An Efficient ID-based Signature Scheme from Pairings

Chunxiang Gu, Yuefei Zhu, and Xiaoyu Pan

Network Engineering Department, Information Engineering University,
P.O. Box 1001-770, Zhengzhou, 450002, P.R.China
gcxiang5209@yahoo.com.cn

Abstract. In this paper, we propose an efficient ID-based signature scheme based on pairing. The number of paring operation involved in the verification procedure is one. Our scheme is proved secure against existential forgery on adaptively chosen message and ID attack under the hardness assumption of computational Diffie-Hellman problem, in the random oracle model.
Keywords: ID-based signatures, pairings, provable security, Forking Lemma.

1 Introduction

In 1984, Shamir [1] first proposed the idea of ID-based public key cryptography (ID-PKC) to simplify key management procedures of traditional certificate-based cryptography. In ID-PKC, the public key of a user is derived directly from his identity information, such as an IP address belonging to a network host, or an e-mail address associated with a user. His private key is generated by a trusted third party called Private Key Generator (PKG). The direct derivation of public keys eliminates the need for certificates and some of the problems associated with them.

Digital signatures are one of the most important security services offered by cryptography. Shamir [1] proposed an ID-based signature scheme based on integer factorization problem. Later, practical solutions for ID-based signature schemes were proposed in [2, 3]. An ID-based signature scheme using pairings was first proposed by Sakai *et.al.* in [4], however they did not present the security analysis in their work. A provably secure ID-based signature scheme was proposed by Cha and Cheon in [5]. They provided a definition of security for ID-based signature schemes called *existential unforgeable under adaptively chosen message and ID attacks* (EUF-ACMIA), which can be view as an extension of the security notion *existential unforgeable under adaptively chosen message attacks* (EUF-ACMA) of non-ID-based signature schemes, and proved their scheme secure in the random oracle model under the hardness assumption of computational Diffie-Hellman problem. An ID-based signature scheme, which is equivalent to [5], was independently proposed by Yi in [6]. Hess [7] proposed an ID-based signature scheme with some advantages in efficiency and

signature length. Cheon *et.al.* [8] proposed an ID-based signature scheme that enables secure batch verification. In Eurocrypt 2004, Bellare *et.al.*, [9] presented a framework to provide security proofs for a large family of ID-based signature schemes constructed from "convertible" identification schemes.

In all these ID-based signature scheme from pairings, the number of paring involved in the verification procedure is at least two. Although fruitful achievements [10, 11] have been made in enhancing the computation of pairings, the computation of pairings are still a heavy burden for verification. In this paper, we propose an efficient ID-based signature scheme based on pairings. The number of pairing involved in the verification procedure is only one. Our scheme is proved to be EUF-ACMIA with the hardness assumption of the computational Diffie-Hellman problem, in the random oracle model.

The rest of this paper is organized as follows: In Section 2, we recall some preliminary works. In Section 3, we present the new scheme and discuss it's implementation issues briefly. In Section 4, we offer security proof for our scheme in the random orale model. Finally, we conclude in Section 5.

2 Preliminary

2.1 Bilinear Pairings

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of order q. $\hat{e} : G_1 \times G_1 \rightarrow G_2$ be a map which satisfies the following properties.

1. Bilinear: $\forall P, Q \in G_1, \forall \alpha, \beta \in Z_q, \hat{e}(\alpha P, \beta Q) = \hat{e}(P, Q)^{\alpha\beta}$;
2. Non-degenerate: If P is a generator of G_1, then $\hat{e}(P, P)$ is a generator of G_2;
3. Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G_1$.

Such an bilinear map is called an *admissible bilinear pairing* [12]. The Weil pairings and the Tate pairings of elliptic curves can be used to construct efficient admissible bilinear pairings.

We review a complexity problem related to bilinear pairings: the computational Diffie-Hellman problem (CDHP). Let P be a generator of G_1, and $a, b \in Z_q$.

