An Efficient and Provably-secure Digital signature Scheme based on Elliptic Curve Bilinear Pairings

SK HAFIZUL ISLAM, G.P. BISWAS

Department of Computer Science and Engineering
Indian School of Mines, Dhanbad-826004, Jharkhand, India
e-mail: hafi786@gmail.com, gpbiswas@gmail.com

Received 8 May 2012, Revised 2 June 2012, Accepted 22 June 2012.

Streszczenie We proposed an efficient and secure digital signature scheme using elliptic curve cryptography (ECC) and bilinear pairings in this paper. The proposed scheme employs the general cryptographic hash function (i.e., SHA-1) instead of map-to-point function, because the map-to-point is a cost-intensive operation and it is usually implemented as a probabilistic algorithm. Further, our scheme is computationally efficient as one bilinear paring and three elliptic curve scalar point multiplication operations are executed for signature generation and verification, and thus the scheme requires much lesser computation cost than other related schemes. In addition, in the random oracle model, our scheme is proven to be existential unforgeable against the adaptive chosen message and identity attacks (EUF-CMA) based on a variation of the collusion attack algorithm with k traitors (k-CAA3) problem.

Keywords: Elliptic curve cryptography, Bilinear pairing, Map-to-point function, Digital signature, Random oracle model, Provably-secure

1. Introduction

The authenticity, integrity and non-repudiation of the digital documents that are transmitted over any public network, can be achieved by using the digital signature. The digital signature is nothing but the string of digital bits and is different from any hand-written signature. That is, the digital signature of a person is changed if the message varies, whereas the handwritten signature computed by a person is fixed for all documents. The signature scheme can be found useful in many applications of network security where the detection of forgery or protection of tampering of digital documents is necessary.
1.1. Literature Review

The public key cryptography (PKC) was proposed by Diffie and Hellman [1] in which two keys are used, called private key and public key. The user chooses his private key that is to be kept secret while the corresponding public key is known to all and thus, it needs to be authenticated by a trusted third party, named as a certificate authority (CA). After the pioneer work of Diffie-Hellman, many digital signature schemes have been proposed [2−5,7−11] based on either the large integer factorization problem (IFP) or the discrete logarithm problem (DLP). Based on IFP and DLP, Shao [5] proposed a digital signature scheme and claimed that it is more secure than ElGamal scheme [3] with similar computation cost. However, Li and Xiao [6] presented a simple attack and proved that Shao’s scheme is insecure. In the RSA scheme [2], the signature length is same as the length of the modulus used and in the ElGamal scheme [3], the length of the signature is twice the length of the modulus used. In order to reduce the signature length, Schnorr [7] proposed a signature scheme where the length of the signature is independent of the length of the modulus used.

To enhance the security, Harn [8] in 1994 also developed a new signature scheme based on two different cryptographic assumptions. He claimed that the security of the scheme could be compromised if both IFP and DLP assumptions are simultaneously solvable by a polynomial time-bounded algorithm. The idea of combining more than one computational problem is good from the security point of view, however, Lee and Hwang [9] proved that if one of the assumption, say DLP is solvable then Harn’s scheme is completely breakable with high probabilities. Subsequently, they proposed a modified scheme based on the hardness assumptions as adopted in the Harn’s scheme to defeat the problem pointed out by them. In 2000, Nyang and Song [10], proposed an efficient digital signature scheme using a zero-knowledge based identification (ZKI) scheme and hash function. The computational performance of the scheme is better than other RSA-like schemes and other well-known signature schemes also. In 2007, Chung et al. [11] proposed another ZKI-based signature scheme using ECC, however, the scheme is not secure as demonstrated by Yang and Chang [12].

