EFFICIENT COMPUTATION OF THE TATE PAIRING ON HYPERELLIPTIC CURVES FOR CRYPTOSYSTEMS

YOUNGJU CHOIE, JAEMYUNG KIM, AND EUNJEONG LEE

Abstract. In this paper, we suggest to use the curve $H_b : y^2 + y = x^5 + x^3 + b, b = 0$ or 1 over $\mathbb{F}_{2^n}$ for a secure and efficient pairing-based cryptosystems. For this curve, we develop efficient algorithms to compute the Tate pairing and give an implementation result of Tate paring on the curve H_0.

1. Introduction

Since the discovery of an identity-based encryption scheme based on the Weil pairing on supersingular elliptic curves, pairing-based cryptography has become one of the most active research fields ([1, 2, 4, 5, 7, 9, 13]). Weil pairing is a quotient of the output of two applications of the Tate pairing, except that the Tate pairing needs an exponentiation. So it is now accepted that the Tate pairing is preferable for its efficiency.

The Tate pairing on a curve C defined over $\mathbb{F}_q$ maps a pair of divisors to a related extension field $\mathbb{F}_{q^k}^*$ for appropriate integer k. Although the Tate pairing can be computed by an algorithm suggested by Miller [17], in practice, it is often the bottleneck in pairing-based systems. In addition, the Miller algorithm on hyperelliptic curves consists of divisor operations which are more complicated than point operations on elliptic curves. From this reason, it was pointed out that hyperelliptic curve cryptosystem(HEC) is not efficient[21]. However, it was also claimed that HEC can be efficient by giving the explicit formulae for group operation on the Jacobian (see [15]).

The efficiency and security of pairing-based cryptosystems mostly depend on the field size q and the extension degree k. For a curve of genus g, the required space for the keys is $g \times |q|$ bits where $|q|$ is the number of bits of q. The security relies not only on the key size but also the extension field size q^k[10]. Here, k is called *security multiplier*. For efficient and secure systems, we consider the following value.

Definition 1.1. *If a pairing-based cryptosystem implemented on a curve C, then define*

$$\epsilon_C := \frac{\log q^k}{\log q^g} = \frac{\textit{security level}}{\textit{space of a key}}.$$

We call ϵ_C the *efficiency factor*. Since the key size determines the size of the computation unit, the larger ϵ_C can provide with the more secure and more efficient systems. For secure system, q^k should be large to make the discrete log problem hard in both the Jacobian group over $\mathbb{F}_q$ and in the finite field $\mathbb{F}_{q^k}$. However, k

This work has been partially supported by ITRC research fund.

should not be too large because the pairing-based cryptosystems adopt computations on the field $\mathbb{F}_{q^k}$.

It is of interest to produce families of curves for which this *security multiplier* k is not too large, but not too small. To obtain a curve which satisfies an appropriate *security multiplier*, supersingular abelian varieties have considered as a suitable setting for pairing-based systems [19].

It is known that the security multiplier of the supersingular elliptic curves is bounded by 6 and the following elliptic curve

$$E : y^2 = x^3 - x + d, d = \pm 1 \text{ over } \mathbb{F}_{3^n} \tag{1}$$

has the maximal 6 and thus $\epsilon_E = 6$. Note that the security multiplier of supersingular hyperelliptic curves of genus 2 over even characteristic is bounded by 12[11]. If they are defined over odd characteristic fields, then the maximal security multiplier is 6 [19] and thus the efficiency factor is 3, which is not the best choice in terms of the efficiency factor. The following hyperelliptic curve

$$H_b : y^2 + y = x^5 + x^3 + b, \quad b = 0 \text{ or } 1 \text{ over } \mathbb{F}_{2^n} \tag{2}$$

has the maximal 12 and thus $\epsilon_{H_b} = 6$. So far, for the supersingular curves C, elliptic or hyperelliptic, $\epsilon_C = 6$ has been the best efficiency factor for the pairing-based cryptosystems.

Up to date, efficient algorithms and implementations of the Tate pairing were provided on the elliptic curve (1)(see [1], [13]). And a closed formula for the Tate pairing on $y^2 = x^p - x + b$ was given in characteristic p, $p \equiv 3 \pmod 4$ with the security multiplier $2p$(see [9]). However, for cryptographic purpose, p can be chosen only 3 or 7 due to the subexponential algorithm of the discrete logarithm problem on the Jacobian of hyperelliptic curves with $g = \frac{p-1}{2} > 3$ [12]. If p is 7, it is the hyperelliptic curve with genus 3. In this case, the efficiency factor for this curve is 14/3 which is less than the best value 6. The Tate pairing was also implemented on hyperelliptic curves over large prime fields [8] which has only 2 as the efficiency factor. Therefore, the best candidates for the pairing-based cryptosystem are the curves (1) and (2). We suggest to use the curve (2) for a secure and efficient pairing-based cryptosystems since most common cryptosystems have based on binary fields.

In this paper, we present efficient algorithms for the Tate pairing on two hyperelliptic curves (2) and give an implementation result for the Tate pairing on H_0. Furthermore, we showed the *compressed pairing* suggested by Barreto et al in [3] can be defined on the curve H_b defined over $\mathbb{F}_{2^n}$, which was explored only in the case of odd characteristics. By compressing, one can efficiently reduce the bandwidth occupied by pairing values without impairing security nor processing time.

In Section 2, we recall the several definitions and basic properties of hyperelliptic curves, divisors and the Tate pairing. We explain how to choose a cryptographically useful curve in detail in Section 3. We give explicit formulae for divisor operations of the hyperelliptic curve $H_b : y^2 + y = x^5 + x^3 + b$ in Section 4. In Section 5, we compare the implementation results with that in [8]. Finally we summarize our results and state open questions regarding fast computation of the Tate pairing.

We want to point out that a similar work has been posted on the preprint archive, http://eprint.iacr.org/2004/375 by and our paper is completely independent from the work.

