Arithmetic on Pairing-Friendly Fields

Sedat AKLEYLEK, Barış Bülent KIRLAR, Ömer SEVER, Zaliha YÜCE

***Abstract*—Pairing-friendly fields are introduced mainly for extension field operations used in pairing based cryptography. In this note, we give the implementation and comparison results of aritmetic operations on extension fields constructed by trinomials and cyclotomic polynomials with extension degree 2,3,4,5 and also pairing-friendly fields with extension degree 2,4.**

***Keywords* —extension field arithmetic, cyclotomic polynomials, pairing-friendly fields.**

I. Introduction

Bilinear pairings were known but has not been used widely in cryptography until the three-party key exchange in one round protocol was proposed by Joux in [9]. Then, identity-based encryption scheme in [5] and the short signature scheme in [4] were given. Afterwards, many protocols based on pairings has seen frequently.

For pairing-based cryptosystems efficient arithmetic on extension fields is a crucial issue for implementing efficient pairings. This arithmetic is primarily based on the field operations. In this context, pairing-friendly fields are defined by Koblitz and Menezes in [12]. In extension field arithmetic we are interested in multiplication, exponentiation, inversion and their computational costs. It is known that addition and substraction are trivial and squaring is almost equivalent to multiplication.

Choosing elliptic curve with appropriate parameters is very important for an efficient and secure pairing implementation issues. Generally; $\mathbb{F}_p$ must be large enough, so that $E(\mathbb{F}_p)$ can prevent the success of elliptic curve discrete logarithm attacks, while $\mathbb{F}_{p^k}$ must be large enough to resist discrete logarithm attacks over finite fields. At the same time, p and k should be as small as possible to minimize time and space usage. In the literature, k is generally the form of $k = 2^i 3^j$, where $i \geq 0$, $j \geq 0$ and not both them are zero. MNT curve proposed in [16] for pairing-based cryptography with the embedding degree $3, 4, 6$, while the embedding degree 3 is not in the class of pairing-friendly fields. Embedding degree smaller than or equal to 6 is enough for near future, embedding degree 10 and 12 curves which currently produce excessively large finite fields may be more desirable as time passes and index calculus improvements are discovered.

When measuring the efficiency of an extension field operation, one must count the number of base field operations. Multiplication is the most used one among the base field operations and that is why we primarily focus on it in this note. As previous works; multiplication and squaring methods for pairing-friendly fields $\mathbb{F}_{p^k}$ with $k \in 2, 3, 4, 6$ has been reviewed in [7]. The formulas of Karatsuba multiplication in extension fields are given in [19]. In exponentiation Lucas method provides a faster way than the classical square-and-multiply method.

Our contribution is to show that cyclotomic polynomials and trinomials can also be used in the construction of extension fields as efficient as pairing-friendly fields.

Organization of this note is as follows. Section II summarizes the mathematical preliminaries. The definition of pairing-friendly fields and construction of extension fields by using trinomials and cyclotomic polynomials are given in Section III. We describe the methods and the implementation results for multiplication, inversion and exponentiation in Section IV. Timings are given in Section V. We conclude in Section VI.

II. Preliminaries

It is well known that when any polynomial $f(x) \in \mathbb{F}_p[x]$ of order k is irreducible over the finite field $\mathbb{F}_p$, one can get a simple algebraic extension $\mathbb{F}_{p^k}$ of $\mathbb{F}_p$ with a root of $f(x)$ as a defining element of $\mathbb{F}_{p^k}$: $\mathbb{F}_{p^k} = \mathbb{F}(\alpha) = \mathbb{F}_p[x]/ < f(x) >= \{\sum c_i \alpha^i | c_i \in \mathbb{F}_p\}$. Multiplication in $\mathbb{F}_{p^k}$ is computed as a multiplication of polynomials with modulo $f(x)$ reduction. Construction of finite fields with suitable irreducible $f(x)$ is important for an efficient arithmetic. When one focuses on multiplication, there are various techniques in the literature (see [3], [11]) for multiplying two n-digit integers, but we only recall briefly Karatsuba technique since this was implemented in open source libraries. Karatsuba's method [10] proceeds by splitting the integers into k parts, and therefore it is natural to consider using it to implement multiplication in quadratic field extensions. For computing the product $c = ab \in \mathbb{F}_{p^2}$ with the polynomial $x^2 - \beta$ using the Karatsuba Method proceeds by precomputing $v_0 = a_0 b_0$, $v_1 = a_1 b_1$, and then

$$\begin{aligned} c_0 &= v_0 + \beta v_1 \\ c_1 &= (a_0 + a_1)(b_0 + b_1) - v_0 - v_1 \end{aligned}$$

We now give the definition of cyclotomic polynomials which are used to get the extension fields in the following sections.

Definition 1. *The roots of the splitting field of $x^n - 1$ over a field K are called the n-th roots of unity over K. A generator of the cyclic group composing of all these roots is called a primitive n-th root of unity over K.*

Manuscript received November 10, 2008.

