A Key Escrow-Free Identity-Based Signature Scheme from Bilinear Parings

Subhas Chandra Sahana\textsuperscript{1} and Bubu Bhuyan\textsuperscript{2}

\textsuperscript{1,2} Department of Information Technology
North Eastern Hill University, Shillong – 793022, INDIA

ABSTRACT
Identity-Based Cryptosystem was initiated with an aim of simplifying the key management technique but undergoes some inherent drawbacks, called key escrow problem. Moreover, it requires a secure channel at the time of private issuance by the Key Generation Centre (KGC). In our paper, we propose an identity-based signature scheme which neither has the key escrow problem nor requires a secure channel between the user and the KGC.

Keywords: Digital Signature, Bilinear pairings, Identity-Based Cryptosystem, Key escrow

1. INTRODUCTION
Identity Based Cryptosystem was introduced to eliminate the pitfall of traditional public key cryptosystem [1-2] which arises the presence of huge number of certificates. These certificates act as a ticket for gaining access in the cryptosystem, which makes it compulsory for all the users. As every user must be equipped with a public key, embedded in a certificate, issued and witnessed by a Certificate Authority (CA), the CA is responsible for maintaining the list of all the registered users along with their corresponding public keys and also maintains the list of valid and invalid users in case any weird situation has occurred. The message sending process needs to perform certificate verification process before encrypting or signing a message, which adds an extra effort. So, the traditional public key cryptosystem is considered unsuitable in terms of storage and computation cost due to certification management activities.

In Identity Based Cryptosystem (IBC) [3] it is quite easy to manage public and private key as public key can be easily obtained from the unique identity of the user and the private key is generated by the Private Key Generator (PKG) or KGC. No certificates are required here and hence it provides simplest key management technique. Despite of its simplicity it undergoes certain drawbacks also. The major drawbacks refer to the key escrow problem and the need of a secure channel between the PKG and the user. Key Escrow problem arises because the users rely on the PKG completely for the private key which results in enhancing the PKG ability for signing and decrypting a message has intended for a particular user. A secure channel is required to prevent insecure transmission of private key through the public channel. IBC was first introduced by Shamir [3]. Many more Identity (ID)-Based schemes [4-5] were introduced later on based on integer factorization and discrete logarithm problem. But later, an identity-based encryption scheme were developed based on bilinear pairings over elliptic curves by Boneh and Franklin [6] and finite fields as elliptic curve cryptography provides lower computation cost. So in this paper we introduce an identity-based signature scheme merged with a binding-blinding technique [7] to eradicate the drawbacks of the existing ID Based Signature schemes.

The paper is organised as follows: Section 2 includes some mathematical background required for our proposed scheme. Section 3 defines the proposed signature scheme along with correctness. Section 4 includes the implementation result of the proposed scheme. Afterwards, in section 5 efficiency analysis of the proposed scheme has been done and the last section, we conclude our work.

2. MATHEMATICAL BACKGROUND
2.1 Elliptic curve : An elliptic curve is a plane curve defined by the form \( y^2 = x^3 + ax + b \), where \( a \) and \( b \) are real numbers. An elliptic curve group over real numbers consists of the points on the corresponding elliptic curve, together with a special point \( \mathcal{O} \) called the point at infinity.

**POINT ADDITION**

1. The negative of a point \( P = (x_P, y_P) \) is its reflection in the x-axis and the point \(-P\) is \((x_P, -y_P)\).
2. If P and Q are two distinct points on an elliptic curve, then the curve joining the points P and Q intersects a point \(-R\) on the elliptic curve which is reflected in the x-axis to the point R and is defined by \(P + Q = R\).

3. Addition of point \(P\) and \(-P\) is a vertical line which does not intersects the elliptic curve at a third point so the elliptic curve includes a point at infinity \(O\) and is defined by \(P + (-P) = O\).

4. Doubling a point : If \(P = Q\)
   - A tangent line to the curve is drawn at the point P, then the tangent line intersects the elliptic curve at exactly one other point, \(-R. -R\) is reflected in the x-axis to \(R\). This operation is called doubling the point \(P\) on an elliptic curve group and is defined by \(P + P = 2P = R\).
   - A tangent line to the elliptic curve at \(P\) is vertical and does not intersect the elliptic curve at any other point and is defined as \(2P = O\).

2.2 DISCRETE LOGARITHM PROBLEM

Considering two prime numbers \(p\) and \(q\) satisfying the condition \(q|(p - 1)\), a random element \(g\) with order \(q\) in \(\mathbb{Z}_p^\ast\) is choosen and let \(g\) generate a random element say \(x\). If \(g^x = y \mod p\), the probability for finding \(x\) using any polynomial time algorithm \(S\) should be negligible.

