The Final Exponentiation in Pairing-Based Cryptography

Barış Bülent Kırlar

Department of Mathematics, Süleyman Demirel University, 32260, Isparta, Turkey
Institute of Applied Mathematics, Middle East Technical University, 06531, Ankara, Turkey
e-mail: bariskirlar@sdu.edu.tr

Abstract—In recent years, there has been many work related to the pairing-based cryptosystems. These systems rely on bilinear non-degenerate maps called pairings, such as Tate pairing defined over elliptic curves. In these systems, there is always a powering of an element to compute. To do this, one can utilize compressed form of the element in the cyclotomic subgroup of the finite fields $\mathbb{F}_{q^k}^*$. Compressed form of field elements also gives rise to define new public key cryptosystems that play an important role in ensuring information security. In this paper, we review how to compute the final powering efficiently. Then we illustrate some algorithms to compute the power of an element in $\mathbb{F}_{q^k}^*$ with $k = 2, 3, 4, 6, 10$ and propose new formulae for $k = 14$. We also show how to define short signature scheme using compressed pairings.

Keywords—exponentiation, compression, pairing-based cryptography.

1. Introduction

From the advent of elliptic curve cryptosystems (ECC), independently by Miller (1985) and Koblitz (1987), elliptic curves have been so much interest from cryptographic researchers. The main reason for the attractiveness of ECC is the fact that there is no sub-exponential algorithm known to solve discrete logarithm problem on a properly chosen elliptic curve. This means that significantly smaller key sizes can be used in ECC than in other systems such as RSA and DSA with equivalent levels of security. Some benefits of having smaller key sizes include faster computations, and reductions in processing power, storage space and bandwidth. This makes ECC ideal for constrained environments such as pagers, PDAs, cellular phones and smart cards.

More recently, elliptic curves have been started to use widely in pairing-based cryptographic protocols. These protocols are based on bilinear pairings, the Weil and Tate pairings derived from certain elliptic curves of embedding degree k with $k \leq 50$. Although, the Weil pairing was initially proposed in [5] as a suitable construction for the realisation of such protocols, it is now usually accepted that the Tate pairing is preferable for its greater efficiency. Efficient computation of the Tate pairing on supersingular elliptic curves and certain ordinary curves that are equally suitable for pairing-based schemes have been suggested in [3], [2], [10]. In fact, ordinary curves offer more flexibility for the choice of security parameters [3], [25].

The computation of Tate pairing and its deriva-

Manuscript received March 15, 2012; revised May 1, 2012.

tives called Eta, Ate and R-Ate pairings [1], [16], [20], [32] consist of two main components, the Miller's algorithm that computes the pairing value and the so-called final exponentiation. The latter is done by raising the output of pairing value to the power of $(q^k - 1)/r$ to get a unique value in the group of r-th roots of unity $G_{r,q,k}$ that is the subgroup of the cyclotomic subgroup $\Phi_k(q)$ in the extension fields $\mathbb{F}^*_{q^k}$. It is well-known that pairing-based cryptographic protocols require these components.

In the recent years, there have been several studies on compressing the elements of certain subgroups of some field extensions. The compression methods fall into two categories in these works. They either use the trace representation of elements or a rational parametrization of algebraic torus. We only consider the trace representation in this work.

In 1994, the first proposal is given by Smith and Skinner [30] using the Lucas sequences. They showed that the elements of a subgroup $G_{r,q,2}$ whose order r divides $\Phi_2(q) = q + 1$ in $\mathbb{F}^*_{q^2}$ could be identified by their traces over $\mathbb{F}_q$. In other words, the elements of $G_{r,q,2}$ can be uniquely identified up to conjugation using the characteristic polynomials over $\mathbb{F}_q$. Moreover, they showed that exponentiation in $G_{r,q,2}$ can be efficiently performed using the trace representation. Their construction provides a compression factor 2.

Gong and Harn [13] showed that the elements of a subgroup $G_{r,q,3}$ whose order r divides $\Phi_3(q) = q^2 + q + 1$ in $\mathbb{F}^*_{q^3}$ could be identified with a compression factor 3/2. They also obtained an efficient exponentiation algorithm for the compressed form of those elements.

Brouwer, Pellikaan and Verheul [8] obtained a factor 3 compression by representing the elements of a subgroup $G_{r,q,6}$ whose order r divides $\Phi_6(q) = q^2 - q + 1$ in $\mathbb{F}^*_{q^6}$ by a pair of elements from $\mathbb{F}_q$. However, they did not give an algorithm to exponentiate the elements of $G_{r,q,6}$ in compressed form. In 2000, Lenstra and Verheul [21] showed that the elements of subgroup $G_{r,q,6}$ whose order r dividing $\Phi_6(q) = q^2 - q + 1$ in $\mathbb{F}^*_{q^6}$ can be uniquely represented by their traces over $\mathbb{F}_{q^2}$. They gave a very efficient exponentiation algorithm in $G_{r,q,6}$ with a factor 3 compression. Verheul et al. in [7] obtained a precise formulation for representations of elements in extension fields of arbitrary degree.

In 2004, Giuliani and Gong [11] obtained a factor 5/2 compression in a subgroup $G_{r,q,10}$ of order r dividing $\Phi_{10}(q) = q^4 - q^3 + q^2 - q + 1$ in $\mathbb{F}^*_{q^{10}}$ using the fifth-order characteristic sequences over $\mathbb{F}_{q^2}$. They obtained an algorithm to exponentiate the compressed form of elements in G and also proposed more efficient algorithm in [12].