S. Akleylek, B. B. Kırlar, Ö. Sever, Z. Yüce are with the Institute of Applied Mathematics, METU, Ankara, Turkey, e-mail: {akleylek,kirlar}@metu.edu.tr, severomer@yahoo.com, zyuce@stm.com.tr

S. Akleylek is also with the Department of Computer Engineering, On-Dokuz Mayıs University, Samsun, Turkey

B. B. Kırlar is also with the Department of Mathematics, Süleyman Demirel University, Isparta, Turkey

Z. Yüce is a software engineer in STM A.Ş.

* This research is partially supported by ASELSAN A.Ş.

Definition 2. *Let $\mathbb{F}_{p^k}$ be a field, $n \in \mathbb{Z}^+$ and $p \nmid n$. Then, the n-th cyclotomic polynomial $\Phi_n(x)$ is defined as*

$$\prod_{\xi}(x-\xi)$$

where ξ ranges over the primitive n-th roots of unity.

We need the Frobenius action that plays a crucial role in the arithmetic of extension field.

Definition 3. *Let $\mathbb{F}_{p^k}$ be a finite field. Then, the Frobenius map $\varphi_p : \mathbb{F}_{p^k} \longrightarrow \mathbb{F}_{p^k}$ is given by the map $x \mapsto x^p$.*

III. Pairing-Friendly Fields

Extension field arithmetic has an important role in pairing-based cryptosystems for an efficient implementation. The problem is how to choose or find suitable irreducible $f(x)$ with degree k for an efficient implementation. In this problem, $f(x)$ should have low-weight such as binomials ($x^k+\beta$, where $\beta \in \mathbb{F}_p$) and trinomials ($x^k+x^h+\delta$, where $k > h$ and $\delta \in \mathbb{F}_p$). In this respect, one must give a restriction on p that how to find such β and δ. Koblitz and Menezes introduced the notion of pairing-friendly fields in [12].

Definition 4. *Let $k = 2^i3^j$ with $i > 0$ and $j \geq 0$. If $p \equiv 1 \pmod{12}$, then $\mathbb{F}_{p^k}$ is called pairing-friendly field.*

Theorem 5. *[12] Let $\mathbb{F}_{p^k}$ be a pairing-friendly field and $\beta \in \mathbb{F}_p$ be neither a square nor a cube in $\mathbb{F}_p$. Then, $x^k - \beta$ in $\mathbb{F}_p[x]$ is irreducible over $\mathbb{F}_p$.*

The advantage of constructing pairing-friendly fields is to easily find a small value of β which helps to reduce the cost of extension field arithmetic. With the help of this construction, one can perform a field multiplication in $\mathbb{F}_{p^k}$, where $k = 2^i3^j$, with 3^i5^j field multiplications in $\mathbb{F}_p$ by Karatsuba method. There are special examples for quadratic extension.

1) [18] Let $p \equiv 3 \pmod 4$ and $k = 2$. Then, $x^2 + 1$ is irreducible over $\mathbb{F}_p[x]$ since -1 is a quadratic non-residue modulo p. Representation of elements is a_0+a_1i, where i is the imaginary square root of -1 and $a_0, a_1 \in \mathbb{F}_p$.
2) [13] Let $p \equiv 2 \pmod 3$ and $k = 2$. Then, x^2+x+1 is irreducible over $\mathbb{F}_p[x]$ (See Lemma 6). Representation of elements is $a_0\alpha+a_1\alpha^2$, where $\alpha^3 = 1$ and $a_0, a_1 \in \mathbb{F}_p$.

Construction of extension fields is not limited with this definition for an efficient implementation for pairing-based cryptographic systems. In this section, we give other classes which uses cyclotomic polynomials for the embedding degrees $2, 4, 6, 10$ instead of binomials and trinomials for the embedding degrees 3, 5 and show that the cost of the multiplication is the same as fields given in Definition 4. Moreover, for every cyclotomic polynomials, where $(k+1)$ is prime one can choose different prime numbers to use in construction of extension fields as efficient as pairing-friendly fields.

Lemma 6. 1) *Let p be a prime such that $p \equiv 2 \pmod 3$. Then, the third cyclotomic polynomial $\Phi_3(x) = (x^3 - 1)/(x-1) = x^2 + x + 1$ is irreducible over $\mathbb{F}_p$.*
2) *Let p be a prime such that $p \equiv \mp 2 \pmod 5$. Then, the fifth cyclotomic polynomial $\Phi_5(x) = (x^5-1)/(x-1) = x^4 + x^3 + x^2 + x + 1$ is irreducible over $\mathbb{F}_p$.*
3) *Let p be a prime such that $p \equiv 3 \pmod 7$ or $p \equiv 5 \pmod 7$. Then, the seventh cyclotomic polynomial $\Phi_7(x) = (x^7-1)/(x-1) = x^6+x^5+x^4+x^3+x^2+x+1$ is irreducible over $\mathbb{F}_p$.*
4) *Let p be a prime such that $p \equiv 2 \pmod{11}$ or $p \equiv 6 \pmod{11}$ or $p \equiv 7 \pmod{11}$ or $p \equiv 8 \pmod{11}$. Then, the eleventh cyclotomic polynomial $\Phi_{11}(x) = (x^{11} - 1)/(x - 1) = x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ is irreducible over $\mathbb{F}_p$.*

