key. We denote the adversary as $A$.

***Random oracles***

Generally there are seven oracles which can be accessed by the adversary. The brief description of those oracles is given below.

1. ***Create user and extract public key:***

   On input an identity $ID \in \{0,1\}^*$, if $ID$ has already been created, nothing is to be carried out. Otherwise, the oracle runs the other algorithms which are the components of the scheme to set the public key, blinding factors, binding parameters, registration $ID$ partial private key and private key. In this case, $ID$ is said to be created. In both cases, the public key $pk_{ID}$ is returned.

2. ***Extract binding parameter:***

   The input to this oracle is the user $ID$. If the $ID$ is already created it returns the corresponding binding parameters else it returns $null$.

3. ***Extract blinding factors:***

   On input user $ID$, it returns the corresponding blinding factors if the $ID$ is already created otherwise it returns $null$.

4. ***Extract partial private key:***

   The input to this oracle is the user $ID$. If the $ID$ is already created it returns the corresponding partial private key else it returns $null$.

5. ***Extract registration ID:***

   On input user $ID$, it returns the corresponding registration $ID$ if the $ID$ is already created otherwise it returns $null$.

6. ***Extarct private key:***

   The input to this oracle is the user $ID$. If the $ID$ is already created it returns the corresponding private key else it returns $null$.

7. ***Sign:***

   On input user $ID$ and a message $m \in \{0,1\}^*$, the oracle proceeds in one of the following cases below:

   - If the $ID$ has not been already created, returns $null$.
   - If the $ID$ has been created already then returns a valid signature $\sigma_{ID}$ such that $true \leftarrow verify(m, \sigma_{ID}, ID, pk_{ID}, params)$

The standard notion of security for a signature scheme is called existential unforgeability against adaptive chosen message attack defined by Goldwasser, Micali and Revist [17]. To define the existential unforgeability of the proposed VIDS scheme against the adversary, we define a game as follows.

***The proposed Game:***

The game is executed between a challenger $F$ and an adaptive chosen message and chosen identity adversary $A$.

- **Setup.** The challenger $F$ runs the algorithm Setup of the signature scheme to obtain both the public parameter **params** and the master secret key ***msk***. The adversary $A$ is given **params** but the master secret key ***msk*** is kept by the challenger.
- **Queries.** $A$ adaptively access all the oracles defined in Section IV in a polynomial number of times.
- **Forgery.** Eventually, $A$ outputs a forgery $(ID^*, pk, m^*, \sigma^*)$ and wins the game if the following conditions hold true:
  1. $true \leftarrow verify(params, m^*, \sigma_{ID^*}, ID^*, pk_{ID^*})$
  2. $(ID^*, m^*)$ has never been submitted to the oracle Sign.
  3. $ID^*$ has never been submitted to the oracle *Extract-Private-Key*.

**Definition 1.** Define $Adv_A$ to be the probability that an adaptively chosen message and chosen identity adversary $A$ wins in the above game, taken over the coin tosses made by $A$ and the challenger $F$. We say a VIDS scheme is secure against the mentioned adversary $A$, if for all probabilistic polynomial-time (PPT) adversary $A$, the success probability $Adv_A$ is negligible.

## THE PROPOSED VIDS SCHEME

The $syspar$ and $sysKey$ algorithms in the VIDS scheme are same as mentioned above. The remaining algorithms are explained as follows.

$\boldsymbol{bindPar(params, ID)}$: Given system parameters, secret binding factors, and $ID$, according to the binding-blinding technique explained in previous section, this algorithm calculates binding parameters $X, Y, Z,$ and $W$ as follows.).

- The user computes his/her public key $pk_{ID} = H(ID)$, where $ID$ is the user's identity.
- The user chooses two secret blinding factors $a, b \in Z_q^*$ and computes $X = a.pk_{ID}, Y = a.b.pk_{ID}, Z = b.P$, and $W = a.b.P$. Then signer sends the binding parameters $(X, Y, Z, W, ID)$ to the PKG over a public channel.
- The PKG checks whether the requested $ID$ already exists in its directory. If not, the PKG asks the $ID$ owner for an email confirmation of his/her partial key request.

