Efficient ID-Based Signature Scheme using Pairings

P.V.S.S.N. Gopal, P. Vasudeva Reddy
Department of Engineering Mathematics, Andhra University, Visakhapatnam, India.
gopalcrypto786@gmail.com, vasucrypto@yahoo.com

Abstract—An ID-based cryptographic scheme enables the user to public keys without exchanging public key certificates. In these schemes, users can generate their public and private keys using their identity. The positive application of bilinear pairings over elliptic curves makes the system easy and efficient in providing security. In this paper, we propose an ID-based signature scheme using bilinear pairings. We prove that the proposed signature scheme is secure against existential forgery under adaptively chosen message and ID attack in the random oracle model with the assumption that the Computational Diffie-Hellman problem is hard. We compare the efficiency of the proposed scheme with some related schemes.

Keywords—ID-based cryptography; Digital signature scheme; Bilinear pairings; Unforgeability; CDH problem.

I. INTRODUCTION

Digital signatures are undoubtedly the key ingredients to provide authentication, integrity and non-repudiation of data being transmitted. The basic idea of a digital signature assures a receiver that the corresponding sender indeed signed the document/message. Several signature schemes in the Public Key Infrastructure (PKI) setting appeared in the literature [6, 7, 13, 16]. In PKI setting the identity of a user is a random string that is unrelated to identity of the user. Certainly this is a big drawback as maintaining such strings need huge storage requirements.

The evolution of identity based cryptography devised by Shamir [19] in 1984 created enthusiasm and interest in many researchers to construct signatures schemes in the ID-based setting to cater different needs of society. In such a cryptosystem the public key of a user is directly derived from his/her identity such as driving license number, I.P. address etc. Moreover, with the advent of elliptic curves which maintains less storage size than the traditional PKI based schemes the need to construct ID-based signatures (IBS) schemes using pairings took a drastic turn. In 2001, the first scheme for encryption in the ID-based setting using Weil pairing over elliptic curves was devised by Boneh et al. [4]. Many signatures schemes using pairings in the ID-based setting appeared in the literature [5, 10, 11, 12, 14, 17, 18, 20].

In 2000, Sakai et al. [18] proposed a cryptosystem based on pairings and constructed an IBS scheme. This scheme requires three pairing operations in the verification phase.

In 2003, Hess [11] proposed an efficient IBS scheme based on pairings. This scheme requires one pairing operation in the signing phase and two pairing operations in the verification phase.

In 2003, Cha and Cheon [5] proposed an IBS scheme from the Gap Diffie-Hellman Groups. This scheme requires two pairing operations in the verification phase.

In 2005, Huang et.al. [12] proposed an efficient IBS scheme and a blind signature scheme using pairings. This scheme requires one pairing operation in the signing phase and two pairing operations in the verification phase.
In 2010, Shim [20] proposed an ID-based aggregate signature scheme with constant pairing computations. In this the author proposed an efficient IBS scheme, which requires two pairing computations in the verification phase.

In 2012, Hafizul Islam et al. [10] proposed an efficient and provably-secure digital signature scheme based on elliptic curve bilinear pairings. In this the authors employed the K-CAA technique for key construction. This scheme requires two pairing operations in the verification phase.

The schemes in [18, 11, 5, 12, 20, 10] are proved secure against existential forgery under adaptive chosen message and ID-attack in the random oracle model with the assumption that the CDH problem is intractable. Their security proofs are obtained through pointcheval and Stern [15].

To improve the computational efficiency, in this paper, we propose a new and efficient IBS scheme using bilinear pairings. This scheme is secure against existential forgery under adaptive chosen message and ID attack with the assumption that the CDH problem is hard. The proposed scheme requires two pairing operations in the verification phase. We compare the proposed scheme with related IBS schemes.

The rest of the paper is organized as follows: In section 2, we describe some preliminaries. In Section 3, syntax and security model for ID-based signature scheme is described. Our proposed scheme is depicted in Section 4. Security analysis of the proposed scheme is presented in Section 5. Section 6 provides efficiency analysis of our scheme and finally Section 7 gives conclusion.

II. PRELIMINARIES

In this section, we briefly review the preliminaries and computational hard problems which form the basis for constructing the proposed scheme.