Proof:

1) Since $p - 1 \equiv 1 \pmod 3$ and $ord(\mathbb{F}_p^*) = p - 1$, there is no element in $\mathbb{F}_p^*$ of order 3. Then, $\alpha^3 - 1 = (\alpha - 1)\Phi_3(\alpha) \neq 0$ for any $\alpha \neq 1$ in $\mathbb{F}_p^*$. As a result, $\Phi_3(x)$ has no roots in $\mathbb{F}_p$. The proofs of (2)-(4) are similar. ■

It should be noticed that k^{th} cyclotomic polynomial can be used when $(k + 1)$ is a prime number. To perform a field multiplication in $\mathbb{F}_{p^k}$ one needs $k(k+1)/2$ field multiplications in $\mathbb{F}_p$. Trinomials with the extension degree 3 and 5 are given in the following lemma.

Lemma 7. 1) *Let $k = 3$. Then, $f(x) = x^3 + x + 1$ is irreducible over $\mathbb{F}_p$ if $p \equiv 1 \pmod 4$.*
2) *Let $k = 5$. Then, $f(x) = x^5 + x^2 + 1$ is irreducible over $\mathbb{F}_p$ if $p \equiv 1 \pmod 6$.*

Table I summarizes how to choose primes when one has irreducible binomial, trinomial or cyclotomic polynomials with small degree k. For every trinomials or cyclotomic polynomials one can use different primes to construct extension fields as efficient as pairing-friendly fields.

TABLE I
Prime Classes for Irreducible Binomial, Trinomial and Cyclotomic Polynomials

	Binomial	Trinomial	Cyclotomic
Quadratic	$p \equiv 1 \pmod{12}$	$p \equiv 2 \pmod 3$	$p \equiv 2 \pmod 3$
Cubic	$p \equiv 1 \pmod{12}$	$p \equiv 1 \pmod 4$	-
Quartic	$p \equiv 1 \pmod{12}$	-	$p \equiv \mp 2 \pmod 5$
Quintic	-	$p \equiv 1 \pmod 6$	-
Sextic	$p \equiv 1 \pmod{12}$	-	$p \equiv 3 \pmod 7$ $p \equiv 5 \pmod 7$
Tenth	-	-	$p \equiv 2 \pmod{11}$ $p \equiv 6 \pmod{11}$ $p \equiv 7 \pmod{11}$ $p \equiv 8 \pmod{11}$

In the following section, we will show that the costs of multiplication, exponentiation and inversion using cyclotomic polynomials and trinomials are more or less identical.

IV. Multiplication, Inversion and Exponentiation on Pairing-Friendly Fields

A. Multiplication

Field multiplication is one of the most used arithmetic operation in the extension fields. In this subsection, we give

quadratic, direct quartic extensions by cyclotomic polynomials and cubic, quintic extensions by trinomials irreducible over $\mathbb{F}_p$ with their costs. Finally, we review the idea how to construct the tenth extension field with quadratic over quintic and quintic over quadratic. The cost for multiplying elements in $\mathbb{F}_{p^k}$ is based on the following operations on $\mathbb{F}_p$: multiplication (M), addition or substraction (A) and multiplication by 2 (S).

1) Quadratic: Let $p = 2 \pmod 3$ and $F_{p^2} = F_p[x]/ < x^2 + x + 1 >$, $x^3 = 1$. Let α be a root of $x^2 + x + 1$.
$a = a_0\alpha + a_1\alpha^2$, where $a \in F_{p^2}$ and $a_0, a_1 \in F_p$
$b = b_0\alpha + b_1\alpha^2$, where $b \in F_{p^2}$ and $b_0, b_1 \in F_p$

$$\begin{aligned} a \times b &= (a_0\alpha + a_1\alpha^2)(b_0\alpha + b_1\alpha^2) \\ &= (a_1b_1 - a_0b_1 - a_1b_0)\alpha \\ &+ (a_0b_0 - a_0b_1 - a_1b_0)\alpha^2 \\ &= c_0\alpha + c_1\alpha^2 \end{aligned}$$

where

$$\begin{aligned} c_0 &= a_1b_1 - a_0b_1 - a_1b_0 \\ c_1 &= a_0b_0 - a_0b_1 - a_1b_0 \end{aligned}$$

Set

$$\begin{aligned} v_0 &= a_0b_0 \\ v_1 &= a_1b_1 \\ v_2 &= (a_0 - a_1)(b_0 - b_1) \end{aligned}$$

Then,

$$\begin{aligned} c_0 &= -v_0 + v_2 \\ c_1 &= -v_1 + v_2 \end{aligned}$$

Cost : 3M + 4A

Table III shows the implementation timings of multiplication in $\mathbb{F}_{p^2}$ with MIRACL based code, our code and NTL module. MIRACL zzn2_mul function is clearly fastest because it uses a combination of Lazy Reduction and Karatsuba's method.