1.2. Motivations and Contributions

Recently, ECC [13, 14] and bilinear pairing [15] have been received great attention due to the following reasons: (1) ECC needs smaller key size, lesser bandwidth, low computation cost and low storage space. (2) The efficient algorithms to compute the bilinear pairing (Weil pairing [15] or Tate pairing [16]) are available in the literature. The computation cost for signature generation and verification of the previous schemes are very high since they employed a cost-intensive operation, called modular exponentiation. Besides, most of the previous schemes are insecure against different attacks [6, 9,
In order to speed up the signature generation and verification, and to provide strong security, we proposed an efficient and secure digital signature scheme using ECC and bilinear pairings, where instead of map-to-point (MTP) function, the general cryptographic hash function (i.e., SHA-1) is used in our scheme. Because, the MTP is a cost-intensive operation and it is usually implemented as a probabilistic polynomial time-bounded algorithm. Based on a variation of the collusion attack algorithm with $k$ traitors ($k$-CAA3) assumption [17, 18], the proposed scheme is existential unforgeable in the random oracle model [19] against the adaptive chosen message and identity adversary. In addition, the computational performance of the proposed scheme is better than other schemes as shown in section 4.3.

1.3. Roadmap of the Paper

The rest of the paper is organized as follows. Section 2 provides the necessary technical details required throughout the paper and in Section 3, we present our digital signature scheme. The analyses in terms security and performance of the proposed scheme are given in Section 4. Finally, Section 5 gives the concluding remarks.

2. Preliminaries

In this section, we briefly introduced the concepts of elliptic curve-based bilinear pairings and some hard mathematical problems.

2.1. Bilinear Pairings

Let $G_q$ be a cyclic additive group, which is generated by $P$, with prime order $q$, and $G_m$ is a cyclic multiplicative group with the same order $q$. Let $\hat{e}: G_q \times G_q \rightarrow G_m$ be an admissible bilinear mapping with the following properties:

- **Bilinearity:** $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_q$ and $a, b \in \mathbb{Z}_q^*$.

- **Non-degeneracy:** There exists $P, Q \in G_q$ such that $\hat{e}(P, Q) \neq 1_m$, where $1_m$ is an identity element in the group $G_m$, where $P$ is a generator of $G_q$ and $\hat{e}(P, P)$ is a generator of $G_m$.

- **Computability:** There is an efficient polynomial time-bounded algorithm that can compute $\hat{e}(P, Q)$ for all $P, Q \in G_q$.

2.2. Computational Problems

**Definition 1.** Collision Attack Assumption 1 (k-CAA1) [17, 18]: For an integer $k$, given $\langle h, P, rP, (h_1, \frac{1}{r+ h_1} P), (h_2, \frac{1}{r+ h_2} P), \ldots, (h_k, \frac{1}{r+ h_k} P) \rangle$, where $P \in G_q$, ...
Choose a general cryptographic hash function. Compute and publish the system’s parameter. For a given security parameter $k \in \mathbb{Z}^+$, this algorithm works as follows:

- Choose a $k$-bit prime number $q$ and determine the tuple $\{G_q, G_m, q, \hat{e}, P\}$, where $\hat{e}$ is an admissible bilinear map and $P$ is the generator of $G_q$.
- Compute $g = \hat{e}(P, P) \in G_m$.
- Choose a general cryptographic hash function $H: \{0,1\}^* \times G_q \rightarrow \mathbb{Z}_q^*$.
- Publish the system’s parameter $\Omega = \{G_q, G_m, q, \hat{e}, g, P, H\}$.

3. Proposed Digital Signature Scheme

In this section, the proposed efficient digital signature scheme using ECC and bilinear pairings is given. It employs the general cryptographic hash function instead of map-to-point function so that our scheme is ease-to-use and becomes computationally efficient. The proposed scheme consists of the following algorithms:

- **Setup:** This algorithm takes a security parameter as input, and returns a list of system’s parameter. For a given security parameter $k \in \mathbb{Z}^+$, this algorithm works as follows:
  
  - Choose a $k$-bit prime number $q$ and determine the tuple $\{G_q, G_m, q, \hat{e}, P\}$, where $\hat{e}$ is an admissible bilinear map and $P$ is the generator of $G_q$.
  
  - Compute $g = \hat{e}(P, P) \in G_m$.
  
  - Choose a general cryptographic hash function $H: \{0,1\}^* \times G_q \rightarrow \mathbb{Z}_q^*$.
  