Bilinear pairing: It is an important cryptographic primitive and is widely adopted in many positive applications of cryptography. Let \( G_1 \) and \( G_2 \) are additive and multiplicative cyclic groups respectively of same prime order \( q \) with \( P \) as a generator of \( G_1 \). A bilinear is a map \( \hat{e} \) defined by \( \hat{e} : G_1 \times G_1 \rightarrow G_2 \) satisfying the following properties:

1) \textbf{Bilinearity:} For all \( P, Q \in G_1 \) and \( a, b \in \mathbb{Z}_q^* \), \( \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} \).

2) \textbf{Non-Degeneracy:} There exists \( P \in G_1 \), such that \( \hat{e}(P, P) \neq 1 \).

3) \textbf{Computability:} There exists an efficient algorithm to compute \( \hat{e}(P, Q) \) for all \( P, Q \in G_1 \).

Such a pairing \( \hat{e} \) is called an admissible pairing, and can be constructed by the modified Weil or Tate pairings on elliptic curves defined over a finite field [2, 3, 8].

We now present some computational hard problems in the group \( G_1 \), which form the basis of security of the proposed scheme.

1) \textbf{Computational Diffie-Hellman (CDH) Problem:} Given \( P, aP, bP \in G_1 \) for some \( a, b \in \mathbb{Z}_q^* \), the CDH problem is to compute \( abP \in G_1 \).

2) \textbf{Computational Diffie-Hellman (CDH) Assumption:} Given \( P, aP, bP \in G_1 \) for some \( a, b \in \mathbb{Z}_q^* \), there does not exist probabilistic polynomial-time adversary \( A \) with non negligible probability who can compute \( abP \in G_1 \). The advantage of \( A \) within running time \( t \) is defined by

\[ \text{Adv}_{\text{CDH}}(t) = \Pr[A(P, aP, bP) = abP / P, aP, bP \in G_1] \]

III. SYNTAX AND SECURITY MODEL OF IBS SCHEME

In this section, we describe the formal definition of an ID-based signature scheme and its security model.

A. Syntax of IBS Scheme

An IBS scheme consists of four polynomial time algorithms namely Setup, Extract, Sign, and Verify. Here we present the detailed functionalities of these algorithms.

\textit{Setup:} The PKG generates the system parameters \( \text{Params} \) and the master key \( s \); \( \text{Params} \) are made public and \( s \) is kept secret. \( \text{Params} \) are implicit input to all the following algorithms.
Extract: Given a user’s identity $ID$, and the master key $s$, the PKG computes the public key $Q_{ID} = H_1(ID)$ and the private key $d_{ID} = sQ_{ID}$. PKG sends $d_{ID}$ to the corresponding user through a secure channel.

Signature Generation: On input signers identity $ID$ corresponding to the $ID$ and a message $M \in \{0, 1\}^*$, this algorithm outputs a signature $\sigma = \text{Sign}(d_{ID}, M)$.

Signature Verification: Given a signer’s identity $ID$, a message $M \in \{0, 1\}^*$, and a signature $\sigma$, a verifier checks the validity of $\sigma$. More precisely, the algorithm $\text{Verify}(M, ID, \sigma)$ and outputs 1 if accepted, or 0 if rejected.

B. Security of IBS Scheme

The most general security notion of a signature scheme is existential unforgeability under an adaptive chosen message attack [9]. It is extended to an IBS scheme, namely, existential unforgeability under an adaptive chosen-message and an adaptive chosen-ID attack. It is formalized as follows:

Unforgeability of IBS scheme against an adaptive chosen message attack and an adaptive chosen-ID attack:

In this model an adversary can choose its messages and its identities adaptively. We give the adversary the power to request private keys on identities of its choice. The adversary is also given access to the signing oracle for any messages for desired identities. An adversary’s advantage $\text{Adv}_{\text{IBS}, A}$ is defined as its probability of success in the following game between a challenger $C$ and an adversary $A$.

Setup: The challenger $C$ takes a security parameter $l$ and runs the setup algorithm of the IBS scheme. It gives the $\text{Params}$ to $A$ and keeps the master secret with itself.