2) Cubic: Let $p = 1 \pmod 4$ and $\mathbb{F}_{p^3} = \mathbb{F}_p[x]/ < x^3 + x + 1 >$, $x^3 = -x - 1$. Let α be a root of $x^3 + x + 1$.
$a = a_0 + a_1\alpha + a_2\alpha^2$, where $a \in \mathbb{F}_{p^3}$ and $a_0, a_1, a_2 \in \mathbb{F}_p$
$b = b_0 + b_1\alpha + b_2\alpha^2$, where $b \in \mathbb{F}_{p^3}$ and $b_0, b_1, b_2 \in \mathbb{F}_p$

$$\begin{aligned} a \times b &= (a_0 + a_1 + a_2)(b_0 + b_1 + b_2) \\ &= a_0b_0 - a_1b_2 - a_2b_1 \\ &+ (a_1b_0 + a_0b_1 - a_1b_2 - a_2b_1 - a_2b_2)\alpha \\ &+ (a_2b_0 + a_1b_1 + a_0b_2 - a_2b_2)\alpha^2 \\ &= c_0 + c_1\alpha + c_2\alpha^2 \end{aligned}$$

where

$$\begin{aligned} c_0 &= a_0b_0 - a_1b_2 - a_2b_1 \\ c_1 &= a_1b_0 + a_0b_1 - a_1b_2 - a_2b_1 - a_2b_2 \\ c_2 &= a_2b_0 + a_1b_1 + a_0b_2 - a_2b_2 \end{aligned}$$

Set

$$\begin{aligned} v_0 = a_0b_0, &\quad v_3 = (a_0 + a_1)(b_0 + b_1), \\ v_1 = a_1b_1, &\quad v_4 = (a_0 + a_2)(b_0 + b_2), \\ v_2 = a_2b_2, &\quad v_5 = (a_1 + a_2)(b_1 + b_2). \end{aligned}$$

Then,

$$\begin{aligned} c_0 &= v_0 + v_1 + v_2 - v_5 \\ c_1 &= -v_0 + v_3 - v_5 \\ c_2 &= -v_0 - 2v_2 + v_4 \end{aligned}$$

Cost : 6M+13A+1S

Table IV explains the implementation timings of multiplication in $\mathbb{F}_{p^3}$ with MIRACL prepared code, our code and NTL module. Our code has a redundancy and MIRACL method is faster in larger prime modulo although Karatsuba are used by all methods and computational cost is the same theoretically.

3) Quartic: Let $p \equiv \mp 2 \pmod 5$ and $\mathbb{F}_{p^4} = F_p[x]/ < x^4 + x^3 + x^2 + x + 1 >$, $x^5 = 1$. Let α be a root of $x^4 + x^3 + x^2 + x + 1$.
$a = a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4$, where $a \in \mathbb{F}_{p^4}$ and $a_0, a_1, a_2, a_3 \in \mathbb{F}_p$
$b = b_0\alpha + b_1\alpha^2 + b_2\alpha^3 + b_3\alpha^4$, where $b \in \mathbb{F}_{p^4}$ and $b_0, b_1, b_2, b_3 \in \mathbb{F}_p$

$$\begin{aligned} a \times b &= (a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a^3\alpha^4) \\ &\times (b_0\alpha + b_1\alpha^2 + b_2\alpha^3 + b^3\alpha^4) \\ &= c_0\alpha + c_1\alpha^2 + c_2\alpha^3 + c_3\alpha^4 \end{aligned}$$

Set

$$\begin{aligned} v_0 = a_0b_0, &\quad v_5 = (a_0 + a_2)(b_0 + b_2), \\ v_1 = a_1b_1, &\quad v_6 = (a_0 + a_3)(b_0 + b_3), \\ v_2 = a_2b_2, &\quad v_7 = (a_1 + a_2)(b_1 + b_2), \\ v_3 = a_3b_3, &\quad v_8 = (a_1 + a_3)(b_1 + b_3), \\ v_4 = (a_0 + a_1)(b_0 + b_1), &\quad v_9 = (a_2 + a_3)(b_2 + b_3). \end{aligned}$$

Then,

$$\begin{aligned} c_0 &= v_0 + 2v_2 - v_6 - v_7 + v_8 \\ c_1 &= 2v_0 + v_1 - v_6 - v_7 + v_9 \\ c_2 &= v_2 + 2v_3 + v_4 - v_6 - v_7 \\ c_3 &= 2v_1 + v_3 + v_5 - v_6 - v_7 \end{aligned}$$

Cost : 10M+28A+4S

The implementation timings of multiplication in $\mathbb{F}_{p^4}$ with our code based on MIRACL and NTL module are demonstrated in Table V. Our code uses cyclotomic polynomial which results as a direct extension.