CDHP. The CDHP is as follows: given $P, aP, bP \in G_1$, output abP. An algorithm $\mathcal{A}$ solves CDHP with the probability ε if

$$Pr[\mathcal{A}(P, aP, bP) = abP] \geq \varepsilon,$$

where the probability is over the random choice of generator $P \in G_1$, the random choice of $a, b \in Z_q$ and random coins consumed by $\mathcal{A}$. We assume through this paper that CDHP is intractable, which means that there is no polynomial time algorithm to solve CDHP with non-negligible probability.

2.2 Digital Signature Schemes and Forking Lemma

Definition 1. *[13] A digital signature scheme is defined by a triple of polynomial-time algorithms:*

- ***KGen****: The key generation algorithm, takes input a security parameter $\lambda \in N$ (given as 1^λ), returns a pair (pk, sk) of matching public and secret keys.*
- ***Sign****: The signing algorithm, takes input secret key sk and a message m, outputs a signature δ.*
- ***Verify****: The verification algorithm, takes input public key pk, message m and a signature δ, outputs 0 or 1. The later implies δ is a valid signature.*

The advantage in existentially forging of an adversary $\mathcal{F}$, given access to the ***Sign*** oracle $S(.)$ and the hash oracle $H(.)$, is

$$Adv_{\mathcal{F}}(\lambda) = \Pr\left[\begin{array}{c}(pk, sk) \leftarrow KGen(1^\lambda), (m, \delta) \leftarrow \mathcal{F}^{S(.),H(.)}(pk) : \\ Verify((m, \delta), pk) = 1, (m, \delta) \notin S_{list}\end{array}\right],$$

where S_{list} is the query/answer list coming from $S(.)$ during the attack. The probability is taken over the coin tosses of the algorithms, of the oracles, and of the forger.

Definition 2. *A digital signature scheme $\{KGen, Sign, Verify\}$ is said to be EUF-ACMA, if for any adversary $\mathcal{F}$, $Adv_{\mathcal{F}}(\lambda)$ is negligible.*

Pointcheval and Stern [13] discussed for a notion of *generic signature scheme* which, given the input message m, produces a triple (σ_1, h, σ_2), where σ_1 randomly takes its values in a large set, h is the hash value of (m, σ_1) and σ_2 only depends on σ_1, h and the secret key. And each signature is independent of the previous ones. They provided the famous Forking Lemma:

Lemma 1 (Forking Lemma [13]). *In the random oracle model, for a generic signature scheme, let $\mathcal{F}$ be a Turing machine whose input only consists of public data. Assume that $\mathcal{F}$ can produces a valid signature $(m, \sigma_1, h, \sigma_2)$ within a time bound T by un-negligible probability $\varepsilon \geq 10(n_s + 1)(n_h + n_s)/q$, where n_h and n_s are the number of queries that $\mathcal{F}$ can ask to the random oracle and the signing oracle respectively. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from $\mathcal{F}$ replacing the signing oracle by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$ in expected time less than $120686 \cdot n_h \cdot T/\varepsilon$.*

2.3 ID-based Signature Schemes

Definition 3. *[5] An ID-based signature scheme consists of four polynomial-time algorithms:*

- ***Setup*****:** *The parameters generation algorithm, takes as input a security parameter $\lambda \in N$ (given as 1^λ), and returns a master key s and system parameters $para$. This algorithm is performed by PKG. PKG publishes $para$ while keeps s secretly.*
- ***Extract*****:** *The private key generation algorithm, takes as input an identity $ID \in \{0,1\}^*$, extracts the secret key D_{ID}. PKG uses this algorithm to extract the user's secret key D_{ID}, and gives D_{ID} to the user by a secure channel.*
- ***Sign*****:** *The signing algorithm, takes input a private key D_{ID} and a message $m \in \{0,1\}^*$, outputs a signature δ.*
- ***Verify*****:** *The verification algorithm, takes input an identity ID, a message m and a signature δ, and outputs 0 or 1. The later implies δ is a valid signature.*

An ID-based digital signature scheme is said to be EUF-ACMIA, if for any polynomial-time adversary $\mathcal{F}$, the advantage defined by

$$Adv_{\mathcal{F}}(\lambda) = \Pr\left[\begin{array}{c} para \leftarrow Setup(1^\lambda), (ID, m, \delta) \leftarrow \mathcal{F}^{S(.),E(.)}(para) : \\ Verify((m,\delta), ID) = 1, (ID, m, \delta) \notin S_{list}, (ID, .) \notin E_{list} \end{array}\right]$$

is negligible, where S_{list} and E_{list} are the query/answer lists coming from $Sign$ oracle $S(.)$ and $Extract$ oracle $E(.)$ respectively during the attack. In the random oracle model, the attackers also have the ability to query to the random oracles. The probability is taken over the coin tosses of the algorithms, of the oracles, and of the forger.