$\boldsymbol{parKey(params, bindPar(params, ID), ID)}$: Given system parameters, binding parameters, and $ID$, this algorithm generates a partial key $D_{ID}$ by the following steps.

- Once the $ID$ owner confirms its request, the PKG computes $pk_{ID} = H(ID)$ and verifies the validity of the user by whether $e(Y,P) = e(x,z) = e(pk_{ID}, W)$.
- If the above step holds, the PKG computes the user's partial key $D_{ID} = s.Y$ and creates a registration-token $Reg_{ID} = s.Z$. Then the PKG publishes $< Reg_{ID}, ID >$ in a public directory and sends $D_{ID}$ to the user over a public channel. The $D_{ID}$ is the partial private key corresponding to the $ID$.

We note that the PKG controls the public directory and checks every partial key request before issuing it. If the requester's $ID$ is present in the directory, the PKG denies the request.

$\boldsymbol{pvtKeyGen(a, D_{ID})}$: This algorithm inputs the user's blinding factor a used in the $bindPar(params, ID)$ algorithm. Upon receiving the partial private key $D_{ID}$, the user checks whether $e(D_{ID}, P) = e(Y, pk_{KGC})$. If this check holds then user unbinds it and generates his/her private key $sk_{ID}$ as $sk_{ID} = a^{-1}D_{ID} = b.s.pk_{ID}$

***Signature Generation***

To sign a message $m$, the signature algorithm, sign $(params, sk_{ID}, m)$, performs the following operations:

- Pick a random $k \in Z_q^*$ .
- Compute $= k.P$ .
- Compute $\sigma_{ID} = k^{-1}(H_1(m).P + H_2(R).sk_{ID})$ The signature on message $m$ is the tuple $(R, S)$.

***Signature Verification***

The verification algorithm,
$verify\ (params, ID, Reg_{ID}, m, \sigma_{ID})$, performs the following operations:

- Accept the signature if and only if $e(R, \sigma_{ID}) = e(P,P)^{H_1(m)}.e(Reg_{ID}, pk_{ID})^{H_2(R)}$ hold.

***Correctness***

$$e(R, \sigma_{ID}) = e(k.P, k^{-1}(H_1(m).P + H_2(R).sk_{ID})$$
$$= e(P,P)^{H_1(m)}.e(P, b.s.pk_{ID})^{H_2(R)}$$
$$= e(P,P)^{H_1(m)}.e(b.s.P, pk_{ID})^{H_2(R)}$$
$$= e(P,P)^{H_1(m)}.e(Reg_{ID}, pk_{ID})^{H_2(R)}$$

## SECURITY PROOFS

There is no security proof given in the Paterson [16] identity-based signature scheme. We have given the security proof of the proposed scheme under some taken assumption.

***Assumption I:*** The adversary $A$ can make request signature query on any $ID$ and any message $m$ except the target identity $ID_t$(target $ID$)

***Assumption II:*** The adversary $A$ will provide the random value $k \in Z_q^*$ chosen by him/her in the constructed forged signature as the proposed signature generation algorithm is probabilistic.

**Theorem 1.** If there exists an adaptively chosen message and chosen $ID$ adversary $A(q_c, q_{ppk}, q_{prk}, q_{reg}, q_{bf}, q_{bp}, q_s q_H, q_{H_1}, q_{H_2}, \varepsilon, t)$ who can ask at most $q_c$ Create-User queries, $q_{ppk}$ Partial-Private-Key-Extract queries, $q_{prk}$ private-key-Extract queries, $q_{reg}$ registration-id queries, $q_{bf}$ blinding factor queries, $q_{bp}$ blinding parameters and $q_s$ sign queries, respectively, and can break the proposed scheme in polynomial time $t$ with success probability $\varepsilon$, then there exists an algorithm $F$ which, using $A$ as a black box, can solve the modified Computational Difie-Hellman problem(CDHP) ( under the assumption the $A$ will provide the random value $k$ chosen by the $A$) with probability $\varepsilon' =\geq (1 - \frac{1}{q_s+1})^{qs}.\varepsilon.1/q_h$ and time $t' = t + t_{SML} + q_c + q_{ppk} + q_{prk} + q_{reg} + q_{bf} + q_{bp} + q_H + q_{H_1} + q_{H_2})$ where $t_{SML}$ is the time for run the simulator