4) Quintic: Let $p \equiv 1 \pmod 6$ and $\mathbb{F}_{p^5} = F_p[x]/ < x^5 + x^2 + 1 >$, $x^5 = -x^2 - 1$. Let α be a root of $x^5 + x^2 + 1$.
$a = a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_4\alpha^4$, where $a \in \mathbb{F}_{p^5}$ and $a_0, a_1, a_2, a_3, a_4 \in \mathbb{F}_p$

$b = b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3 + b_4\alpha^4$, where $b \in \mathbb{F}_{p^5}$ and $b_0, b_1, b_2, b_3, b_4 \in \mathbb{F}_p$

$$\begin{aligned} a \times b &= (a_0 + a_1\alpha + a_2\alpha^2 + a^3\alpha^3 + a^4\alpha^4) \\ &\times (b_0 + b_1\alpha + b_2\alpha^2 + b^3\alpha^3 + b^4\alpha^4) \\ &= c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3 + c_4\alpha^4 \end{aligned}$$

Set

$$\begin{aligned} v_0 &= a_0b_0, & v_8 &= (a_0 + a_4)(b_0 + b_4), \\ v_1 &= a_1b_1, & v_9 &= (a_1 + a_2)(b_1 + b_2), \\ v_2 &= a_2b_2, & v_{10} &= (a_1 + a_3)(b_1 + b_3), \\ v_3 &= a_3b_3, & v_{11} &= (a_1 + a_4)(b_1 + b_4), \\ v_4 &= a_4b_4, & v_{12} &= (a_2 + a_3)(b_2 + b_3), \\ v_5 &= (a_0 + a_1)(b_0 + b_1), & v_{13} &= (a_2 + a_4)(b_2 + b_4), \\ v_6 &= (a_0 + a_2)(b_0 + b_2), & v_{14} &= (a_3 + a_4)(b_3 + b_4). \\ v_7 &= (a_0 + a_3)(b_0 + b_3), \end{aligned}$$

Then,

$$\begin{aligned} c_0 &= 2v_0 + v_2 + v_3 + 2v_4 - v_{11} - v_{12} \\ c_1 &= -v_0 - v_1 + v_2 - v_3 + v_4 + v_5 - v_{13} \\ c_2 &= -v_0 + 2v_1 + 2v_3 + 3v_4 + v_6 - v_{11} - v_{12} - v_{14} \\ c_3 &= -v_0 - v_1 - 2v_3 + 2v_4 + v_7 + v_9 - v_{13} \\ c_4 &= -v_0 - v_1 + v_2 + v_8 + v_{10} - v_{14} \end{aligned}$$

Cost : 15M+49A+7S

The implementation timings of multiplication in $\mathbb{F}_{p^5}$ with our code and NTL module are given in Table VI. Our method is used MIRACL's base field operations and runs faster than NTL in larger prime modulos but slower in smaller prime modulos.

5) Tenth Extension: We consider two possibilities for building a tenth extension : quadratic over quintic and quintic over quadratic.

1) We build $\mathbb{F}_{p^{10}}$ as $\mathbb{F}_{p^5}[y]/(y^2-\gamma)$, where $\mathbb{F}_{p^5} = \mathbb{F}_p/(y^5-\beta)$, β is a quintic non-residue in $\mathbb{F}_p$. An element $a \in \mathbb{F}_{p^{10}}$ is represented as $a_0 + a_1y$, where $a_0, a_1 \in \mathbb{F}_{p^5}$.
2) We construct $\mathbb{F}_{p^2} = \mathbb{F}_p[x]/(x^2 - \beta)$ and $\mathbb{F}_{p^{10}} = \mathbb{F}_{p^2}[y]/(y^5-\gamma)$, where β is a quadratic non-residue in $\mathbb{F}_p$ and $\gamma = \sqrt[5]{\beta}$ is a quintic non-residue in $\mathbb{F}_{p^2}$. An element $a \in \mathbb{F}_{p^{10}}$ is represented as $a_0+a_1y+a_2y^2+a_3y^3+a_4y^4$, where $a_0, a_1, a_2, a_3, a_4 \in \mathbb{F}_{p^2}$.

The implementation timings of multiplication in $\mathbb{F}_{p^{10}}$ with our codes and NTL module are shown in Table VII. We implemented two different towering. The results show that quadratic over quintic is faster than quintic over quadratic as expected, since establishing towering from bigger modulos to smaller modulos is faster than vice versa.

Summary of multiplicative costs for quadratic, cubic, quartic and quintic extensions is given in Table II. It is observed that there is no significant difference when one compares with pairing-friendly fields. However, by using cyclotomic polynomial instead of binomial in quadratic extension one can implement quadratic field multiplication with 1A+1S less than binomials.