2. Preliminaries

In this section, we recall the basic definitions and properties(see [14] for further details). Let $\mathbb{F}_q$ be a finite field with q elements and $\bar{\mathbb{F}}_q$ be the algebraic closure of $\mathbb{F}_q$. Hyperelliptic curves defined over $\mathbb{F}_q$ are algebraic curves with genus g which are described the following equation;

$$H/\mathbb{F}_q : y^2 + h(x)y = F(x), \tag{3}$$

where $F(x) \in \mathbb{F}_q[x]$ is a monic polynomial with $\deg(F) = 2g+1$, $h(x) \in \mathbb{F}_q[x], \deg(h) \leq g$ and there are no singular points on H.

Now let

$$H = \{(a, b) \in \bar{\mathbb{F}}_q \times \bar{\mathbb{F}}_q \mid b^2 + h(a)b = F(a)\} \cup \{\mathcal{O}\} \tag{4}$$

and let $H(\mathbb{F}_q) = H \cap (\mathbb{F}_q \times \mathbb{F}_q)$ be a set of rational points on H with the infinite point $\mathcal{O}$.

2.1. Divisors.

A *divisor* D is a formal sum of points on the curve H

$$D = \sum_{P \in H} n_P P$$

where n_P is an integer and $n_P = 0$ for almost all points $P \in H$. If K is the function field defined by (3) then the set of all divisors, denoted by $\mathrm{Div}(K)$, forms a free abelian group.

For a divisor $D = \sum_{P\in H} n_P P$, the *degree* of a divisor D is $\deg(D) = \sum_{P\in H} n_P$ and the *support* of D is $supp(D) = \{P \mid n_P \neq 0\}$. The greatest common divisor of $D_1 = \sum_{P\in H} m_P P$ and $D_2 = \sum_{P\in H} n_P P$ in $\mathrm{Div}(K)$ is

$$\text{g.c.d.}(D_1, D_2) = \sum_{P\in H} min(m_P, n_P)P - (\sum_{P\in H} min(m_P, n_P))\mathcal{O}.$$

Let's consider a subgroup

$$\mathrm{Div}_0(K) = \{D \in \mathrm{Div}(K) \mid \deg(D) = 0\}$$

which is called a group of zero divisors. It is well-known that the set of principle divisors

$$\mathbb{P}_H = \{\mathrm{div}(g) \mid \mathrm{div}(g) = \sum_{P\in H} v_P(g)P,\ g \in K\},$$

where v is a valuation map from K to $\mathbb{Z}$, forms a subgroup of $\mathrm{Div}_0(K)$. Two divisors D_1 and $D_2 \in \mathrm{Div}_0$ are said to be equivalent, $D_1 \sim D_2$, if $D_1 = D_2 + \mathrm{div}(f)$ for some $f \in K^*$. The set of equivalence classes

$$J_H = \mathrm{Div}_0(K)/\mathbb{P}_H$$

forms a divisor class group which is called the *Jacobian* of H.

Note that each divisor class can be uniquely represented by the *reduced divisor* using the Mumford representation [18]. For the curve H, a reduced divisor is summarized as follows;

Theorem 2.1 (**Reduced divisor** [18], [14]). *Let K be the function field given by H defined over $\mathbb{F}_q$.*

(1) *Then each nontrivial divisor class of J_H can be represented by*

$$D = \sum_{i=1}^{r} P_i - r\mathcal{O},\ \textit{where } r \leq g,\ P_i \neq \mathcal{O},\ P_i \in H.$$

(2) *Put $P_i = (a_i,\ b_i)$, $1 \leq i \leq r$. Let $u_D(x) = \prod_{i=1}^{r}(x - a_i)$. Then there exists a unique polynomial $v_D(x) \in \bar{\mathbb{F}}_q[x]$ satisfying*

1) $\deg(v_D) < \deg(u_D) \leq g$
2) $b_i = v_D(a_i)$
3) $u_D(x) \mid v_D(x)^2 + v_D(x)h(x) - F(x)$.

Then $D = \text{g.c.d.}(\text{div}(u_D(x)),\ \text{div}(v_D(x) + y))$.

We will denote a divisor class as $D = [u_D, v_D]$, where D is a reduced divisor and u_D, v_D are polynomials satisfying the three conditions in Theorem 2.1.

2.2. **Tate pairing.** Now we recall the definition of the Tate pairing(see [10] for further details). Let ℓ be a positive integer with $\gcd(\ell,\ q) = 1$ and k be the smallest integer such that $\ell \mid (q^k - 1)$ which is called *the security multiplier*. Let $J_H[\ell] = \{D \in J_H \mid \ell D = \mathcal{O}\}$. The Tate pairing is a map

$$
\begin{aligned}
t : J_H[\ell] \times J_H(\mathbb{F}_{q^k})/\ell J_H(\mathbb{F}_{q^k}) &\rightarrow \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^\ell \\
t(D, E) &= f_D(E') \qquad (5)
\end{aligned}
$$

where $\text{div}(f_D) = \ell D$ and $E' \sim E$ with $supp(E') \cap supp(\text{div}(f_D)) = \emptyset$.

It's well-known that the Tate pairing satisfies the following three properties(see also [10]);

- (non-degeneracy) For each $D \in J_H[\ell] - \{\mathcal{O}\}$ there exits $E \in J_H(\mathbb{F}_{q^k})$ such that $t(D,\ E) \notin (\mathbb{F}^*_{q^k})^\ell$ (and vice versa).
- (bilinearity) For any integer m, $t(mD,\ E) = t(D,\ mE) = t(D,\ E)^m$ in $\mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^\ell$.
- (computability) For any two divisors D and E, the Tate pairing $t(D,\ E)$ can be computed in polynomial time of $k \log q$.

3. Choice of cryptographically strong curves

In this section, we describe how to select a good hyperelliptic curve for cryptosystem and explain the properties of the hyperelliptic curve $H_b : y^2 + y = x^5 + x^3 + b$ defined over $\mathbb{F}_{2^n}$ where $b = 0$ or 1.

Definition 3.1 (HCDLP)**.** *Let H be a hyperelliptic curve defined over $\mathbb{F}_q$ and D_1 and D_2 be reduced divisors in $J_H(\mathbb{F}_q)$. Then the hyperelliptic curve discrete logarithm problem(HCDLP) is defined as follows:*

Determine a positive integer n such that $D_2 = nD_1$ if such an integer exists.