Queries: The adversary $A$ adaptively makes different queries to the challenger $C$. Each query can be one of the following.

Extract Queries: When $A$ requests a private key corresponds to an identity $ID$ of its choice, $C$ responds by running the extract algorithm of the scheme and corresponds to $A$ with its private key $d_{ID}$.

Sign Queries: When $A$ requests adaptively a signature on a given message $M$, with an identity $ID$, $C$ returns a signature $\sigma$.

Hash Queries: When the involved hash functions are modeled by random oracles, $A$ also performs adaptive queries to the hash functions. The Challenger must answer queries of the adversary of this oracle, providing it with consistent and totally random values.

All these queries can be made in an adaptive way; i.e. each query may depend on the answers obtained to the previous queries.

Output: $A$ outputs $(M, ID, \sigma)$ and we say that $A$ succeeds if:

(i) It has not requested an extraction query for $ID$
(ii) $\sigma$ has not been obtained as an answer of the challenger to a sign query $(M, ID)$
(iii) $\sigma$ is a valid signature.

The advantage of $A$ in the above game is defined as $\text{Adv}_{A} = \Pr[A \text{ succeeds}]$, where the probability is taken overall coin tosses made by $C$ and $A$. We note that the above game captures the notion of strong unforgeability, introduced by An et al. in [1].

Definition 3.1: A forger $A(t, q_E, q_S, q_H, \epsilon)$ breaks an IBS scheme if $A$ runs in time at most $t$, $A$ makes at most $q_E$ Extract queries, $q_S$ sign queries and $q_H$ hash queries, and $\text{Adv}_{\text{IBS}, A}$ is at least $\epsilon$ in the above game. An IBS scheme is $(t, q_E, q_S, q_H, \epsilon)$-existentially unforgeable under an adaptive chosen-message attack and adaptive chosen ID-attack if no forger $(t, q_E, q_S, q_H, \epsilon)$ breaks it.

IV. NEW ID-BASED SIGNATURE SCHEME

In this section, we present a new and efficient IBS scheme. This scheme consists of the following four polynomial time algorithms: Setup, Key Extract, Signature Generation and Signature Verification. We now describe the detailed functionalities of these algorithms.
Setup: Given a security parameter $l$, the PKG chooses additive and multiplicative cyclic groups say $G_1, G_2$ respectively of same prime order $q \geq 2^l$ with a bilinear pairing $\hat{e}: G_1 \times G_1 \rightarrow G_2$; and $P \in G_1$ as a generator of $G_1$. PKG selects a random integer $s \in_R Z_q^*$, computes the system public key $P_{pub} = sP$ and $g = \hat{e}(P_{pub}, P)$. PKG picks two hash functions $H_1 : [0, 1]^* \rightarrow G_1$, $H_2 : [0, 1]^* \times G_2 \rightarrow Z_q^*$, publishes the system parameters as $\text{Params} = < G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, g >$, and keeps the master key $< s >$ as secret.

Key Extract: Given a user’s identity $ID \in [0, 1]^*$, the PKG computes $Q_ID = H_1(ID) \in G_1$, and $d_{ID} = sQ_ID \in G_1$. PKG transmits $d_{ID}$ the user’s private key, via a secure channel.

Signature Generation: To sign a message $M \in [0, 1]^*$, the signer first chooses a random integer $r \in Z_q^*$ and then computes $U = g^r \in G_2$, $h = H_2(M, U) \in Z_q^*$, $V = hd_{ID} + rP_{pub} \in G_1$. The signature on message $M$ is $\sigma = (U, V) \in G_2 \times G_1$.

Signature Verification: On receiving a message $M$ and signature $\sigma = (U, V)$, a verifier computes $h = H_2(M, U) \in Z_q^*$. He then accepts the signature if and only if $\hat{e}(P, V) = \hat{e}(P_{pub}, hQ_ID)U$.

V. SECURITY ANALYSIS OF THE PROPOSED IBS SCHEME

In this section, we analyze the proof of correctness and security of our proposed scheme.