TABLE II
MULTIPLICATIVE COSTS FOR QUADRATIC, CUBIC, QUARTIC AND QUINTIC EXTENSIONS

	Quadratic	Cubic	Quartic	Quintic
Multiplication	3	6	10	15
Addition	4	13	28	49
Shift	-	1	4	7

B. Inversion

Inversion in extension field with degree k corresponds to computing a multiplicative inverse of a polynomial modulo an irreducible polynomial $f(x)$ of degree k. There is an efficient algorithm in [11]. By using this algorithm and irreducible polynomials defined in Section 3, one can obtain the following examples.

Example 8. 1) *Let $p \equiv 1 \pmod{12}$ and $x^2 - \beta$ be irreducible over $\mathbb{F}_p$ where β is quadratic non-residue in $\mathbb{F}_p$. Let $a \in \mathbb{F}_{p^2}$ be the form $a = a_0 + a_1\alpha$, where α is a root of $x^2 - \beta$. Then, $a^{-1} = (-a_0 + a_1\alpha)/(-a_0^2 + \beta a_1^2)$ in $\mathbb{F}_{p^2}$.*

2) *Let $p \equiv 1 \pmod{12}$ and $x^3 - \beta$ be irreducible over $\mathbb{F}_p$ where β is cubic non-residue in $\mathbb{F}_p$. Let $a \in \mathbb{F}_{p^3}$ be the form $a = a_0 + a_1\alpha + a_2\alpha^2$, where α is a root of $x^3 - \beta$. Then, $a^{-1} = a_2((a_0^2 - \beta a_1a_2) + (-a_0a_1 + \beta a_2^2)\alpha + (-a_0a_2 + a_1^2)\alpha^2)/(-a_0a_1 + \beta a_2^2)$ in $\mathbb{F}_{p^3}$.*

3) *[17] Let $p \equiv \mp 2 \pmod{5}$ and $x^4 + x^3 + x^2 + x + 1$ be irreducible over $\mathbb{F}_p$. Let $a \in \mathbb{F}_{p^4}$ be the form $a = a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4$, where α is a root of $x^4 + x^3 + x^2 + x + 1$. Then, the inverse of a in $\mathbb{F}_{p^4}$ can be defined by $a^{-1} = a_3\alpha + a_2\alpha^2 + a_1\alpha^3 + a_0\alpha^4$. This trick can be used point addition and doubling in Elliptic Curve Cryptosystems [6].*

4) *There is a method originally developed for using with binary extension fields applied to $\mathbb{F}_{p^k}$ [8]. In this method, main idea is to use Frobenius map. Let $\alpha \in \mathbb{F}_{p^k}$. Then, using this algorithm, we get $a^{-1} = a^{s-1}/(a^s)$, where $s = (p^k - 1)/(p - 1)$.*

Table VIII gives inversion timings in $\mathbb{F}_{p^2}$ with MIRACL's method and NTL method. It is seen that MIRACL is strongly faster than NTL. Their inversion implementation method is based on the binomials.

C. Exponentiation

There is a relatively low cost implementation way of exponentiation in $\mathbb{F}_{p^2}$. If the order of subgroup divides $p+1$, Lucas sequences can be used in the exponentiation of even embedding degree. Let P and Q be in $\mathbb{F}_p$. Then, Lucas sequences $\{U_n\}$ and $\{V_n\}$ are two particular solutions of the general second order linear recurrence relation $T_n = PT_{n-1} - QT_{n-2}$ which corresponds to the polynomial equation $x^2 - Px + Q = 0$. Let α and β be the roots of this polynomial. Then, they are defined by

$$U_n = \frac{\alpha^n - \beta^n}{\alpha - \beta}, \quad V_n = \alpha^n + \beta^n$$

where $U_0 = 0$, $U_1 = 1$ and $V_0 = 2$, $V_1 = P$ are the initial conditions. In the light of this sequences, raising any element of $\mathbb{F}_{p^2}$ to a power m is given by the following equation:

$$(a + ib)^m = V_m(2a)/2 + U_m(2a)b\alpha,$$

where $i^2 = \delta$ for some quadratic non-residue $\delta \in \mathbb{F}_p$.

Table IX and Table X show Lucas exponentiation timing with our codes in $\mathbb{F}_{p^2}$ with MIRACL's base field operations and exponentiation timing of NTL module, respectively. Our implementation is faster since NTL does not use Lucas method for $\mathbb{F}_{p^2}$ does square-and-multiply for exponentiation. For bigger exponent k's the difference increases since Lucas method has complexity with $\log(k)$.

We now give the following example related to the Frobenius action, where k is the extension degree and $\alpha \in \mathbb{F}_{p^k}$ is the root of the irreducible polynomial in $\mathbb{F}_p$.