**Proof:**

***Setup*** In order to solve the problem, $F$ utilizes the adversary $A$ as a black-box. $F$ will simulate the adversarial environments of the proposed scheme and the oracles which $A$ can access. In this proof, we regard the hash functions $H_1, H_2$ as random oracles. $F$ starts by picking an admissible bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$, and sets the master public key $P_{pub} = sP$. According to our game discussed above, the value of master secret key $s$ is known to the adversary. The ***params*** $= (G_1, G_2, e, P, P_{pub}, H, H_1, H_2)$ is sent to the adversary.

It is to be noted that the adversary $A$ is well-behaved in the sense that it request hash query $H_1$ and $H_2$ on the message $m_i$ and $R_i$ respectively before the sign query as the outputs of those hash function used in creating the signature.

***Training Phase*:** During this phase the adversary $A$ has access to the following oracles:

$F$ maintains a list $L_1$ of tuples for all the quesries except hash function queries to track the responses which are given to the adversary $A$. A list $L_2$ is maintained for the hash function queries. When a reponse/value is returned to the adversary $A$ for a $ID_i$ that response/value is inserted in the tuple corresponding to the $ID_i$. The lists ($L_1$ and $L_2$) consisting of tuples will be beneficial for constructing a signature to be given to the adversary $A$ for a particular $ID$, when a signature request comes from a particular $ID$.

1. **Create-User:** $A$ can query this oracle by given an identity $ID_i$. When the adversary makes a Create-User query with the user identity $ID_i$, $F$ picks a random bit $S_i \leftarrow \beta_\gamma$. We define $\beta_\gamma$ which will be optimized later is the probability distribution over $\{0,1\}$, where 1 is drawn with probability $\gamma$, and 0 with probability $(1-\gamma)$. The value of $S_i$ is

inserted into the corresponding tuple of the list $L_1$.We also assign the

$$Prob[ID_i = ID_t] = \beta_\gamma$$

F treat the query as follows:

a) If $i \neq t$, $F$ chooses a random number $l_i \in Z_q^*$. After that, set the value of $H(ID_i) = l.P$ and return the value as public key, $pk_{ID}$.

b) If $i = t$, $F$ set the value of $H(ID_i) = xP$ and return the value as public key $pk_{ID}$.

In both the cases, it is said that the user corresponding to the $ID_i$ is created.

2. **Extract-binding-parameter:** At any time, $A$ can query the oracle by given an identity $ID_i$. $F$ outputs a symbol $null$ if $ID_i$ has not been created.

   a) If $ID_i$ has been created and $i \neq t$, then $F$ returns four binding parameters $X_i = a_i.pk_{ID_i}, Y_i = a_i.b_i pk_{ID_i}$, $Z_i = b_i P$ and $W_i = a_i.b_i pk_{ID_i} P$ , where $a_i.b_i \in_R Z_q^*$.

   b) If $ID_i$ has been created and $i = t$, then $F$ returns four binding parameters $X_i = a_i.pk_{ID_i} = a_i xP$, $Z_i = yP$ and $W_i = a_i.yP$ , where $a_i \in_R Z_q^*$. Here it is to be noted that the value of $Y$ is not returned to the adversary $A$, because returning the value $Y$ is itself a CDHP problem.