Menezes, Okamoto and Vanstone proposed a subexponential time algorithm to solve the elliptic curve discrete logarithm problem(ECDLP) over a supersingular elliptic curve E defined over a finite field $\mathbb{F}_q$ [16]. It uses the Weil pairing to reduce the ECDLP to the discrete logarithm problem in finite field. On the other hand, Frey and R*ü*ck suggested an algorithm to solve the discrete logarithm problem over the divisor class group using the Tate pairing [10]. We call this algorithm the F-R algorithm.

Algorithm 3.2 (F-R algorithm [10])**.**

Input: *Large prime number $\ell \mid \#J_H(\mathbb{F}_q)$ and $D_1,\ D_2 \in J_H(\mathbb{F}_q)[\ell]$.*
Output: *An integer n $(0 \leq n < \ell)$ such that $D_2 = nD_1$.*
Step1: *Determine the smallest integer k such that $\ell \mid q^k - 1$.*
Step2: *Find $E \in J_H(\mathbb{F}_{q^k})/\ell J_H(\mathbb{F}_{q^k})$ such that $t(D_1,\ E)$ has order ℓ.*
Step3: *Compute n' such that $t(D_1,\ E)^{n'} = t(D_2,\ E)$ in $\mathbb{F}^*_{q^k}$. Then $n \equiv n'$ $(\text{mod } \ell)$.*

TABLE 1. A large prime factor of $\#J_{H_0}(\mathbb{F}_q)$

q	a large prime factor of $\#J_{H_0}(\mathbb{F}_q)$ where $H_0 : y^2 + y = x^5 + x^3$	bits
2^{89}	14851642607221752942766012585821135190909	134
2^{103}	6395375588121100883440814657083560825282870457413014051377	193
2^{113}	8532224489137138306160059160077540585447813491609487653073	193

In Algorithm 3.2, it takes a divisor E in $J_H(\mathbb{F}_{q^k})$ for the nontrivial value of the Tate pairing. The following Lemma 3.3 explains why one needs such a divisor.

Lemma 3.3 ([8]). *Let H be a hyperelliptic curve of genus 2 defined over $\mathbb{F}_q$ and ℓ be a factor of $\#J_H(\mathbb{F}_q)$ with $\gcd(\ell, q) = \gcd(\ell,\ q-1) = 1$. Then $f(E) \in (\mathbb{F}_{q^k}^*)^\ell$ for a rational function $f \in \mathbb{F}_q(H)$ and any divisor $E \in J_H(\mathbb{F}_q)$ such that $\mathrm{supp}(E) \cap \mathrm{supp}(\mathrm{div}(f)) = \emptyset$.*

Proof. See Lemma 3 in [8]. $\square$

In general, there is no known deterministic method to find divisors $D,\ E$ to get a nontrivial value of $t(D,\ E)$, *i.e*, $t(D,\ E) \notin (\mathbb{F}_{q^k}^*)^\ell$. However, one can obtain such divisors using distortion map in case when H is a supersingular curve [22].

If k is the smallest integer such that $\ell \mid \#J_H(\mathbb{F}_q)$, then F-R algorithm tells us that hyperelliptic curve H offers no more security than a discrete logarithm problem in $\mathbb{F}_{q^k}$. Hence the security multiplier k is necessary to be large to keep high security in cryptographic applications. On the other hand, k should not be too large for the computational efficiency because pairing-based protocols require the computations on the extended fields such as $t(D,\ E)^m$ for some integer m. To obtain an appropriate *security multiplier* k, the use of supersingular curves has been suggested. In elliptic curves, the security multiplier is bounded by 6 and that curves are defined in characteristic 3. But the security multiplier can be 12 on the hyperelliptic curves of genus 2 which are defined in characteristic 2 [11].

Recall that the binary field is the most commonly used field in the cryptosystems. In this viewpoint, we suggest the use of

$$H_b : y^2 + y = x^5 + x^3 + b, \quad b = 0 \text{ or } 1 \tag{6}$$

which is defined over a binary field. Note that these curves have the efficiency factor 6 as discussed in the Section 1. Furthermore, the divisor class group of the curve H_b has a good group structure. To determine ℓ, we need to know the orders of $J_{H_b}(\mathbb{F}_{2^n})$, and they are given as follows.

Theorem 3.4 ([23]). *Let $\gcd(n, 6) = 1$. For the curve H_b, we have*

$$\#J_{H_0}(\mathbb{F}_{2^n}) = 2^{2n} + 2^n + 1 + (-1)^{[(n+1)/4]} 2^{(n+1)/2}(2^n + 1), \tag{7}$$

$$\#J_{H_1}(\mathbb{F}_{2^n}) = 2^{2n} + 2^n + 1 - (-1)^{[(n+1)/4]} 2^{(n+1)/2}(2^n + 1), \tag{8}$$

where $[\ ]$ denotes the floor function value, and $J_{H_b}(\mathbb{F}_{2^n})$ is a cyclic group.

The following Table 1 lists large prime factors of $\#J_{H_0}(\mathbb{F}_q)$ when $q = 2^{89},\ 2^{103},\ 2^{113}$ for $H_0 : y^2 + y = x^5 + x^3$. Note that there exists an optimal normal basis of $\mathbb{F}_q$ for $q = 2^{89},\ 2^{113}$. We implement the Tate pairing on this curve H_0 in Section 5.

4. Efficient operations on J_{H_b}

Cantor [6] introduced an algorithm for the divisor operations in the Jacobian of hyperelliptic curves and Miller [17] described a method to compute a rational function which comes from divisor operations. The Tate pairing can be computed by the repetition of the Miller algorithm. In [8], using the Miller algorithm, the rational functions are explicitly given according to the cases of divisors for the general hyperelliptic curves with genus 2.

Here, since we work on the specific curves, $H_b : y^2 + y = x^5 + x^3 + b$, the formulae can be obtained directly from the Cantor algorithm. Especially, it turns out that doubling of a divisor on the curve is computationally very simple, of which complexity is almost that same as doubling of a point on elliptic curves.

For pairing based cryptosystems, we have to compute nD where D is a generator of some cyclic group and n is an integer. If a pairing is defined on elliptic curves then divisors have one-to-one correspondence to points on the curves. However, for hyperelliptic curves, divisors should be expressed not by points but Mumford representation as Theorem 2.1. Therefore we describe the operation formulae in terms of polynomials u, v in K explained in Theorem 2.1.