  - Publish the system’s parameter $\Omega = \{G_q, G_m, q, \hat{e}, g, P, H\}$.
• **KeyGen**: This algorithm takes the system parameter $\Omega$ as input and returns the private key of the user. For the user $ID$, the algorithm select a random number $d_{ID} \in \mathbb{Z}_q^*$ and computes $Q_{ID} = d_{ID}P \in G_q$. The public/private key pair of the user $ID$ is $(d_{ID}, Q_{ID})$.

• **Sign**: To sign a message $m \in \{0,1\}^*$, the signer $ID$ with private key $d_{ID}$ chooses a number $r \in \mathbb{Z}_q^*$ and then computes the signature as follows:

  (a) Compute $R = rP$ and $h = H(m, R)$.
  (b) Compute $V = (r + hd_{ID})^{-1}P$.
  (c) Output the signature $(R, V)$ for the message $m$ and sends it to the verifier for verification.

• **Verify**: To verify the signature $(R, V)$ on a message $m$, the verifier uses the public key $Q_{ID}$ of the signer $ID$ and then performs the following steps:

  (a) Compute $h = H(m, R)$ and $\sigma = \hat{e}(V, R + hQ_{ID})$.
  (b) Checks whether the equation $\sigma = g$ holds. If so, the verifier accepts the signature $(R, V)$; otherwise rejects it.

### 4. Analysis of the Proposed Scheme

In this section, the security and the performance of the proposed scheme are analyzed. As stated earlier, the proposed scheme is existential unforgeability against the adaptive chosen message and identity attacks based on the variation of Collusion Attack Algorithm with $k$ traitors ($k$-CAA3) assumption, which is addressed now.

#### 4.1. Correctness of the Proposed Scheme

Since $Q_{ID} = d_{ID}P, R = rP, h = H(m, R)$ and $V = (r + hd_{ID})^{-1}P$, we have

$$\sigma = \hat{e}(V, R + hQ_{ID})$$
$$= \hat{e}((r + hd_{ID})^{-1}P, rP + hd_{ID}P)$$
$$= \hat{e}((r + hd_{ID})^{-1}P, (r + hd_{ID})P)$$
$$= \hat{e}(P, P) = g$$

Therefore, $\sigma = g$ is satisfied and it proves the correctness of the proposed scheme.
4.2. Security Analysis of the Proposed Scheme

**Theorem (Existential Unforgeability).** If there exists an adaptively chosen message and identity adversary $\mathcal{A}$ who can breach the security of the proposed scheme in polynomial time $t$ with success probability $\varepsilon$, then there exists an algorithm $\mathcal{C}$ that can solve the $k$-CAA3 problem using $\mathcal{A}$ with probability $\varepsilon' \geq \varepsilon \left( \frac{q_S}{q_H} \right)^{q_S}$ and within the time-bound $t' = t$, where $\mathcal{A}$ can make at most $q_S$ queries to the Sign-oracle, $q_V$ queries to the Verify-oracle and $q_H$ queries to the $H$-oracle.

**Proof.** Assume that the forger $\mathcal{A}$ ($\varepsilon$, $t, q_H, q_S, q_V$) tries to breach the security of the proposed scheme. Then we show that there exist a polynomial time-bounded algorithms $\mathcal{C}$ that can solve the $k$-CAA3 problem by using $\mathcal{A}$ as a black box. To solve the $k$-CAA3 problem, $\mathcal{C}$ randomly selects an identity $ID^*$ as the challenged identity, sets the signer’s private/public key as $(d_{ID} = a, Q_{ID} = aP)$ and then sends the system’s parameter $\Omega = \{G, G_m, q, \hat{e}, g, P, H, Q_{ID} = aP\}$ to $\mathcal{A}$, where $a \in R Z_q^*$ is unknown.

- **KeyGen Queries:** If $ID = ID^*$, $\mathcal{C}$ stops the simulation, otherwise $\mathcal{C}$ selects a number $d_{ID} \in R Z_q$, sets $Q_{ID} = d_{ID}P$ and returns $d_{ID}$ as a private key of the user $ID$.