3. **Extract-blinding-factors:** On input user $ID_i$, it checks whether the $ID_i$ is already created or not. If the $ID_i$ is not created, it returns *null* otherwise it does the followings:

   a) If $i \neq t$, then $F$ returns corresponding blinding factors $a_i.b_i$,.

   b) If $ID_i = ID_t$ then abort the simulation.

4. **Extract-partial-private-key:** The input to this oracle is the user $ID_i$. If the $ID_i$ has not been already created it returns $null$, otherwise it reacts the following way:

   a) If $i \neq t$, then $F$ returns corresponding blinding factors $a_i.b_i$,.

   b) If $i = t$, nothing is returns as itself a solution to the CDHP problem but it does not abort the simulation.

5. **Extract-registration-ID:** The input to this oracle is the user id $ID_i$. The oracle reacts as follows:

   a) If $i \neq t$ and received $ID_i$ created, then the corresponding registration-id $Reg_{ID_i} = a_i b_i P$ is returned.

   b) If $i = t$ and received $ID_i$ created, then it sets the corresponding registration-id $Reg_{ID_i} = a_i yP$ and returned.

   c) If received $ID_i$ has not been created then it returns $null$.

6. ***$H_1$* Query:** On input any message $m$, this oracle simply chooses a random number $h_{1_i} \in Z_q^*$ and return the value $H(m_i) = h_{1_i}$ to the adversary.

7. ***$H_2$* Query:** The input to this oracle is an element $R = k_i P \in G_1$, where $k$ is the response of the $H_1$ query. It sets the value $h_{2_i} = H_2(R_i) = r_i P$ where $r_i$ choosen randomly from the set $Z_q^*$.

8. **Sign***:* On input user$\{ID_i, m_i\}$ it constructs the corresponding signature $\{\sigma_{ID_i}, R_i\}$ using the values kept in the tuples for the user $ID_i$, maintained in list ($L_1$ and $L_2$). It creates the signature for the user $ID_i$ as follows.

   a) If $i \neq t$, then $F$ returns corresponding $\{\sigma_{ID_i}, R_i\}$ where $R_i = k_i P$ and $\sigma_{ID_i} = k^{-1}((P,P)^{h_{1_i}} + s b_i l_i h_{2_i} P)$, where $h_{1_i}$and $h_{2_i}$ are taken from list $L_2$.

   b) If $ID_i = ID_t$ then abort the simulation.

**Forgery:** After all the queries, $A$ eventually outputs a forgery $\{ID^*, m^*, R^*, \sigma_{ID_*}\}$.

If $ID^* \neq ID_t$, then abort the simulation, otherwise $F$ extract $h_{1_{i*}}$and $h_{2_{i*}}$ are taken from list $L_2$ and $Reg_{ID^*}$ and $pk_{ID^*}$ from list $L_1$. $F$ checks the equation $e(R^*, \sigma_{ID_*}) = e(P,P)^{H_1(m)}.e(Reg_{ID^*}, pk_{ID^*})$ holds or not. If the equation does not hold, abort the simulation. If the equation holds then $\sigma_{ID^*} = {k_i}^{-1}(h_{1_i}.P + h_{2_i} sxyP)$. So the solution to the CDHP problem is $xyp = (\sigma_{ID_*} k_i - h_{1_i}.P).{h_{2_i}}^{-1}.s^{-1}$.

$E_1$: F does not abort when the adversary $A$ asked sign query.

$E_2$: The adversary $A$ generates a valid message-signature forgery.

$E_3$: The event $E_2$ occurs and $ID^* = ID_t$

Let $q_s$ is the total number of sign queries, the adversary made. $F$ asked $q_h$ number queries to $H$. Then we could have that $Prob[E_1] \geq (1 - \frac{1}{q_s+1})^{qs}$, $Prob[E_2|E_1] \geq \varepsilon$ and $Prob[E_2|E_1 and\ E_3] \geq 1/q_h$.