Lemma 4.1. *Let's denote reduced divisors in J_{H_b} by $D_i = [u_i, v_i]$ for $i = 1,\ 2$ such that $D_2 + \mathrm{div}(f) = 2D_1$. Assume $\deg u_1 \neq 0$. Then*

(1) *If $u_1 = x + u_{10}$, then*

$$u_{21} = 0,\ u_{20} = u_{10}^2,\ v_{21} = (u_{10}^2 + u_{10})^2,\ v_{20} = v_{10}^2 \tag{9}$$

(2) *If $\deg u_1 = 2$, the formula for D_2 and f are described in* Table 2.

Note that when $u_1 = x + u_{10}$, no multiplication is needed for D_2.

Table 2. Doubling when $\deg u_1 = 2$

Input	$\bar{D_1} = [u_1, v_1]$ where $u_1 = x^2 + u_{11}x + u_{10}, v_1 = v_{11}x + v_{10}, F = x^5 + x^3 + b$
Output	$\bar{D_2} = [u_2, v_2], l(x)$ such that $D_2 + \mathrm{div}((y + l)/u_2) = 2D_1$
Step	Expression
1	If $u_{11} = 1$ goto 2'
2	Compute $l(x) = (s_1x + x_0)u_1 + v_1 = s_1x^3 + l_2x^2 + l_1x + l_0$ $s_1 = 1 + u_{11}^2,\ l_2 = v_{11}^2, l_1 = u_{10}^2, l_0 = v_{10}^2 + b$
3	Compute $u_2 = monic(\frac{F+l^2+l}{u_1^2}) = x^2 + u_{21}x + u_{20}$ $w_1 = s_1^{-1},\ u_{21} = w_1^2, u_{20} = (l_2w_1 + u_{11})^2$
4	Compute $v_2 = l + 1 \mod u_2$ $w_2 = w_1 + l_2, w_3 = u_{20}w_2,\ v_{21} = (u_{21} + u_{20})(w_2 + s_1) + w_3 + w_1 + l_1, v_{20} = w_3 + l_0 + 1$
Cost	1I, 3M, (6S)
2'	$l_2 = v_{11}^2, l_1 = u_{10}^2, l_0 = v_{10}^2 + b, u_{20} = l_2^2, v_{20} = u_{20}^2l_2 + v_{10}^4 + 1$
Cost	1M, (6S)

From the doubling formula, we can get the $8P$ formula without inversion in $\mathbb{F}_q$.

Lemma 4.2. *Let H_b be a hyperelliptic curve defined by $y^2+y=x^5+x^3+b$ over $\mathbb{F}_2$. Then for a divisor $D=[x-x_0,\ y_0]$ in J_{H_b},*

$$\begin{aligned}
8D = \operatorname{div}\left(\frac{g(x,\ y)}{u_4(x)^2u_8(x)}\right) &+ [u_8(x),\ v_8(x)],\ \textit{where}\\
g(x,\ y) &= g_4(x,\ y)^2\cdot g_8(x,\ y)\\
g_4(x,\ y) &= y+x^3+({x_0}^2+x_0)^4x^2+x_0^4x+y_0^4+b\\
g_8(x,\ y) &= y+(x_0^2+1)^{16}x^2+(x_0^2+x_0)^{16}x+(x_0^3+x_0+y_0+b+1)^{16}\\
u_4(x) &= x^2+x+(x_0^2+x_0)^8\\
u_8(x) &= x+(x_0^{64}+1)\\
v_8(x) &= (x_0^2+y_0)^{64}+1
\end{aligned}$$

Proof. The formula can be directly computed using (9) and Table 1. □

5. Efficient computation of Tate pairing on H_b

Let $H_b: y^2+y=x^5+x^3+b$ be the hyperelliptic curve over $\mathbb{F}_{2^n}$ and $\gcd(n,6)=1$. In this section, we concern with the twisted Tate pairing

$$\begin{aligned}
\hat{t}\ :\ J_{H_b}(\mathbb{F}_{2^n})[\ell]\times J_{H_b}(\mathbb{F}_{2^n})/\ell J_{H_b}(\mathbb{F}_{2^{12n}}) &\longrightarrow \mathbb{F}^*_{2^{12n}}\\
\hat{t}(D,\ E) &= f_D(\phi(E))^{\frac{2^{12n}-1}{\ell}}.
\end{aligned}$$

Endomorphism ϕ will be explained in the following section 5.1.

5.1. **Endomorphism on H_b.** We identify $\mathbb{F}_{2^{12n}}\cong\mathbb{F}_2(\alpha,\ \tau,\ \varepsilon)$ as the following way.

(1) Take $\alpha\in\mathbb{F}_{2^n}$ whose minimal polynomial $irr(\alpha,\ \mathbb{F}_2)\in\mathbb{F}_2[x]$.
(2) Take $\tau\in\mathbb{F}_{2^{6n}}$ whose minimal polynomial $irr(\tau,\ \mathbb{F}_{2^n})=x^6+x+1\in\mathbb{F}_{2^n}[x]$.
(3) Take $\varepsilon\in\mathbb{F}_{2^{12n}}$ whose minimal polynomial $irr(\varepsilon,\ \mathbb{F}_{2^{6n}})=x^2+x+\tau^5\in\mathbb{F}_{2^{6n}}[x]$.

Then we can obtain the following tower of fields.

$$\begin{array}{l}
\mathbb{F}_{2^{12n}}\cong\mathbb{F}_2(\alpha,\ \tau,\ \varepsilon)\\
|\qquad \varepsilon^2+\varepsilon+\tau^5=0\\
\mathbb{F}_{2^{6n}}\cong\mathbb{F}_2(\alpha,\ \tau)\\
|\qquad \tau^6+\tau+1=0\\
\mathbb{F}_{2^n}\cong\mathbb{F}_2(\alpha)\\
|\\
\mathbb{F}_2
\end{array}$$

Furthermore, the map

$$\phi: H_b(\mathbb{F}_{2^{12n}})\longrightarrow H_b(\mathbb{F}_{2^{12n}})$$

defined by $\phi(x,\ y)=(\tau^5+\tau^4+\tau^2+x,\ x^2\tau^5+x\tau^3+x\tau^2+(x^2+x)\tau+x+y+\varepsilon)$ is an endomorphism. In particular, if $(x_0,\ y_0)\in H_b(\mathbb{F}_{2^n})\subset H_b(\mathbb{F}_{2^{12n}})$ then the *x-coordinate* of $\phi(x_0,\ y_0)$ is in $\mathbb{F}_{2^{6n}}$ and *y-coordinate* of $\phi(x_0,\ y_0)$ is in $\mathbb{F}_{2^{12n}}$.