3 A New ID-based Signature Scheme

3.1 The Scheme

The scheme is described as following:

- ***Setup***: Takes as input a security parameter $\lambda \in N$, outputs a master key s and system parameters $para = (G_1, G_2, q, \hat{e}, P, P_{pub}, \mu, H_1, H_2)$, where $(G_1, +)$ and $(G_2, \cdot)$ are cyclic groups of order q, $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear pairing, $P_{pub} = sP$, $\mu = \hat{e}(P, P)$, $H_1 : \{0,1\}^* \rightarrow Z_q^*$ and $H_2 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$ are hash functions.
- ***Extract***: Takes as input an identity $ID_X \in \{0,1\}^*$, computers $D_X = (H_1(ID_X) + s)^{-1}P$, and lets D_X be the user's secret key. The user's public key can be computed as $Q_X = H_1(ID_X)P + P_{pub}$.
- ***Sign***: For input secret key D_X and a message m, selects randomly $r \in Z_q^*$, computes $\xi = \mu^r$, $h = H_2(m, \xi)$, $U = (r + h)D_X$, and outputs (h, U).
- ***Verify***: For input a message m and signature (h, U) of identity ID_X, the verifier accept the signature if and only if $h = H_2(m, \xi)$, where

$$\xi = \mu^{-h} \cdot \hat{e}(U, H_1(ID_X)P + P_{pub}).$$

We call the ID-based signature scheme IBSS. Consistency of IBSS is easily proved as follows: If $\delta = (h, U)$ is a valid signature of a message m for an identity ID_X, then $\xi = \mu^r$, $h = H_2(m, \xi)$ and $U = (r + h)D_X$ for $r \in Z_q^*$. Thus

$$\begin{aligned}\xi &= \hat{e}(P,P)^{-h} \cdot \mu^r \cdot \hat{e}(P,P)^h \\ &= \mu^{-h} \cdot \hat{e}((H_1(ID_X) + s)^{-1}P, (H_1(ID_X) + s)P)^{(r+h)} \\ &= \mu^{-h} \cdot \hat{e}(D_X, H_1(ID_X)P + P_{pub})^{(r+h)} \\ &= \mu^{-h} \cdot \hat{e}(U, H_1(ID_X)P + P_{pub})\end{aligned}$$

as desired.

3.2 Efficiency

Some general performance enhancements can be applied to our schemes. For pre-selected $P \in G_1$ and $\mu \in G_2$, there are efficient algorithms [14] to compute kP and μ^l for random $k, l \in Z_q$ by pre-computing and storing. We may assume that such computations are at most 1/5 an ordinary scalar multiplication in $(G_1, +)$ and an ordinary exponentiation in $(G_2, .)$. In our scheme, P and μ are fixed system parameters. The signer's private key is fixed for himself. Denote by M an ordinary scalar multiplication in $(G_1, +)$, by E an Exp. operation in $(G_2, .)$, and by $\hat{e}$ a computation of the pairing. We compare IBSS to the most recent schemes [4, 5, 7, 8] which are also based on parings in the following table.

schemes	Sign	Verify	Signature domain
Sakai-Ohgishi-Kasahara[4]	$1.2M$	$3\hat{e}$	$G_1 \times G_1$
Cha-Cheon[5]	$0.4M$	$2\hat{e} + 1M$	$G_1 \times G_1$
Hess[7]	$0.2E + 0.4M$	$1E + 2\hat{e}$	$G_1 \times Z_q^*$
Cheon-Kim-Yoon[8]	$0.6M$	$2\hat{e} + 0.2M$	$G_1 \times G_1$
IBSS	$0.2E + 0.2M$	$0.2M + 0.2E + 1\hat{e}$	$G_1 \times Z_q^*$

On the other hand, the new scheme introduce $\mu = \hat{e}(P, P)$ to the system parameter. And users' secret keys are in the form $(H_1(ID)P + s)^{-1}P$, which is different from some existed ID-based schemes including the Boneh-Franklin's ID-based encryption scheme [12].