2.3 Elliptic Curve Discrete Logarithm Problem

Given elliptic curve points \(P\) and \(Q\) in the group, to find a number \(k\) such that \(kP = Q\) (k is called the discrete logarithm of \(Q\) to the base \(P\)) is referred to as the ECDL problem.

2.4 Computational Diffie-Hellman Problem

Given \(P, aP, bP\) for \(a, b \in \mathbb{Z}_q\) computing \(abP\) is referred to as computational Diffie-Hellman problem.

2.5 Decisional Diffie-Hellman Problem

Given \(P, aP, bP, cP\) for \(a, b, c \in \mathbb{Z}_q\) deciding whether \(c = ab \mod q\) is referred to as decisional diffie-hellman problem.

2.6 Gap Diffie-Hellman Group

If in a group \(G\), decisional Diffie-Hellman problem is solvable using any probabilistic polynomial time algorithm but computational diffie-hellman problem is not solvable, then that group is referred to as gap Diffie-Hellman group.

2.7 Bilinear Diffie Hellman Problem

Given \(P, aP, bP\), \(cP\) for \(a, b, c \in \mathbb{Z}_q\) computing \(a(P, P)^{bc}\) is referred to as bilinear Diffie-Hellman problem.

3. PROPOSED IDENTITY BASED DIGITAL SIGNATURE SCHEME

The parties involved in the scheme are as follows:
- Key Generation Centre (KGC) : furnishes the signer with partial private key if the signer’s identity is valid.
- Signer: creates a signature.
- Verifier: has the ability to check or determine the correctness of the received signature for a message.

3.1 Binding-Blinding Technique proposed by DAS et al. [8]
This technique is put forward with an aim of preventing the leakage of useful information. Using this technique, the user at first chooses two secrets blinding parameters, using which four binding parameters are generated. These blinding parameters are then forwarded to the KGC over a public channel along with the user’s identity so that it can avoid unregistered identity attack. For avoiding the unregistered attack the email id should act as the ID of the user also the KGC validates the user’s ID asking confirmation message from the email-id owner. The KGC provides the user with the partial private key, sent through an insecure channel, only after the confirmation message is received and the validation of the blinding parameters. After receiving the partial key, the user calculates its private key after validation of the partial key.

3.2. The Proposed scheme in the VIDS model.

The proposed scheme is fitted in the VIDS model proposed by M.L. Das [7]. The scheme consists of the following algorithms:

- **System Parameters Generation:** This algorithm is employed for generating the parameters used for generating the signature. The parameters include \( G_1(\text{additive group}) \) and \( G_2(\text{multiplicative group}) \) of prime order \( q \). A bilinear pairing \( e: G_1 \times G_2 \rightarrow G_P \) is the generator of group \( G_2 \) and a map-to-point hash function \( H: \{0,1\}^* \rightarrow G_2 \).

- **KGC’s Key Generation:** KGC sets its public and private key in this step. A secret value \( s \in \mathbb{Z}_q^* \) is selected by the KGC which is set as its private key \( s_{PRK} \) and its public key is set as \( s_{PK} = s.P \).

- **User’s Key Generation:** User’s private key and public key is calculated using the binding-binding technique [8].

1. **Binding Parameter Generation:** In this step two secret binding factors are chosen using which binding parameters are generated along with the system parameters.

   - Using his identity ID, the user computes \( s_{PKID} = H(ID) \).

   - Two secret blinding factors \( a, b \in \mathbb{Z}_q^* \) are chosen randomly by the user using which creates the four binding parameters \( X = a.s_{PKID}, Y = a.b.s_{PKID}, z = b.P \) and \( W = a.b.P \). Afterwards \( \{X, Y, Z, W, ID\} \) has been sent to the KGC with the ID through the public channel.

   - If the ID does not exist in its directory, the KGC validates the user asking for a confirmation message from the email id owner. This step is used to prevent unregistered ID attack.

2. **Partial Key Generation:** The partial key \( D_{ID} \) is calculated as follows:

   - After the KGC receives the confirmation message, it calculates \( s_{PKID} = H(ID) \) and verification of the user is done by checking the condition \( e(Y, P) = e(X, Z) = e(s_{PKID}, W) \).

   - The user’s partial key \( D_{ID} = s.Y \) and a registration-token \( Reg_{ID} = s.Z \) is calculated by the KGC if the above condition holds and \( < Reg_{ID}, ID > \) is published in the KGC’s directory and send \( D_{ID} \) to the user by the KGC.

3. **Private Key Generation:** Using this algorithm the user’s private key is evaluated using one of the binding factor \( a \) and generator \( P \) and the partial key \( D_{ID} \) in the following way.