5.2. **Efficient computation of the Tate pairing on H_b.** Now we describe how to compute the Tate pairing $\hat{t}(D,\ E) = f_D(\phi(E))^{\frac{2^{12n}-1}{\ell}}$ where ϕ is the endomorphism in Section 5.1.

Lemma 5.1. *Let $\hat{t}$ be the twisted Tate pairing on $H_b : y^2 + y = x^5 + x^3 + b$ ($b = 0$ or 1);*

$$\hat{t} : J_{H_b}(\mathbb{F}_{2^n})[\ell] \times J_{H_b}(\mathbb{F}_{2^{12n}})/\ell J_{H_b}(\mathbb{F}_{2^{12n}}) \longrightarrow \mathbb{F}^*_{2^{12n}}$$

Then, for two divisors $D, E \in J_H(\mathbb{F}_{2^n})$, $\hat{t}(D,\ E) = \tilde{f_D}(\phi(E))^{2^{6n}-1}$, where $\tilde{D} = 2^{6n}D - \mathrm{div}(\tilde{f_D})$.

Proof. Let $\mathrm{div}(f_D) = \ell D$. Since $2^{6n}D = \mathrm{div}(\tilde{f_D}) + \tilde{D}$ and ℓ divides $2^{6n} + 1$,

$$\mathrm{div}(f_D^{\frac{2^{6n}+1}{\ell}}) = (2^{6n} + 1)D = \mathrm{div}(\tilde{f_D}(x,\ y)) + \tilde{D} + D = \mathrm{div}(\tilde{f_D}(x,\ y)) + \mathrm{div}(u_D(x)).$$

where $\tilde{D} + D = u_D(x)$. Furthermore, $u(\phi(E)) \in \mathbb{F}_{2^{6n}}$ makes $\hat{t}(D,\ E)$ simple.

$$\begin{aligned}\hat{t}(D,\ E) = t(D, \phi(E)) &= \left[f_D(\phi(E))^{\frac{2^{6n}+1}{\ell}}\right]^{2^{6n}-1} = \left[\tilde{f_D}(\phi(E))u(\phi(E))\right]^{2^{6n}-1} \\ &= \tilde{f_D}(\phi(E))^{2^{6n}-1}\end{aligned}$$

□

From this lemma, we can get $\hat{t}(D,\ E)$ by computing $2^{6n}D$ instead of computing $(2^{6n} + 1)D$. Furthermore, in Lemma 4.2, since the degree of $u_8(x)$ where $(2^3D) = [u_8,\ v_8]$ is again 1, the lemma gives a method to compute 8^nD efficiently for every $n \geq 1$. Thus $\tilde{f_D}(\phi(E))$ can be obtained efficiently by Lemma 4.2 and 5.1. So, one can derive the following algorithm 5.2. We only consider the reduced divisors with $\deg(u_D) = 2$ since the most reduced divisors in J_{H_b} have the form of $D = P_1 + P_2 - 2\mathcal{O}$ [6].

Algorithm 5.2.

Input: $D = [u_1,\ v_1],\ E = [u_2,\ v_2] \in J_{H_b}(\mathbb{F}_q),\ q = 2^n,$ *endomorphism* ϕ
Output: $\hat{t}(D,\ E) = \tilde{f_D}(\phi(E))^{q^6-1}$ *where* $\mathrm{div}(\tilde{f_D}) + \tilde{D} = q^6D$
Step1: *Compute* $P_1,\ P_2$ *and* $Q_1,\ Q_2$ *such that* $D = P_1 + P_2 - 2\mathcal{O}$ *and* $E = Q_1 + Q_2 - 2\mathcal{O}$.
Step2: $f_1 \leftarrow 1,\ f_2 \leftarrow 1,\ \hat{Q_1} \leftarrow \phi(Q_1),\ \hat{Q_2} \leftarrow \phi(Q_2), D_1 \leftarrow P_1 - \mathcal{O},\ D_2 \leftarrow P_2 - \mathcal{O}$
Step3: *For* $i = 1$ *to* $2n$ *do*
  step3-1: *for j=1 and 2, compute* g_j *and* $\hat{D_j}$ *such that* $8D_j = \mathrm{div}(g_j/u_j) + \hat{D_j}$
  step3-2: *for j=1 and 2,* $f_j \leftarrow f_j^8 \cdot g_j(\hat{Q_1}) \cdot g_j(\hat{Q_2}),\ D_j \leftarrow \hat{D_j}$
Step4: *Return* $(f_1 \cdot f_2)^{q^6-1}$

One needs to solve quadratic equation on $\mathbb{F}_{2^n}$ in step 1 of Algorithm 5.2. For a given quadratic equation $x^2 + u_1x + u_0$ defined over $\mathbb{F}_{2^n}$, the following algorithm gives its roots. Here we assume that $n = 2d + 1$ is an odd integer.

Algorithm 5.3.

Input: $u(x) = x^2 + u_1x + u_0$ *in* $\mathbb{F}_{2^n}[x],\ n = 2d + 1$
Output: *a root* α *of* $u(x)$
Step1: *If* $u_1 = 0$*, then* $\alpha \leftarrow u_0^{2^{n-1}}$. *Print out* α *and stop.*
Step2: $a \leftarrow (u_1^{-1})^2u_0$.