4 Security Proof

We first define a related public key signature scheme (not an ID-based scheme) $PKSS = (KGen, Sign', Verify')$ as following:

- ***KGen***: Takes as input a security parameter $\lambda \in N$,
  1. runs $Setup(1^\lambda)$ of IBSS to generate a master key s and system parameters $para = (G_1, G_2, q, \hat{e}, P, P_{pub}, \mu, H_1, H_2)$,
  2. selects randomly $ID \in \{0,1\}^*$, computes $Q = H_1(ID)P + P_{pub}$, $D = (H_1(ID) + s)^{-1}P$,
  3. returns D as private key and $(para, Q)$ as public key.

- ***Sign'***: For input secret key D and a message m, selects randomly $r \in Z_q^*$, computes $\xi = \mu^r$, $h = H_2(m, \xi)$, $U = (r + h)D_X$, and outputs (h, U) as the signature.
- ***Verify'***: For input a message m and a signature (h, U) of public key $(para, Q)$, the verifier computes $\xi = \mu^{-h} \cdot \hat{e}(U, Q)$, and accept the signature if and only if $h = H_2(m, \xi)$.

It is obviously that the PKSS is a *generic signature scheme*. If we assume that $H_1 : \{0,1\}^* \to Z_q^*$ is a random function, then the function H defined by

$$\begin{aligned} H : \{0,1\}^* &\to Z_q^* \\ ID_X &\mapsto H_1(ID_X)P + P_{pub}. \end{aligned}$$

is a random function. In the security proof, we take $H(.)$ as a random oracle.

Lemma 2. *In the random oracle mode, if there is an adversary $\mathcal{F}_0$ whose input only consists of public data, and can succeed in existential forgery against IBSS within a time bound T by un-negligible probability ε, then there is another adversary $\mathcal{F}_1$ who can succeed in existential forgery against PKSS, within expected time T with un-negligible probability ε/n_h, where n_h is the number of queries that $\mathcal{F}_0$ can ask to the random oracle $H(.)$.*

Proof: Without any loss of generality, we may assume that for any ID, $\mathcal{F}_0$ queries $H(.)$ with ID before ID is used as (part of) an input of any query to $Extract(.)$ and $Sign(.)$.

From $\mathcal{F}_0$, we can construct adversary $\mathcal{F}_1$ against PKSS as follows:

1. A challenger $\mathcal{C}$ runs $(D, (para, Q)) \leftarrow KGen(1^\lambda)$, where $para = (G_1, G_2, q, \hat{e}, P, P_{pub}, \mu, H_1, H_2)$, and gives $(para, Q)$ to $\mathcal{F}_1$.
2. $\mathcal{F}_1$ sets $z = 1$, picks randomly $t, 1 \le t \le n_h$ and $x_i \in Z_q, i = 1, 2, ...n_h$.
3. $\mathcal{F}_1$ runs $\mathcal{F}_0$ with input $para$. During the execution, $\mathcal{F}_1$ emulates $\mathcal{F}_0$'s oracles as follows:
   - $H(.)$: For input ID, $\mathcal{F}_1$ checks if $H(ID)$ is defined. If not, he defines $H(ID) = \begin{cases} Q & z = t \\ x_i P & z \neq t \end{cases}$, and set $ID_z \leftarrow ID, z \leftarrow z+1$. $\mathcal{F}_1$ returns $H(ID)$ to $\mathcal{F}_0$.
   - $H_2(.)$: If $\mathcal{F}_0$ makes a query (m, ξ) to random oracle $H_2(.)$, $\mathcal{F}_1$ checks if $H_2(m, \xi)$ is defined. If not, it picks a random $c \in Z_q$, and sets $H_2(m, \xi) \leftarrow c$. $\mathcal{F}_1$ returns $H_2(m, \xi)$ to $\mathcal{F}_0$.
   - $Extract(.)$: For input ID_i, if $i = t$, then abort. Otherwise, $\mathcal{F}_1$ computes $D_i = x_i^{-1} \cdot P$ and lets D_i be the reply to $\mathcal{F}_0$.
   - $Sign(.)$: For input ID_i and message m, if $i \neq t$, $\mathcal{F}_1$ uses $D_i = x_i^{-1} P$ as the private key to sign on m. Otherwise, $\mathcal{F}_1$ simulates ID_t's signing oracle with his own signing oracle $Sign'(.)$.
4. If $\mathcal{F}_0$'s output is (ID_i, m, h, U) satisfying: $Verify(ID_i, m, (h, U)) = 1$ and $i = t$, then $\mathcal{F}_1$ can get a forgery $(m, (h, U))$ of **PKSS** corresponding to $(para, Q)$.