- **Hash Queries to H:** Here, we assume that the output of the $H$-oracle is uniformly distributed over $Z_q^*$ and the $H$-oracle will give the correct answer for any hash query. To replay quickly and to avoid the inconsistency, $\mathcal{C}$ maintains an $H$-oracle list $L_H^{list}$ that contains the tuple of the form $(m, R, h)$. If $\mathcal{A}$ asks a hash query to $H$-oracle on $(m, R)$, $\mathcal{C}$ outputs the previous $h$ if a tuple $(m, R, h)$ is found in $L_H^{list}$, otherwise, chooses a number $h \in R Z_q$, outputs it and then adds the tuple $(m, R, h)$ to the list $L_H^{list}$.

- **Sign Queries:** When $\mathcal{A}$ submits a Sign query on $(ID, m)$, $\mathcal{C}$ then replies as follows:
  (a) If $ID = ID^*$, $\mathcal{C}$ chooses a number $h \in R Z_q$ and computes the signature as follows:
    - Set $R \leftarrow (P - haP)$ and $H(m, R) \leftarrow h$.
    - Set $V \leftarrow P$ and output the signature $(R, V)$.
  (b) If $ID = ID^*$, $\mathcal{C}$ chooses a number $r \in R Z_q$ and computes the signature as follows:
    - Compute $R = rP$ and $h = H(m, R)$.
    - Compute $V = (r + hd_{ID})^{-1}P$ and then output the signature $(R, V)$.
• **Verify Queries:** Upon receiving a Verify query on \((R, V)\) with \((ID, m)\), \(C\) then does as follows:

(a) If \(ID = ID^*\), holds, \(C\) then quits the simulation.

(b) Else, verify the signature \((R, V)\) using the public key \(Q_{ID}\) of \(ID\) according to the proposed Verify algorithm.

• **Forgery:** Finally, \(A\) outputs a valid and forged signature \((R^*, V^*)\) on \((m^*, ID^*)\) if \(ID = ID^*\) holds. Otherwise, \(A\) outputs failure and quits the execution. Since \((R^*, V^*)\) is valid signature, so that \(\hat{e}(V^*, R^* + h^* Q_{ID}) = \hat{e}((r^* + h^* a) - 1 P, (r^* + h^* a) P) = \hat{e}(P, P) = g\). Therefore, \(C\) solves an instance of \(k\)-CAA3 problem.

• **Probability Assessment:** The hash function \(H\) behaves as a random oracle thus, \(A\)'s simulation and the real simulation of the proposed scheme cannot be distinguishable by \(C\). Hence, \(A\)'s execution time is equal to the running time of \(C\) i.e., \(t' = t\). When \(A\) executes a Sign-oracle query for each \(h_i (1 \leq i \leq q_H)\), \(C\) then returns \(V_i = (r + ah_i)^{-1} P\) to \(A\) with probability \(\left(\frac{qs}{q_H}\right)^{q_H}\). Here, each \(h_i (1 \leq i \leq q_H)\) is the answer of the \(H\)-oracle queries on the messages \(m_i\) for \(1 \leq i \leq q_H\). In order to break the security of the proposed signature scheme, \(A\) must output a forged signature \((R^*, V^*)\) on \((ID^*, m^*)\), which gives \(\text{Verify}(ID^*, m^*, Q_{ID}^*, R^*) = 1\), and for which \(A\) must compute \(V^* = (r^* + h^* a)^{-1} P\) with probability \(\left(\frac{qs}{q_H}\right)^{q_H}\), where \(a, h, r, h_i \in \mathbb{Z}_q, 1 \leq i \leq q_H\) but \(h \notin \{h_1, h_2, \cdots, h_{q_H}\}\). Thus, the success probability of the adversary \(A\) to break the security of the proposed scheme is \(\varepsilon' \geq \varepsilon \left(\frac{qs}{q_H}\right)^{q_H}\).