Step3: *If* $tr_{\mathbb{F}_{2^n}/\mathbb{F}_2}(a) = 0$*, i.e, the roots of* u *are in* $\mathbb{F}_{2^n}$*, then*

$$\alpha' \leftarrow a + a^{2^2} + a^{2^4} + \cdots + a^{2^{2d}}$$

Step4: *Else, i.e,* $tr_{\mathbb{F}_{2^n}/\mathbb{F}_2}(a) = 1$*, the roots of* u *are in* $\mathbb{F}_{2^{2n}} = \mathbb{F}_{2^n}(\beta)$*, where* $\beta^2 + \beta + 1 = 0$ *and* $\beta = \tau^5 + \tau^4 + \tau^3 + \tau$ *in Section 5.1.*

$$\alpha' \leftarrow \beta + a + a^{2^2} + a^{2^4} + \cdots + a^{2^{2d}}$$

Step5: *Print out* $\alpha = u_1\alpha'$

Let $\tilde{f_D}(\phi(E)) = c + d\varepsilon,\ c,\ d \in \mathbb{F}_{2^{6n}}$. Finally, it still remains to compute $(c + d\varepsilon)^{2^{6n}-1}$ in step4 of Algorithm 5.2. Note that the conjugate of ε, denoted by $\bar{\varepsilon}$, is $\varepsilon + 1$.

$$\begin{aligned}\hat{t}(D,\ E) = (c + d\varepsilon)^{2^{6n}-1} &= \frac{c + d\bar{\varepsilon}}{c + d\varepsilon} = \frac{(c + d\bar{\varepsilon})^2}{\mathrm{Norm}(c + d\varepsilon)} \\ &= \frac{c^2 + d^2(\tau^5 + 1)}{c(c + d) + d^2\tau^5} + \frac{d^2}{c(c + d) + d^2\tau^5}\varepsilon\end{aligned}$$

5.3. **Compressed pairing.** The concept of the *compressed pairing* was suggested by Barreto et al and they claimed one can efficiently reduce the bandwidth occupied by pairing values without impairing security nor processing time [3]. The results in [3] were developed for the curve $y^2 = x^3 - x + 1$, thus the compressed pairing was working on the odd characteristic case. Here, we define compressed pairing in even characterstic case and give a useful fact for exponentiating the result of the Tate pairing. Let $q = 2^n$ where n is an prime.

Definition 5.4. *The compressed pairing is defined as*

$$\delta(D, E) = tr_{\mathbb{F}_{q^{12}}/\mathbb{F}_{q^6}}(\hat{t}(D, E)).$$

Since $\hat{t}(D, E)$ is an element in $\mathbb{F}_{q^{12}}$ of the form $c + d\varepsilon$, where $c, d \in \mathbb{F}_{q^6}$, the compressed pairing is $\delta(D, E) = d$ which depends only d. If we use the compressed pairing instead of the Tate pairing for the protocols, then it's enough to store or send d instead of $c + d\varepsilon$. However, this compression is valuable when one can compute $tr(\hat{t}(D, E)^m)$ for any integer m only using $\delta(D, E)$. As noticed in [3], the exponentiation for pairing values happens in many cryptographic protocols. The compressed pairing for $tr(\hat{t}(D, E)^m) = tr((c + d\varepsilon)^m) = d_m$ can be computed by knowing only $\delta(D, E) = d$ from the following sequence;

Proposition 5.5. *For* $\hat{t}(D, E) = c + d\varepsilon \in \mathbb{F}_{q^{12}},\ c,\ d \in \mathbb{F}_{q^6}, d \neq 0$ *and a positive integer* m*, let* $(c + d\varepsilon)^m = c_m + d_m\varepsilon$*. Then* d_m *is computed by the sequence*

$$d_0 = 0,\ d_1 = d,\ d_m = dd_{m-1} + d_{m-2} \tag{10}$$

which depends only d*.*

Proof. By induction on m, when $m = 1$ the equation holds. First note that

$$\mathrm{Norm}(c + d\varepsilon) = c^2 + cd + d^2\tau^5 = 1 \tag{11}$$

For general m, using $\varepsilon^2+\varepsilon+\tau^5=0$ and the relation (11), we can compute c_m and d_m as follows;

$$\begin{aligned}
c_m &= cc_{m-1}+dd_{m-1}\tau^5\\
d_m &= cd_{m-1}+dc_{m-1}+dd_{m-1}\\
&= c(cd_{m-2}+dc_{m-2}+dd_{m-2})+d(cc_{m-2}+dd_{m-2}\tau^5)+d(cd_{m-2}+dc_{m-2}+dd_{m-2})\\
&= d_{m-2}(cd+1)+d^2(c_{m-2}+d_{m-2})\\
&= d_{m-2}+d(cd_{m-2}+dc_{m-2}+dd_{m-2})\\
&= dd_{m-1}+d_{m-2}.
\end{aligned}$$

□

Note that the trace value of the Tate pairing does not impair an important data.

6. An Implementation result

Since the hyperelliptic curve $H_b : y^2+y=x^5+x^3+b$ over $\mathbb{F}_{2^n}$ has 12 as security multiplier, we may choose $J_{H_0}(\mathbb{F}_{2^{89}})$ for a security reason. Here, we implemented Algorithm 5.2 using NTL library for $H_0 : y^2+y=x^5+x^3$ and $n=89$. A large prime ℓ which divides $\#J_{H_0}(\mathbb{F}_{2^{89}})$ was taken as listed in Table 1 and minimal polynomial of α was taken as $irr(\alpha,\ \mathbb{F}_2)=x^{89}+x^{38}+1$. Note that $\ell\approx 2^{134}$. The elapsed time for computing the Tate pairing essentially depends on the time cost of addition, multiplication and squaring over $\mathbb{F}_{2^{89\cdot 6}}$. So it is important to check the average time for the field operations.