If $\mathcal{F}_0$ succeed in his attack, then $\mathcal{F}_0$ has not query to $Extract(.)$ with input ID_t. Hence the responses of $\mathcal{F}_1$'s emulations are indistinguishable from $\mathcal{F}_0$'s real oracles. Because t is chosen randomly, $\mathcal{F}_1$ can output a forgery corresponding to $(para, Q)$ of **PKSS** within expected time T with probability ε/n_h.

We now prove that the signature triples (ξ, h, U) of PKSS can be simulated without the knowledge of the signer's secret key, with an indistinguishable distribution probability.

Lemma 3. *Given* $(G_1, G_2, q, \hat{e}, P, P_{pub} = sP, \mu = \hat{e}(P, P), H_1, H_2)$ *and an identity* ID, $Q = H_1(ID)P + P_{pub}$, $D = (H_1(ID) + s)^{-1}P$, *the following distributions are the same.*

$$\delta = \left\{ (\xi, h, U) \left| \begin{array}{c} r \in_R Z_q^* \\ h \in_R Z_q \\ \xi = \mu^r \\ U = (h + r) \cdot D \end{array} \right. \right\} \text{ and } \delta' = \left\{ (\xi, h, U) \left| \begin{array}{c} U \in_R G_1 \\ h \in_R Z_q \\ \xi = \mu^{-h} \cdot \hat{e}(U, Q) \\ \xi \neq 1 \end{array} \right. \right\}$$

Proof: First we choose a triple (α, β, γ) from the set of the signatures: $\alpha \in G_2^*$, $\beta \in Z_q$, $\gamma \in G_1$ such that $\alpha = \mu^{-\beta} \cdot \hat{e}(\gamma, Q) \neq 1$. We then compute the probability of appearance of this triple following each distribution of probabilities:

$$\Pr_{\delta}\left[(\xi, h, U) = (\alpha, \beta, \gamma)\right] = \Pr_{r \neq 0}\left[\begin{array}{l} \mu^r = \alpha \\ h = \beta \\ (h + r) \cdot D = \gamma \end{array}\right] = \frac{1}{q(q-1)}.$$

$$\Pr_{\delta'}\left[(\xi, c, U) = (\alpha, \beta, \gamma)\right] = \Pr_{\xi \neq 1}\left[\begin{array}{l} \alpha = \xi = \mu^{-h} \cdot \hat{e}(U, Q) \\ h = \beta \\ U = \gamma \end{array}\right] = \frac{1}{q(q-1)}.$$

That is, we can construct a simulator $\mathcal{M}$, which produces triples (ξ, h, U) with an identical distribution from those produced by the signer, as follows.

- **Simulator** $\mathcal{M}$: For input $(G_1, G_2, q, \hat{e}, P, P_{pub}, \mu, H_1, H_2)$ and $Q = H_1(ID)P + P_{pub}$ and a message m,
  1. randomly chooses $U \in G_1$ and $h \in Z_q$, and computes $\xi = \mu^{-h} \cdot \hat{e}(U, Q)$. In the (unlikely) situation where $\xi = 1$, we discard the results and restart the simulation.
  2. returns the triple (ξ, h, U).