Example 9. 1) *Let $k = 2$, x^2+1 is irreducible over $\mathbb{F}_p$ and $a = a_0 + ia_1 \in \mathbb{F}_{p^2}$, where $a_0, a_1 \in \mathbb{F}_p$ and i quadratic non-residue. Then, $a^p = (a_0 + ia_1)^p = (a_0 - ia_1)$*

2) *Let $k = 2$, $p \equiv 2 \pmod 3$ and $a = a_0\alpha + a_1\alpha^2 \in \mathbb{F}_{p^2}$ be an element of $\mathbb{F}_{p^2}$, where $a_0, a_1 \in \mathbb{F}_p$. Then, the third cyclotomic polynomial is irreducible over $\mathbb{F}_p$. $a^p = (a_0\alpha + a_1\alpha^2)^p = (a_1\alpha + a_0\alpha^2)$.*

3) *Let $k = 4$, $p \equiv \mp 2 \pmod 5$ and $a = a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4 \in \mathbb{F}_{p^4}$, where $a_0, a_1, a_2, a_3 \in \mathbb{F}_p$. Then, the fifth cyclotomic polynomial is irreducible over $\mathbb{F}_p$. $a^p = (a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4)^p = a_2\alpha + a_0\alpha^2 + a_3\alpha^3 + a_1\alpha^4$.*

V. Timings

Tables given in APPENDIX show the timing comparison for extension field arithmetics implemented on MIRACL and GMP-NTL. Both libraries are installed as default installation in which MIRACL is used without its optimizations. For MIRACL we worked with default library ms32.lib. For GMP we run Cygwin as Linux simulator on the same computer, installed NTL based on GMP functions and used NTL's ZZ_pE module for field extension arithmetic.

The performance of arithmetic operations was measured on an Intel Core Duo 2*1,6 GHz (but uses only one core) with 2 GB RAM, running Windows XP SP2. We have coded in C for MIRACL and compiled in Visual C/C++ 6.0. We have used standard ZZ_pE class for GMP/NTL and compiled by GNU C++ Compiler.

We implemented Karatsuba multiplication for $k = 2, 3, 5$ and 10 in MIRACL. We implemented extension degree 10 in two different towering : quadratic over quintic and quintic over quadratic. We used irreducible cyclotomic polynomial for quadratic, field and trinomials for others. Random primes were generated by MIRACL.

In GMP/NTL we used ready class ZZ_pE and its functions. Timings start by generating two random field elements and ends by multiplication of that elements. For exponentiation we implemented Lucas sequences for quadratic exponentiation in MIRACL but used NTL's prepared polynomial powering function. For inversion we used the prepared functions of MIRACL and NTL in quadratic extension.

VI. Conclusions

We have presented arithmetic operations; namely multiplication, exponentiation and inversion in the fields with extension degree 2,3,4,5 and 10. We have used cyclotomic and trinomial irreducible polynomials for representing the extension fields. This representation is of course different than the ones used for pairing-friendly fields. Our first observation is that cyclotomics and trinomials are also as efficient as ones used in pairing-friendly fields. Theoretically, for the multiplication on $\mathbb{F}_{p^k}$, we have shown that the number of arithmetic operations performed on $\mathbb{F}_p$ are comparable with ones corresponding to pairing-friendly fields.

Acknowledgment

We would like to thank Professor Ersan Akyıldız for his valuable comments.

References

[1] IEEE $P1636.3^{TM}$/D1Draft Standard for Identity-based Public-key Cryptography Using Pairings
[2] J.C.B. Bajard, L.Imbert, C.Negre and T. Plantard, " Efficient Multiplication $GF(p^k)$ for Elliptic Curve Cryptography", *Proceedings of the 16^{th} IEEE Symposium on Computer Arithmetic(ARITH'03)*, pp. 182, 2003.
[3] Daniel J. Bernstein. "Multidigit Multiplication for Mathematicians", *Available from http://cr.yp.to/arith.html#m3*, 2001.
[4] D. Boneh, B. Lynn and H. Shacham, "Short Signature form the Weil Pairing", *Advances in Cryptology - ASIACRYPT01, LNCS 2248*, pp. 514-532, 2001.
[5] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", *Advances in cryptology CRYPTO01, LNCS 2139*, pp. 213229, 2001.
[6] M. Brown, D. Hankerson, J Lopez and A. Menezes, "Software Implementation of the NIST elliptic curves over prime fields", *In Proceedings of the 2001 conference on topics in cryptology: the cryptographers track at RSA, LCNS 2020*, pp. 250265, 2001.
[7] A. Devegili, C. Eigeartaigh, M. Scott and R. Dahab, "Multiplication and Squaring on Pairing-Friendly Fields", *IACR ePrint Archive*, `eprint.iacr.org/2006/471.pdf`, 2006.
[8] T. Itoh and S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ Using Normal Bases", *Information and Computation*, Vol.78, pp.171-177, 1988.
[9] A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman", *Proceedings of ANTS 4, LNCS 1838*, pp. 385-394, 2000.
[10] A. Karatsuba and Y. Ofman, "Multiplication of Multidigit Numbers on Automata", *Soviet Physisc Doklary 7(7)*, pp. 595-596, 1963.
[11] D. Knuth, "The Art of Computer Programming", Vol. 2, Third Edition, 1998.
[12] N. Koblitz and A. Menezes, "Pairing-Based Cryptography at High Security Levels", *In Cryptography and Coding - IMA 2005, LNCS 3796 volume 3796*, pp. 13-36, 2005.
[13] A. Lenstra and E. Verheul, "The XTR Public Key System", *CRYPTO'00*, pp. 1-19, 2000.
[14] B. Lynn, PhD. Thesis with the title "On The Implementaton Of Parng-Based Cryptosystems", June 2007.
[15] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *The Annual ACM Symposium on Theory of Computing (STOC)*, ACM Press, pp. 8089, 1991.
[16] A. Miyaji, M. Nakabayashi and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction", *IEICE Transactions on Fundamentals, E84-A(5)*, pp. 1234-1243, 2001.
[17] Y. Park, S. Jeong and J. Lim "Fast Exponentiation in Subgroups of Finite Fields", *Electronics Letters, Vol.38 No.13*, pp. 629-630, 2002.
[18] M. Scott, "Implementing Cryptographic Pairings", 2004.
[19] A. Weimerskirch and C. Paar, "Generalizations of the Karatsuba Algorithm for Efficient Implementations", *IACR ePrint Archive*, `eprint.iacr.org/2006/224.pdf`, 2006.