Since the operations in $\mathbb{F}_2(\alpha,\ \tau)$(see Section 5.1) are very slow, one may find $\zeta\in\mathbb{F}_2(\alpha,\ \tau)$ such that $\mathbb{F}_2(\zeta)\cong\mathbb{F}_2(\alpha,\ \tau)$ to improve a computing speed. Here, we did another implementation using an isomorphism of fields. To find an isomorphism between $\mathbb{F}_2(\zeta)$ and $\mathbb{F}_2(\alpha,\ \tau)$, let $\alpha\in\mathbb{F}_{2^{89}}$ whose minimal polynomial is

$$\begin{aligned}
irr(\alpha,\ \mathbb{F}_2) &= x^{89}+x^{87}+x^{86}+x^{85}+x^{81}+x^{78}+x^{77}+x^{76}+x^{75}\\
&\quad +x^{73}+x^{69}+x^{65}+x^{64}+x^{63}+x^{62}+x^{61}+x^{60}+x^{59}\\
&\quad +x^{58}+x^{57}+x^{54}+x^{53}+x^{51}+x^{46}+x^{44}+x^{43}+x^{39}\\
&\quad +x^{37}+x^{34}+x^{33}+x^{32}+x^{30}+x^{29}+x^{27}+x^{26}+x^{23}\\
&\quad +x^{22}+x^{21}+x^{20}+x^{17}+x^{15}+x^{14}+x^{11}+x^{9}+x^{6}\\
&\quad +x^{5}+x^{3}+x^{2}+1.
\end{aligned}\tag{12}$$

rather than $irr(\alpha,\ \mathbb{F}_2)=x^{89}+x^{38}+1$. Let $\tau\in\mathbb{F}_{2^{89\cdot 6}}$ whose minimal polynomial is $irr(\tau,\ \mathbb{F}_{2^{89}})=x^6+x+1$. And let $\zeta\in\mathbb{F}_{2^{6\cdot 89}}$ whose minimal polynomial is $irr(\zeta,\ \mathbb{F}_2)=x^{534}+x^{161}+1$. Then $\mathbb{F}_{2^{6\cdot 89}}\cong\mathbb{F}_2(\alpha,\ \tau)\cong\mathbb{F}_2(\zeta)$ and we can express α and τ in terms of ζ (see Appendix A).

The following Table 3 shows an average time for the field operations on $\mathbb{F}_p$, for prime p, $\mathbb{F}_2(\alpha,\ \tau)$ and $\mathbb{F}_2(\zeta)$ respectively. It was obtained on a 2GHz Pentium IV with 512 Mb RAM under windows. The compiler was Microsoft Visual C++ 6.0 and NTL library was used. This is the average timing from 10000 trials using the above fields. Time is given in μs.

Table 4 is the our main result, which compares the timing of computation of the Tate pairing with that by Y. Choie and E. Lee [8] which is unique implementation result of the Tate pairing on hyperelliptic curves. The fields $\mathbb{F}_2(\alpha,\ \tau)$ and $\mathbb{F}_2(\zeta)$ in Table 4 are same as the the same as those in Table 3.

TABLE 3. Average time for the field operations

	$\mathbb{F}_p \cong \mathbb{F}_{2^{534}}$	$\mathbb{F}_2(\alpha,\tau) \cong \mathbb{F}_{2^{534}}$	$\mathbb{F}_2(\zeta) \cong \mathbb{F}_{2^{534}}$
minimal polynomials	$p \approx 2^{534}$ prime	$irr(\alpha,\ \mathbb{F}_2) = x^{89} + x^{38} + 1$, $irr(\tau,\ \mathbb{F}_{2^{89}}) = x^6 + x + 1$	$irr(\zeta, \mathbb{F}_2) =$ $x^{534} + x^{161} + 1$
addition	3.2	6.2	1.5
multiplication	34.4	156.3	37.5
squaring	28.7	71.1	4.7
inversion	204.0	845.3	156.3

TABLE 4. Comparison of results

	Results in [8]	Our results	
environments	2GHz Pentium IV, 256 RAM with MP library	2GHz Pentium IV, 512 RAM with NTL library	
curve	$y^2 = x^5 + a$, $a \in \mathbb{F}_p^*$ over $\mathbb{F}_p$	$y^2 + y = x^5 + x^3$ over $\mathbb{F}_2$	
field	$\mathbb{F}_{p^4}$, $p \approx 2^{256}$, $p \equiv 2,\ 3 \pmod 5$	$\mathbb{F}_2(\alpha,\ \tau)(\varepsilon) \cong \mathbb{F}_{2^{89\cdot 12}}$	$\mathbb{F}_2(\zeta)(\varepsilon) \cong \mathbb{F}_{2^{89\cdot 12}}$
average time	$515 \sim 594\ ms$	$3640\ ms$	$479.5\ ms$

7. CONCLUSION

Since the Tate pairing was suggested to construct a cryptosystem, many efforts to improve the computational speed of the Tate pairing has been researched. However, there are only a few of implementation results of the Tate pairing reported. Since most of cryptosystems have based on binary field, it may be meaningful to implement the Tate pairing on such a field.

In this paper, we suggest an efficient algorithm for computing the Tate pairing on the hyperelliptic curves $H_b : y^2+y = x^5+x^3+b$, which is known to have the maximal security multiplier, and implemented the Tate pairing on $H_0 : y^2 + y = x^5 + x^3$ defined over $\mathbb{F}_2$. We also found the extension field $\mathbb{F}_2(\zeta) \cong \mathbb{F}_2(\alpha,\tau)$ and did another implementation of the Tate pairing on $\mathbb{F}_2(\zeta)$ to improve a computing speed. This is the first attempt to compute Tate paring of hyperelliptic curve over the binary fields. We also give an explicit description how to find an isomorphism between two given large fields in Appendix A. Furthermore, we showed the *compressed pairing* can be defined on the curve H_b defined over $\mathbb{F}_{2^n}$, which was explored only in the case of odd characteristics in [3].

From the Table 4, it seems that the computation of the Tate pairing on binary fields is as efficient as on prime fields. One may get even better result using optimal normal basis of the ground fields. Therefore, we may conclude that a binary field is more suitable than prime field for computing the Tate pairing, when the same number of operations are required, from Table 3.

REFERENCES

[1] P.S.Barreto, H.Y.Kim, B. Lynn and M. Scott, Efficient Algorithms for Pairing-Based Cryptosystems, *Advances in Cryptology-Crypto 2002*, LNCS **2442**, pp.354-368 (2002).
[2] P. Barreto, B. Lynn and M. Scott, On the Selection of Pairing Friendly Groups, *SAC 2003*, LNCS **3006** pp.17-25 (2004).