Proof of Correctness: It is easy to verify that the proposed signature scheme is correct as shown in the following:

\[
\hat{e}(P, V) = \hat{e}(P, hd_{ID} + rP_{pub}) = \hat{e}(P, hsQ_ID) \hat{e}(P, P_{pub})^r = \hat{e}(sP, hQ_ID) \hat{e}(sP, P)^r = \hat{e}(P_{pub}, hQ_ID)U. \]

Now, we prove that the proposed IBS scheme is secure against existential forgery on an adaptive chosen message and ID attack in the random oracle model with the assumption that the CDH problem is intractable.

Theorem 5.1: If the CDH Problem is $(t', \varepsilon')$–hard, the scheme IBS is $(t, q_{H_1}, q_{H_2}, q_E, q_S, \varepsilon)$–secure against existential forgery under adaptive chosen-message and adaptive chosen ID-attack, for any $t$ and $\varepsilon$ satisfying $\varepsilon \geq \varepsilon' \cdot (q_E + 1) \cdot \varepsilon'$. $t \leq t' - c_{G_1} c_{G_2} (q_{H_1} + q_E + 2q_S + 3)$, where $e$ is the base of the natural logarithm, $c_{G_1}$ is the time of computing a scalar multiplication in $G_1$ and an inversion in $Z_q^*$. $c_{G_2}$ is the time for calculating one pairing.

Proof: Let $A$ is a forger who breaks the proposed IBS scheme. A CDH instance $(P, aP, bP)$ is given for $a, b \in_R Z_q^*$. By using the forgery algorithm $A$, we will construct an algorithm $B$ which outputs the CDH solution $abP$ in $G_1$. Algorithm $B$ performs the following simulation by interacting with the forger $A$.

Algorithm $B$ simulates a real signer to get a valid signature from the forger $A$. If $B$ does not fail this simulation, he/she gets a valid signature, and using the oracle replaying technique [15] he can solve the CDH problem.

Setup: Algorithm $B$ sets the system public key as $P_{pub} = aP$ and starts by giving $A$ the system parameters $\text{Params}$. At any time, $A$ can query the random oracles $H_1, H_2$, Extract and Sign queries. Without loss of generality, we assume that, for any extract query, sign query involving an identity, an $H_1$ oracle query has previously been made on that identity. To respond these queries, $B$ does the following:
$H_1$ Queries: When A queries the oracle $H_1$ at a point $ID \in \{0, 1\}^*$, algorithm B maintains a list referred as $H_1$ - list of tuples $(ID, w, x, y)$ and responds to A as explained below.

1. If the query $ID$ already appears on the $H_1$ - list in a tuple $(ID, w, x, y)$ then B responds with $H_1(ID) = w \in G_1$.
2. Otherwise, B picks a random coin $y \in \{0, 1\}$, such that $pr[y = 0] = 1/(q_E + 1)$.
3. Algorithm B picks a random $x \in Z_q^*$.
   - If $y = 0$, B computes $w = x(bP) \in G_1$.
   - If $y = 1$, B computes $w = xP \in G_1$.
4. B adds the tuple $(ID, w, x, y)$ to the $H_1$ - list and responds to A with $H_1(ID) = w \in G_1$.

$H_2$ Queries: At any time A queries the oracle $H_2$ at $(ID, M, U)$, algorithm B maintains a list referred as $H_2$ - list of tuples $(ID, M, U, v)$ and responds to A as explained below.

1. If the query tuple $(ID, M, U)$ already appears on the $H_2$ - list in a tuple $(ID, M, U, v)$ then B responds with $H_2(ID, M, U) = v \in Z_q^*$.
2. Otherwise, B picks a random $v \in Z_q^*$ and adds the tuple $(ID, M, U, v)$ in the $H_2$ - list and responds to A with $H_2(ID, M, U) = v \in Z_q^*$.

Extract Queries: When A queries the private key associated to $ID$, B first recovers the corresponding tuple $(ID, w, x, y)$ from the $H_1$ - list.

1. If $y = 0$, then B output failure and halts.
2. Otherwise, B computes $d_{ID} = xP_{pub} = x(aP) = a(xP) \in G_1$ by using the tuple $(ID, w, x, y)$ in the $H_1$ - list and returns to A with $d_{ID}$.