So, $Prob[E_1 and E_2\ and\ E_3] \geq (1 - \frac{1}{q_s+1})^{qs}.\varepsilon.\frac{1}{q_h}$

which is $\varepsilon' \geq (1 - \frac{1}{q_s+1})^{qs}.\varepsilon.\frac{1}{q_h}$

and the time $t' = t + t_{SML} + q_c + q_{ppk} + q_{prk} + q_{reg} + q_{bf} + q_{bp} + q_H + q_{H_1} + q_{H_2})$.

Here it is to be noted that a single signature query by the adversary $A$, the challenger $F$ has to perform $\delta_{SM,}\, \delta_{P-ADD}, \delta_{H_1,} \delta_{H_2}$ and $\delta_{INV}$ operations where $\delta_{SM,}\, \delta_{P-ADD}$ and $\delta_{INV}$ are for the time to compute scalar multiplication, point addition and inverse calculation in $G_1$ respectively. $\delta_{H_1}$ and $\delta_{H_2}$ denote the time to perform hash functions operation $H_1$ and $H_2$ respectively. Therefore, if an adversary $A$ who can break our proposed VIDS scheme in polynomial time with non-negligible probability then CDHP problem can be solved in polynomial time with non-negligible probability.

## EFFICIENCY ANALYSIS

The proposed scheme is extension of Paterson's scheme [16] which is a classical identity-based signature scheme. Actually, the flavor of binding-blinding technique have been added with the Paterson's scheme [16] to get an identity-based signature scheme which can overcome the key escrow problem and eliminate the requirement of secure channel between PKG and user at the private key issuance phase.

For efficiency analysis of the proposed scheme, we have compared the scheme with a similar type scheme [14] proposed by M. L. Das. Both the schemes do not have any difference with respect to the involved operations in the process of private key generation but the signature generation process and the signature verification process of the proposed scheme are totally different from his proposed scheme [14]. So, the comparison of both the schemes is given in terms of involved operations in the process of signature generation and signature verification. $\partial_{HF,}$ $\partial_{EXP,}\partial_{SM}$, $\partial_{PA,}$ $\partial_{PO}, \partial_{MUL,}$ $\partial_{INV}$ and $\partial_{P\_INV}$ denote the hash operation, exponentiation operation in $G_2$, scalar multiplication in $G_1$, point addition in $G_1$, pairing operation, multiplication in the target group $G_2$, inversion in $Z_q$ and point inversion in $G_1$ respectively. Table 1 summarizes the result.

**Table 1:** Efficiency comparison of the proposed scheme

| VIDS Scheme | Signature Generation Process | Signature Verification Process |
|---|---|---|
| *M. L. Das Scheme* [14] | $1\partial_{HF,}$ $1\partial_{EXP}$ $2\partial_{SM}$, $1\partial_{PA,}$ $1\partial_{PO}$ | $1\partial_{HF,}$ $1\partial_{EXP}$ $1\partial_{MUL,}$ $\mathbf{1}\partial_{PO}$, $1\partial_{P\_INV}$ |
| *Proposed Scheme* | $2\partial_{HF,}$ $1\partial_{EXP}$ $3\partial_{SM}$, $1\partial_{PA,}$ $1\partial_{INV}$ | $2\partial_{HF,}$ $2\partial_{EXP,}$ $1\partial_{MUL,}$ $\mathbf{1}\partial_{PO}$ |

## CONCLUSION

We have proposed an identity-based signature scheme using blinding-binding method and bilinear pairings. The scheme is not only key escrow free but also eliminates the requirement of a secure channel at the key issuance phase. Moreover, our proposed scheme is provably secure based on the assumption that Computational Diffie-Hellman Problem is a hard problem. We compared the proposed scheme with similar kind of identity-based signature scheme proposed by M. L. Das. It has been observed that the proposed scheme is more efficient than the Das's scheme as no pairing operations are needed to sign a message and it is well known that the pairing operation is cost effective operation compare to other operations involved in a scheme. Moreover, the value of $e(P, P)$ in our scheme and the value of $e(pk_{ID}, Reg_{ID})$ in both the schemes can be pre-computed before the signature verification, such that the number of pairing operation can be reduced. As a result, only one pairing operation is needed to perform signature verification process in both the schemes. In addition to this, unlike the Das's scheme, no point inversion operation is involved in our proposed scheme.