Theorem 1. *In the random oracle model, if there is an adversary* $\mathcal{F}_0$ *who performs, within a time bound* T, *an existential forgery against IBSS with probability* $\varepsilon \geq 10 n_h (n_s + 1)(n_{h_2} + n_s)/q$, *where* n_h, n_{h_2} *and* n_s *are the number of queries that* $\mathcal{F}_0$ *can ask to the oracles* $H(.)$, $H_2(.)$ *and* $Sign(.)$ *respectively. Then there is a Turing machine* $\mathcal{M}_1$ *that can output* $a^{-1}P$ *on input of any given* $P, aP \in G_1^*$ *within expected time less than* $120686 \cdot n_h \cdot n_{h_2} \cdot T/\varepsilon$.

Proof: With the Lemma 2, using adversary $\mathcal{F}_0$, we can construct another adversary $\mathcal{F}_1$, given $(para, Q)$, who can produce a valid signature of PKSS, within expected time T with un-negligible probability $\varepsilon' = \varepsilon/n_h$. PKSS is a generic signature scheme and the signature triples (ξ, h, U) can be simulated without the knowledge of the signer's secret key, with an indistinguishable distribution probability (proved in Lemma 3). $\varepsilon' = \varepsilon/n_h \geq 10(n_s+1)(n_{h_2}+n_s)/q$. Hence, with the Forking Lemma (described in Lemma 1), there is another machine $\mathcal{F}_2$ which has control over the machine obtained from $\mathcal{F}_1$ replacing the signing oracle by simulation and produces two valid signatures (m, ξ, h, U) and (m, ξ, h', U) such that $h \neq h'$ in expected time less than $120686 \cdot n_{h_2} \cdot T/\varepsilon' = 120686 \cdot n_h \cdot n_{h_2} \cdot T/\varepsilon$.

From the adversary $\mathcal{F}_2$, we can construct a Turing machine $\mathcal{M}_1$ such that $\mathcal{M}_1$ can output $a^{-1}P$ on input of any given $P, aP \in G_1^*$ as follows:

1. A challenger $\mathcal{C}$ generates $(G_1, G_2, q, \hat{e})$ and selects randomly $P, aP \in G_1$. $\mathcal{C}$ gives $(G_1, G_2, q, \hat{e}, P, aP)$ to $\mathcal{M}_1$ as inputs.
2. $\mathcal{M}_1$ selects $H_1 : \{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$ as hash functions, selects randomly $s \in Z_q^*$ sets $P_{pub} = sP$.
3. $\mathcal{M}_1$ runs $\mathcal{F}_2$ with input $(para = (G_1, G_2, q, \hat{e}, P, P_{pub}, \hat{e}(P, P), H_1, H_2), aP)$ until $\mathcal{F}_2$ outputs two valid signatures (m, ξ, h, U) and (m, ξ, h', U') such that $h \neq h'$.
4. $\mathcal{M}_1$ can computes and outputs $a^{-1}P$ as follows:

$$a^{-1}P = (h - h')^{-1}(U - U')$$

Theorem 2. *Suppose there is a Turing machine $\mathcal{M}_1$ that can output $a^{-1}P$ on input of any given $P, aP \in G_1^*$ with probability ε, in expected time bound T. Then there is a Turing machine $\mathcal{M}_2$ which outputs abP on input of any given $P, aP, bP \in G_1^*$ with probability ε^3 in expected time $3T$.*

Proof. From $\mathcal{M}_1$, we can construct Turing machine $\mathcal{M}_2^*$ as follows:

1. $\mathcal{M}_2$'s input is $P, aP, bP \in G_1^*$.
2. $\mathcal{M}_2$ runs $\mathcal{M}_1$ with input aP, P (P can be used as $a^{-1}aP$). If $\mathcal{M}_1$ outputs $Y_1 = aaP = a^2P$, then goto the next step.
3. $\mathcal{M}_2$ runs $\mathcal{M}_1$ with input $bP, P = b^{-1}bP$. If $\mathcal{M}_1$ outputs $Y_2 = bbP = b^2P$, then goto the next step.
4. $\mathcal{M}_2$ runs $\mathcal{M}_1$ with input $(a+b)P, P = (a+b)^{-1}(a+b)P$. If $\mathcal{M}_1$ outputs $Y_3 = (a+b)^2P$, then goto the next step.
5. $\mathcal{M}_2$ computes and outputs $Y = 2^{-1}(Y_3 - Y_1 - Y_2)$.