   - The condition \( e(D_{ID}, P) = e(Y, s_{PKID}) \) is checked after \( D_{ID} \) is received by the user.

   - The user unbinds \( D_{ID} \) and evaluates the private key \( S_{ID} = a^{-2}D_{ID} = b.s.P_{ID} \) if the above condition holds.
• Sign: The signer randomly chooses \( r \in \mathbb{Z}_q^* \) and computes the following:
  
  1. \( S_1 = S_{ID} \cdot r \cdot H(M) \)
  
  2. \( S_2 = r \cdot P \)

  and publishes the signature \( \{ S_1, S_2 \} \) on message \( M \) for the identity \( ID \).

• Verify: The signature is valid only if \( e(\phi_{ID}^*, \kappa_{ID}^*) \cdot e(H(M), S_2) = e(S_1, P) \).

3.3. Correctness.

\[
e(\phi_{ID}^*, \kappa_{ID}^*) \cdot e(H(M), S_2) = e(\phi_{ID}^*, \kappa_{ID}^*) \cdot e(H(M), r \cdot P) \\
= e(\phi_{ID}^*, \kappa_{ID}^*) \cdot e(H(M), r \cdot P) \\
= e(\phi_{ID}^*, \kappa_{ID}^*) \cdot e(r \cdot H(M), P) \\
= e(S_2, P) \cdot e(r \cdot H(M), P) \\
= e(S_2, P)
\]

4. EFFICIENCY ANALYSIS

We have compared the proposed key escrow free identity-based signature scheme with some previously similar established existing signature scheme [7]. We have implemented both the proposed signature scheme and Das’s scheme in Linux systems with system configuration Intel Core i3 CPU 2.53GHz and 2GB RAM using Pairing Based Cryptography (PBC) library [10] in C. Table 1 depicts the running time consumption for both the schemes. As both the schemes undergo the same private key generation process therefore consumes same amount of time, but differs in terms of the signature generation process and signature verification process.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Key Generation (MS)</th>
<th>Signing (MS)</th>
<th>Verifying (MS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed scheme</td>
<td>71.853</td>
<td>7.851</td>
<td>12.887</td>
</tr>
</tbody>
</table>

Table 1. Running time consumption

![Figure 1. Bar chart representation of the consumed running time](image-url)
We use the identity-based signature scheme proposed by Sakai et al. [11] as a base scheme which is merged with the binding-blinding technique occurs as an efficient scheme removing the key escrow problem and the necessity of the secure channel. Different operations in signature generation and verification process are denoted as $\sigma_{Exp}$-exponentiation, $\sigma_{H}$-hash operation, $\sigma_{M}$-operation, $\sigma_{M}$-scalar multiplication in group $G_t$, $\sigma_{Inv}$-inversion in $Z_n$ -point addition in $G_t$, $\sigma_{Pair}$-pairing operation, $\sigma_{Pair}$-the point inversion in $G_t$, $\sigma_{Mult}$-Multiplication in group $G_t$, $\sigma_{Map}$-Map-to-Point hash function.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Private-Key Generation</th>
<th>Signing</th>
<th>Verification</th>
<th>Generated signature length</th>
</tr>
</thead>
<tbody>
<tr>
<td>[7]</td>
<td>$16\sigma_{M}+6\sigma_{H}+5\sigma_{Pair}+16\sigma_{Inv}$</td>
<td>$16\sigma_{Pair}+2\sigma_{Exp}+10\sigma_{Pair}+10\sigma_{Exp}$</td>
<td>$2\sigma_{Pair}+10\sigma_{Pair}+16\sigma_{Pair}+10\sigma_{Pair}+16\sigma_{Exp}+16\sigma_{Inv}$</td>
<td>$</td>
</tr>
<tr>
<td>Proposed</td>
<td>$16\sigma_{M}+6\sigma_{H}+5\sigma_{Pair}+16\sigma_{Inv}$</td>
<td>$16\sigma_{M}+6\sigma_{H}+5\sigma_{Pair}+16\sigma_{Inv}$</td>
<td>$16\sigma_{M}+36\sigma_{Pair}$</td>
<td>$</td>
</tr>
</tbody>
</table>

5. CONCLUSION

We have proposed a scheme which has same number of operations in the process of private-key generation but less number of operations in the processes of both signing and verification than the similar type of scheme proposed by M.L. Das. So the proposed scheme is more efficient than the Das’s scheme. Like the Das’s scheme, our scheme is also key escrow-free and does not require any secure channel for the transmission of the private generated by the KGC. It has been observed that the proposed scheme consumes lesser running time compare to the other scheme.

References