## REFERENCES

[1] Menezes, A., P. C. van Oorschot, and S. Vanstone. 1996. Handbook of Applied Cryptography, CRC Press

[2] Shamir, A. 1984. Identity-Based Cryptosystems and Signature Schemes. In Advances in Cryptology - CRYPTO'84, LNCS 196, edited by G. R. Blakley and D. Chaum. Springer-Verlag, pp. 47–53.

[3] Guillou, L. and J. J. Quisquater. 1998. A Paradoxical Identity-Based Signature Scheme Resulting from Zero-Knowledge. In Advances in Cryptology - CRYPTO'88, LNCS 403,edited by S. Goldwasser. Berlin: Springer-Verlag, pp. 216–231.

[4] Fiat, A. and A. Shamir. 1986. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Advances in Cryptology - CRYPTO'86, LNCS 263, edited by A. M. Odlyzko. Berlin: Springer-Verlag, pp. 186–194.

[5] Boneh, D. and M. Franklin. 2001. Identity-Based Encryption from the Weil Pairing. In Advances in Cryptology - CRYPTO'01, LNCS 2139, edited by J. Kilian. Berlin: Springer-Verlag, pp. 213–229.

[6] Barreto, P. S. L. M. The Pairing-Based Crypto Lounge. http://www.larc.usp.br/ pbar-reto/pblounge.html

[7] Chen, L., K. Harrison, N. P. Smart, and D. Soldera. 2002. Application of Multiple Trust Authorities in Pairing Based Cryptosystems. In International Conference on Infrastructure Security-INFRASEC'02, LNCS 2437. Berlin: Springer-Verlag, pp. 260–275.

[8] Gentry, C. 2003. Certificate-Based Encryption and the Certificate Revocation Problem. In Advances in Cryptology - EUROCRYPT'03, LNCS 2656, edited by

E. Biham. Berlin: Springer-Verlag, pp. 272–293.

[9] Al-Riyami, S. and K. Paterson. 2003. Certificateless Public Key Cryptography. In Advances in Cryptology - ASIACRYPT'03, LNCS 2894, edited by C. S. Laih. Berling: Springer-Verlag, pp. 452–473.

[10] Lee, B., C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. 2004. Secure Key Issuing in ID-based Cryptography. In Proceedings of Australian Information Security Workshop- AISW'04, pp. 69–74.

[11] Das, M. L., A. Saxena, and D. B. Phatak. 2007. Proxy Signature Scheme with Effective Revocation using Bilinear Pairings. International Journal of Network Security, 4(3):312–317.

[12] Menezes, A. J., T. Okamoto, and S. A. Vanstone. 1993. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. IEEE Transactions on Information Theory, 39:1639–1646.

[13] Joux, A. 2000. One Round Protocol for Tripartite Die-Hellman. Proceedings of ANTS Lecture Notes in Computer Science, 4(1838):385–394.

[14] Das, Manik Lal. "A key escrow-free identity-based signature scheme without using secure channel." Cryptologia 35.1 (2010): 58-72

[15] Hess, F. 2002. An Efficient Identity Based Signature Schemes Based on Pairings. In Selected Areas in Cryptography. SAC'02, LNCS 2595, edited by K. Nyberg and H. Heys. Berlin: Springer-Verlag, pp. 310–324.

[16] Paterson K G. ID-based signatures from pairings on elliptic curves. IEEE Electronic Letters, 2002, 38(18): 1025–1026

[17] Goldwasser, Shafi, Silvio Micali, and Ronald L. Rivest. "A digital signature scheme secure against adaptive chosen-message attacks." SIAM Journal on Computing 17, no. 2 (1988): 281-308.