Obviously, $Y = 2^{-1}(Y_3 - Y_1 - Y_2) = 2^{-1}(2abP) = abP$. Hence, in expected time $3T$, $\mathcal{M}_2$ can output abP with success probability ε^3.

With Theorem 1 and Theorem 2, we can get the conclusion that the new ID-based signature scheme is EUF-ACMIA under the hardness assumption of CDHP in the random oracle model.

5 Conclusion

In this paper, we proposed an ID-based signature scheme based on pairing whose verification procedure involves only one paring operation. Our scheme is proved EUF-ACMIA in the random oracle model, with the hardness assumption of the CDHP. The new scheme introduces new parameter $\mu = \hat{e}(P, P)$ to the system parameter and users' secret keys are in the form $(H_1(ID)P + s)^{-1}P$, which is different from some existed ID-based schemes including the Boneh-Franklin's ID-based encryption scheme [12].

References

1. A. Shamir. Identity-based cryptosystems and signature schemes. In Advances in Cryptology - CRYPTO'84, LNCS 0196, pp. 47-53. Springer-Verlag, 1984.
2. U. Fiege, A. Fiat and A. Shamir. Zero-knowledge proofs of identity, Journal of Cryptology, Vol.1, pp. 77-94, Springer-Verlag, 1988.
3. A. Fiat and A. Shamir, How to prove yourself: Practical Solutions to identification and signature problems, In Proc. of Crypto'86, LNCS 0263, pp. 186-194, Springer-Verlag, 1987.
4. R. Sakai, K. Ohgishi, and M. Kasahara, Cryptosystems based on pairing. In Proc. of Symposium on Cryptography and Information Security SCIS'2000, Okinawa, Japan, Jan. 2000.
5. J.C. Cha and J.H. Cheon. An identity-based signature from gap Diffie-Hellman groups. In Y. Desmedt, editor, Public Key Cryptography - PKC 2003, LNCS 2567, pages 18-30. Springer-Verlag, 2003.
6. X. Yi, An identity-based signature scheme from the Weil pairing. IEEE Communications Letters 7(2) (2003), pp. 76–78.
7. F. Hess. Efficient identity based signature schemes based on pairings. In K. Nyberg and H. Heys, editors, Selected Areas in Cryptography 9th Annual International Workshop, SAC 2002, LNCS 2595, pp. 310-324. Springer-Verlag, 2003.
8. J. Cheon, Y. Kim, and H. Yoon. Batch Verifications with ID-based Signatures. In: Information Security and Cryptology - ICISC 2004, LNCS 3506, pp.233-248. Springer-Verlag, 2005.
9. M.Bellare, C.Namprempre, and G.Neven. Security Proofs for Identity-Based Identification and Signature Schemes. In: EUROCRYPT 2004, LNCS 3027, pp. 268-286. Springer-Verlag, 2004.
10. P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In Advances in Cryptology C Crypto'2002, LNCS 2442, pp. 354 C 368. Springer-Verlag, 2002.
11. I. Duursma and H. Lee. Tate pairing implementation for hyperelliptic curves $y^2 = x^p + x + d$. In Advances in Cryptology C Asiacrypt'2003, LNCS 2894, pp. 111 C 123. Springer-Verlag, 2003.
12. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, Advances in Cryptology- CRYPTO 2001, LNCS 2139, pp. 213-229. Springer-Verlag, 2001.
13. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 13(3):361-369,2000.
14. Y. Sakai, K. Sakurai. Efficient Scalar Multiplications on Elliptic Curves without Repeated Doublings and Their Practical Performance. ACISP 2000, LNCS 1841, pp. 59-73. Springer-Verlag 2000.