APPENDIX

TABLE III
MULTIPLICATION IN $\mathbb{F}_{p^2}$ WITH MIRACL LIBRARY VS OUR IMPLEMENTATION VS NTL

	160-bit	192-bit	256-bit	512-bit
Karatsuba (zzn2_mul)	0,003	0,004	0,005	0,011
Our implementation	0,020	0,032	0,041	0,085
NTL (ZZ_pE)	0,019	0,023	0,029	0,064

TABLE IV
MULTIPLICATION IN $\mathbb{F}_{p^3}$ WITH MIRACL LIBRARY VS OUR IMPLEMENTATION VS NTL

	160-bit	192-bit	256-bit	512-bit
Karatsuba (zzn3_mul)	0,020	0,019	0,021	0,039
Our implementation	0,034	0,067	0,095	0,119
NTL (ZZ_pE)	0,027	0,030	0,039	0,081

TABLE V
MULTIPLICATION IN $\mathbb{F}_{p^4}$ WITH OUR IMPLEMENTATION VS NTL

	160-bit	192-bit	256-bit	512-bit
Our implementation	0,063	0,073	0,085	0,128
NTL (ZZ_pE)	0,040	0,049	0,059	0,119

TABLE VI
MULTIPLICATION IN $\mathbb{F}_{p^5}$ WITH OUR IMPLEMENTATION VS NTL

	160-bit	192-bit	256-bit	512-bit
Our implementation	0,069	0,074	0,085	0,128
NTL (ZZ_pE)	0,040	0,065	0,080	0,156

TABLE VII
MULTIPLICATION IN $\mathbb{F}_{p^{10}}$ WITH OUR IMPLEMENTATION (DIFFERENT TOWERINGS) VS NTL

	160-bit	192-bit	256-bit	512-bit
Our implementation (quadratic over quintic)	0,309	0,319	0,346	0,494
Our implementation (quintic over quadratic)	0,411	0,420	0,443	0,576
NTL (ZZ_pE)	0,125	0,140	0,172	0,372

TABLE VIII
INVERSION IN $\mathbb{F}_{p^2}$ WITH MIRACL VS NTL

	160-bit	192-bit	256-bit	512-bit
MIRACL (zzn2_inv)	0,023	0,027	0,036	0,081
NTL (ZZ_pE)	0,073	0,080	0,092	0,170

TABLE IX
LUCAS EXPONENTIATION IN $\mathbb{F}_{p^2}$ WITH OUR IMPLEMENTATION FOR DIFFERENT EXPONENT K VS. NTL

	160-bit	192-bit	256-bit	512-bit
$k = 10$	0,030	0,035	0,044	0,080
$k = 50$	0,033	0,057	0,061	0,134
$k = 200$	0,058	0,069	0,090	0,183
$k = 600$	0,047	0,082	0,107	0,216
$k = 1000$	0,053	0,094	0,115	0,233
$k = 2000$	0,060	0,097	0,124	0,258

TABLE X
EXPONENTIATION IN ZZ_PE NTL FOR DIFFERENT POWERS K

	160-bit	192-bit	256-bit	512-bit
$k = 10$	0,094	0,105	0,127	0,245
$k = 50$	0,153	0,174	0,212	0,422
$k = 200$	0,192	0,219	0,270	0,531
$k = 600$	0,253	0,288	0,351	0,707
$k = 1000$	0,290	0,333	0,412	0,836
$k = 2000$	0,317	0,361	0,443	0,890