Sign Queries: When A queries a signature on a message $M$ for an identity $ID$, B first find the tuple $(ID, W, x, y)$ from $H_1$ - list, chooses a random integer $t \in Z_q^*$ and computes $U = g^t$, where $g = \hat{e}(P_{pub}, P)$. If the tuple $(ID, M, U, v)$ already appears on the $H_2$ - list, B chooses $v \in Z_q^*$ and tries again. Otherwise, B computes $V = (vx + t)P_{pub}$ and stores $(ID, M, U, v)$ in the $H_2$ - list, responds to A with $\sigma = (U, V)$ as a signature on the message $M$.

All responses to sign queries are valid; indeed, the output $(U, V)$ of sign query is a valid signature on $M$ under $ID$. To see this

\[
\hat{e}(P_{pub}, hQ_{ID})\hat{e}(P_{pub}, P)^t = \hat{e}(aP, hQ_{ID})\hat{e}(aP, tP) = \hat{e}(P, ha(xP))\hat{e}(P, tP_{pub}) = \hat{e}(P, hXP_{pub})\hat{e}(P, tP_{pub}) = \hat{e}(P, (vx + t)P_{pub}) = \hat{e}(P, V).
\]

Output: Finally, A output a forgeable signature $(U^*, V^*)$ on a message $M^*$ for an identity $ID^*$. According to Forking Lemma [15], B rewind A to the point which it just queries $h^*$ and returns a different value with the same input to hash query. Thus B obtains two valid signatures $\sigma = (ID^*, M, U^*, V^*)$ and $\sigma' = (ID^*, M, U'^*, V'^*)$ within a polynomial time, where $V^* = h^*d_{ID}^* + tP_{pub}$, $V'^* = h^*d_{ID}' + tP_{pub}$. Then
\(B\) finds the corresponding tuple \((ID, w, x, y)\) from the \(H_1\)-list. If \(y = 1\), \(B\) declares failure and halts. Otherwise, \(B\) computes \(V^* = V^{*'} = [h^* - h'^*]d_ID^*\)
\[\Rightarrow V^* - V^{*'} = [h^* - h'^*]d(xbP)\]
\[\Rightarrow abP = \frac{x^{-1}(V^* - V^{*'})}{(h^* - h'^*)}.\]

This completes the description of \(B\). It remains to show that \(B\) solves the given instance of the CDH problem in \(G_1\) with probability at least \(\varepsilon'\). To do so, we analyze three events needed for \(B\) to succeed:

- \(E_1\): \(B\) does not abort as a result of any \(A\)’s Extract query.
- \(E_2\): \(A\) generates a valid and nontrivial signature forgery \(\sigma = (U, V)\).
- \(E_3\): Event \(E_2\) occurs and \(y = 0\) for the tuple containing \(ID\) on the \(H_1\)-list.

Algorithm \(B\) succeeds if all of these events happen. The probability \(\Pr[E_1 \land E_2 \land E_3]\) can be decomposed as
\[\Pr[E_1 \land E_2 \land E_3] = \Pr[E_1] \cdot \Pr[E_2/E_1] \cdot \Pr[E_3/E_1 \land E_2]\]
(1)

The following claims give a lower bound for each of these terms.

**Claim 1:** The probability that algorithm \(B\) does not abort as a result of \(A\)’s Extract query is at least
\[1 - \left(1 - \frac{1}{(q_E + 1)}\right)^{q_E}.\] Since \(A\) makes at most \(q_E\) queries to the Extract oracle and
\[\Pr[y = 1] = 1 - \frac{1}{(q_E + 1)},\] we have
\[\Pr[E_1] = \left(1 - \frac{1}{(q_E + 1)}\right).\]

**Claim 2:** If \(B\) does not abort as a result of \(A\)’s Extract query then \(A\)’s view is identical to its view in the real attack. Hence \(\Pr[E_2/E_1] \geq \varepsilon\).

**Claim 3:** The probability that \(B\) does not abort \(A\) outputs a valid and nontrivial forgery is at least \(\frac{1}{(q_E + 1)}\).

Algorithm \(B\) will abort only if \(A\) generates a forgery such that \(y = 1\).