### 4.3. Efficiency Analysis of the Proposed Scheme

In order to evaluate the efficiency of the proposed scheme, we compare different schemes [8–11] with ours in terms of the computation cost. According to [11, 22], the Table 1 includes the various time complexities and their conversion to the time complexity for executing the modular multiplication \(T_{ML}\). The Table 2 compares the proposed scheme with other existing schemes [8–11], and it can be noted that our scheme is computationally efficient than [8–10]. It can be seen that the proposed scheme increases the computational cost slightly with respect to the scheme [11], however, none of the schemes [8–11] are secure as shown in [6, 9, 12] whereas our scheme is existential unforgeable against the adaptive chosen message and identity attacks based on \(k\)-CAA3.
assumption in the random oracle model. Hence, our scheme in all respect is more ef-
cient than other schemes.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Definition and Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ML}$</td>
<td>Time complexity for executing the modular multiplication</td>
</tr>
<tr>
<td>$T_{EX}$</td>
<td>Time complexity for executing the modular exponentiation, $1T_{EX} \approx 240T_{ML}$ [11]</td>
</tr>
<tr>
<td>$T_{EM}$</td>
<td>Time complexity for executing elliptic curve scalar point multiplication, $1T_{EM} \approx 29T_{ML}$ [11]</td>
</tr>
<tr>
<td>$T_{BP}$</td>
<td>Time complexity for executing the bilinear pairing operation, $1T_{BP} \approx 87T_{ML}$ [22]</td>
</tr>
<tr>
<td>$T_{IN}$</td>
<td>Time complexity for executing the modular inversion operation in $\mathbb{Z}<em>q$, $1T</em>{IN} \approx 11.6T_{ML}$ [22]</td>
</tr>
</tbody>
</table>

Tab 1. Definition and conversion of various operation units

<table>
<thead>
<tr>
<th>Protocol/Phases</th>
<th>Signature Generation</th>
<th>Signature Verification</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harn [8]</td>
<td>$2T_{EX}$</td>
<td>$3T_{EX}$</td>
<td>$5T_{EX} \approx 1200T_{ML}$</td>
</tr>
<tr>
<td>Lee-Hwang [9]</td>
<td>$2T_{EX}$</td>
<td>$2T_{EX}$</td>
<td>$4T_{EX} \approx 960T_{ML}$</td>
</tr>
<tr>
<td>Nyang-Song [10]</td>
<td>$2T_{EX}$</td>
<td>$2T_{EX}$</td>
<td>$4T_{EX} \approx 960T_{ML}$</td>
</tr>
<tr>
<td>Chung et al. [11]</td>
<td>$2T_{EM}$</td>
<td>$3T_{EM}$</td>
<td>$5T_{EM} \approx 145T_{ML}$</td>
</tr>
<tr>
<td>Proposed</td>
<td>$2T_{EM}+T_{IN}$</td>
<td>$1T_{EM}+T_{BP}$</td>
<td>$3T_{EM}+T_{IN}+T_{BP} \approx 185T_{ML}$</td>
</tr>
</tbody>
</table>

Tab 2. Performance comparison of the proposed scheme with others

5. Concluding Remarks

In this paper, an efficient and provably-secure digital signature scheme, which is ba-
ded on ECC and bilinear pairings and without using map-to-point function, is designed. We proved that our scheme is existential unforgeable against the adaptive chosen mes-
 sage and identity attacks based on the variation of Collusion Attack Algorithm with $k$ traitors ($k$-CAA3) assumption in the random oracle model. The comparative study signi-
fies that the proposed scheme has low computation cost than almost all previous schemes (except one), however, all are not well secured. Thus, our scheme is efficient and can be applied in real applications.

Acknowledgements

This research was supported by the Department of Science and Technology (DST), Govt. of India, under the INSPIRE fellowship Ph.D program (Reg. No. IF10247) and the Department of Information Technology (DIT), Ministry of Communication and In-
formation Technology, Govt. of India, under the Information Security Education and Awareness (ISEA) program (Project No. MIT(2)/2006–08/189/CSE). The authors wo-
uld like to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian School of Mines, Dhanbad-826004, India for providing their research support, as without such help this work could not be carried out.
References