[3] P.S.Barreto, Compressed Pairings, *Advances in Cryptology-Crypto 2004*, LNCS **3152**, pp.140-156 (2004).
[4] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing. *SIAM J. Comput.* **32**, No. 3, pp.586–615 (2003) (electronic).
[5] D. Boneh, H. Shacham and B. Lynn, Short signatures from the Weil pairing, *Advances in Cryptology-AsiaCrypt 2001*, LNCS **2248**, pp. 514-532 (2001).
[6] D.G.Cantor, Computing in the Jacobian of a Hyperelliptic Curves, *Math. Comp*, *48*, No.177, pp.95-101 (1987).
[7] L. Chen and C. Kudla, Identity Based Authenticated Key Agreement Protocols from Pairings, Cryptology eprint Archives, Available at http://eprint.iacr.org, (2002), Number 2002/184.
[8] Y. Choie and E. Lee, Implementation of Tate Pairing on Hyperelliptic Curves of Genus 2, *Information security and cryptology-ICISC 2003*, LNCS **2971**, pp.97-111 (2004)
[9] I. Duursma and H. Lee, Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$, *Advances in cryptology-AsiaCrypt 2003*, LNCS **2894**, pp. 111–123 (2003).
[10] G. Frey and H-G. Rück, A remark concerning m-divisibility in the divisor class group of curves, *Math.Comp.* **62**, No.206, pp.865-874 (1994).
[11] S. Galbraith, Supersingular curves in Cryptography, *Advances in Cryptology-AsiaCrypt 2001*, LNCS **2248**, pp.495-513 (2002).
[12] P. Gaudry, An algorithm solving the discrete log problem on hyperelliptic curves, *Advances in Cryptology-Eurocrypt 2000*, LNCS **1807**, pp.19-34 (2000).
[13] S. Galbraith, K. Harrison, and D. Soldera, Implementing the Tate pairing, *Algorithmic Number Theory Symposium - ANTS-V*, LNCS **2369**, pp.324-337 (2002).
[14] N. Koblitz, *Algebraic aspects of cryptography*, Springer-Verlag (1998).
[15] T. Lange, Efficient Arithmetic on Genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae, Cryptology eprint Archives, Available at http://eprint.iacr.org, (2002), Number 2002/121.
[16] A.J. Menezes, T. Okamoto and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions of Information Theory*, **39**, No 5, pp.1639-1646 (1993).
[17] V. Miller, Short Programs for Functions on Cuvres, Unpublished manuscript, (1986).
[18] D. Mumford, *Tata Lectures on Theta II*, Birkhäuser, (1984).
[19] K. Rubin and A. Silverberg, The Best and Worst of Supersingular Abelian Varieties in Cryptology, eprint 2002/006.
[20] A. Shamir, Identity-based cryptosystems and signature schemes, *Advances in Cryptology-CRYPTO 84*, LNCS **196**, pp.47-53 (1985).
[21] N.P. Smart, On the performance of Hyperelliptic Cryptosystems, *Advances in Cryptology-Eurocrypt 99*, LNCS **1592**, pp.165-175 (1999).
[22] E.R. Verheul, Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems, *Advances in Cryptology-AisaCrypt 2001*, LNCS **2248**, pp.433-551 (2002).
[23] C. Xing, On supersingular abelian varieties of dimension two over finite fields, *Finite fields and their application*, **2**, pp.407-421 (1996).

Appendix A. Field isomorphisms

Let $\zeta \in \mathbb{F}_{2^{534}}$ with $irr(\zeta,\ \mathbb{F}_2) = x^{534} + x^{161} + 1$. Then $\mathbb{F}_{2^{534}} \cong \mathbb{F}_2(\zeta) \cong \mathbb{F}_2[x]/(x^{534} + x^{161} + 1)$. Note that $\zeta^2 + \zeta + 1$ is a primitive $(2^{534} - 1)$th root of unity in $\mathbb{F}_2(\zeta)$. So $\mathbb{F}_{2^{89}} \cong \mathbb{F}_2(\alpha)$ where $\alpha = (\zeta^2 + \zeta + 1)^{\frac{2^{534}-1}{2^{89}-1}}$. It can be computed that $irr(\alpha,\ \mathbb{F}_2) = \prod_{i=0}^{88}(x - \sigma^i(\alpha))$ where σ is a Frobenius automorphism. The computation of $irr(\alpha,\ \mathbb{F}_2)$ is shown in the equation (12). The expression of α and τ in terms of ζ is listed below. We express $a_0 + a_1\zeta + \cdots + a_n\zeta^n$ as $[a_0a_1 \cdots a_n]$, where $a_i \in \{0,\ 1\}$. Then

$$
\begin{aligned}
\alpha = [&0010010110101000111101101010011101001110011100 1101\\
&00011001110001001100110110101111011010001011011001\\
&00111101011011010110000100110100011100101101111001\\
&11111100101001100110001110111011001011001110001000\\
&00010111111000011011111110100110101101000111000000\\
&10011010011101000101010000000001111011100010011100 0\\
&11001000010110000111001011001011001011100100110001\\
&10110110110110100001010001111111000101101110111011\\
&00011110111010100110010000110111000111001000010101\\
&00011111011101010011110100101000011110000111111000\\
&0111101011011011010011010011011]\\
\tau = [&01111010100010111001111101000101001010111101011100\\
&00010010101110000000111100110110000111011010000101\\
&10100100000110001111100100010111110111111011111110\\
&11110001110111101110101101000110110001001110001001\\
&10100000110110111010000011011100110001010001111001\\
&00111011110111000010010010111111001110010000100000\\
&11000100000000100010000110110110011000010100110110\\
&01100001100011100101010001101101011011101110101001\\
&01101010100110111001011110110100000110010000000000\\
&10111111000100101011111111111111101110000011011100\\
&1101110110010100000001000001111 0011].
\end{aligned}
$$

Department of Mathematics, Pohang University of Science and Technology, Pohang 790-784, Korea
E-mail address: yjc@postech.ac.kr

Department of Mathematics, Pohang University of Science and Technology, Pohang 790-784, Korea
E-mail address: ant@postech.ac.kr

School of Computational Sciences, Korea Institute for Advanced Study, Seoul 130-722, Korea.
E-mail address: ejlee@kias.re.kr