Hence
\[\Pr[E_3/E_1 \land E_2] \geq \frac{1}{(q_E + 1)}.\]

To complete the proof of Theorem 5.1, we use the bounds from the claims above in equation (1). Algorithm \(B\) produces the correct answer with the probability at least
\[\left(1 - \frac{1}{(q_E + 1)}\right)^{q_E} \cdot \frac{1}{(q_E + 1)} \geq \frac{1}{e} \cdot \frac{e}{(q_E + 1)} \geq \varepsilon'.\]

Hence \(e(q_E + 1)e' < e\) as required.

Algorithm \(B\)’s running time is the same as \(A\)’s running time plus the time it takes to respond to \((q_{H_1} + q_{H_2})\) hash queries, \(q_E\) Extract queries and \(q_S\) Sign queries, and the time to transform \(A\)’s final forgery into the CDH solution. The \(H_1\) and Extract queries requires a scalar multiplication. The sign query requires one exponentiation, 2 scalar multiplications and one pairing evaluation. The output phase requires 1 inversion and a scalar multiplication. We assume that \(c_{G_i}\) is the time taking for a scalar multiplication in \(G_i\) and an inversion in \(Z^*_q\); \(c_{G_2}\) is the time taking for one pairing evaluation in \(G_2\). Hence, the total running time is at most \(t + c_{G_1} c_{G_2} (q_{H_1} + q_E + 2q_s + 3) \leq t'\) as required.
VI. EFFICIENCY ANALYSIS

To compare the computing efficiency of our scheme with other schemes of this kind, we consider the time-exhausting operations. Let $C_a$ denotes the cost of point addition over $G_1$. $C_m$ denotes the cost of point scalar multiplication over $G_1$. $C_p$ denotes the cost of pairing operation. $C_e$ denotes the cost of exponent operation over $G_2$. $C_i$ cost of inversion in $\mathbb{Z}_q^*$.

Despite a number of attempts [2, 3, 8] to reduce the complexity of pairing, still the operation is very costly. For example, according to the result in [3], one pairing operation is about 11110 multiplications in $F_{3^{63}}$, while a point scalar multiplication of $E/F_{3^{63}}$ is a few hundred multiplications in $F_{3^{63}}$. In our scheme, $g = \hat{e}(P_{pub}, P)$ can be precomputed and published as a system parameter. So there are only two pairing operations required for verification. Therefore, compared with the schemes in [11, 12, 18], our scheme is efficient, since the schemes in [11, 12, 18] requires three pairing computations in both signing and verification phases. Further, the proposed scheme is equally efficient when compared with the schemes in [5, 10, 20].

TABLE 1. EFFICIENCY COMPARISON

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Signature Generation</th>
<th>Signature Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cha-Cheon [5]</td>
<td>$2C_m$</td>
<td>$2C_p + 4C_m + 1C_e$</td>
</tr>
<tr>
<td>Islam et al. [10]</td>
<td>$1C_i + 3C_m + 1C_a$</td>
<td>$2C_p + 4C_m + 1C_a$</td>
</tr>
<tr>
<td>Hess [11]</td>
<td>$1C_i + 2C_m + 1C_a$</td>
<td>$2C_p + 1C_e$</td>
</tr>
<tr>
<td>Huang et al. [12]</td>
<td>$1C_p + 4C_m + 1C_e$</td>
<td>$2C_p + 1C_e$</td>
</tr>
<tr>
<td>Sakai et al. [18]</td>
<td>$2C_m + 1C_a$</td>
<td>$3C_p$</td>
</tr>
<tr>
<td>Shim [20]</td>
<td>$3C_m + 1C_a$</td>
<td>$2C_p + 1C_m + 1C_a$</td>
</tr>
<tr>
<td>Our Scheme</td>
<td>$1C_m + 1C_a + 1C_e$</td>
<td>$2C_p + 1C_m$</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

Digital signature is an important tool in public key cryptography for secure authentication. With the exploit of bilinear pairings several efficient and secure ID-based signature schemes have been proposed till now. In this paper, we have proposed an ID-based signature scheme using bilinear pairings. We have proved that the proposed scheme is unforgeable in the random oracle model with the assumption that the CDH problem is intractable. The proposed scheme is efficient when compared with the schemes in [11, 12, 18], since our scheme requires less number of pairing computations.

REFERENCES