More recently, Shirase et al. [27] considered that the elements of subgroup $G_{r,q,6}$ whose order r dividing $q - \sqrt{3q} + 1$ in $\mathbb{F}^*_{q^6}$ where $q = 3^t$ for some odd t and they showed that those elements in $G_{r,q,6}$ can be uniquely represented (up to conjugation) with a compression factor 6 over $\mathbb{F}_q$. They also presented an algorithm for exponentiation of those elements. In 2009, using the same trick in [27], Karabina [18] observed that the elements of order dividing $q \pm \sqrt{3q} + 1$ in $\mathbb{F}^*_{q^6}$ where $q = 3^t$ for some odd t and the elements of order dividing $q \pm \sqrt{2q} + 1$ in $\mathbb{F}^*_{q^4}$ where $q = 2^t$ for some odd t can be uniquely represented by their traces over $\mathbb{F}_q$ with a compression factor 6 and 4, respectively. He presented five exponentiation algorithms for compression factor 4 and six exponentiation algorithms for compression factor 6. Moreover, he compared those exponentiation algorithms.

In [19], we obtained a factor 7/3 compression by representing the elements of a subgroup $G_{r,q,14}$ whose order r divides $\Phi_{14}(q) = q^6 - q^5 + q^4 - q^3 + q^2 - q + 1$ in $\mathbb{F}^*_{q^{14}}$ by a triple of elements from $\mathbb{F}_{q^2}$.

However, we have not found an algorithm yet to exponentiate the elements of $G_{r,q,14}$ in compressed form.

This paper is organized as follows. In Section 2, we review mathematical concepts related to the bilinear pairings, Tate pairing and the final exponentiation. In Section 3, we discuss compressed form of the field elements and provide explicit formulae to represent those elements. We also give the related exponentiation algorithms. In Section 4, we describe compressed pairings used in cryptographic protocols.

2. Mathematical Background

2.1. Bilinear Pairings

Let G_1, G_2 and G_3 be cyclic groups of order n. A bilinear pairing e is an efficiently computable map

$$e : G_1 \times G_2 \longrightarrow G_3$$

such that

- (bilinearity) For all $P \in G_1$, $Q \in G_2$ and $a, b \in \mathbb{Z}$, $e(aP, bQ) = e(P, Q)^{ab}$.
- (non-degeneracy) For all $P \in G_1$ with $P \neq Id_{G_1}$, there is some $Q \in G_2$ such that $e(P, Q) \neq Id_{G_3}$. For all $Q \in G_2$ with $Q \neq Id_{G_2}$, there is some $P \in G_1$ such that $e(P, Q) \neq Id_{G_3}$.

There are many pairing-based protocols using the properties of the bilinear pairings. In this study, we will only give one of the most important protocol which is called short signature scheme by Boneh, Lynn and Shacham (BLS) [6]: Let $P \in G_1$ such that $G_1 =< P >$ of order n.

- **Key Generation :** Pick a random $c \in \mathbb{Z}_n^*$ and compute cP. The secret key is c and the public key is cP.
- **Sign :** Let $H : \{0,1\}^* \to G_1^*$ be a cryptographic hash function. Given a secret key c and a message $m \in \{0,1\}^*$, compute the signature $S = cH(m) \in G_1$.
- **Verify :** Given a public key cP, a message m and a signature S, verify $e(P, S) = e(cP, H(m))$.

Weil and Tate pairings on elliptic curves over finite fields are the classical examples of bilinear pairings used in pairing-based cryptographic protocols. The derivatives of the Tate pairing was defined later which are also used in protocols.

Let E be an elliptic curve over a finite field $\mathbb{F}_q$ of characteristic p with the identity element ∞. Let r be the largest prime divisor of the group of order $n = \#E(\mathbb{F}_q)$ which is co-prime to q. The embedding degree of E is defined to be the smallest integer k such that $r \mid q^k - 1$. Let $P \in E(\mathbb{F}_{q^k})$ such that $rP = \infty$. Then P has an order r and call it an r-torsion point. We denote the group of r-torsion points in $E(\mathbb{F}_{q^k})$ by $E(\mathbb{F}_{q^k})[r]$. Furthermore, $rE(\mathbb{F}_{q^k}) = \{rP \mid P \in E(\mathbb{F}_{q^k})\}$ is a subgroup of $E(\mathbb{F}_{q^k})$ and the quotient group $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ is a group of exponent r. Let $Q \in E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ and consider the divisor $D = (Q + R) - (R)$ with a random point $R \in E(\mathbb{F}_{q^k})$ such that D is co-prime with $(P) - (\infty)$. In the light of the above primitives, we define the Tate pairing in the following map:

$$\begin{array}{rcl} \langle \cdot, \cdot \rangle_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) & \to & \mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^r \\ (P, Q) & \mapsto & f_{r,P}(D) \end{array}$$

It is clear that one can think of the quotient group $\mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^r$ as the set of equivalence classes of $\mathbb{F}_{q^k}^*$ under the equivalence relation $a \equiv b$ if and only if there exists $c \in \mathbb{F}_{q^k}^*$ such that $a = bc^r$. This means that the Tate pairing is only defined up to a multiple by an r-th power in $\mathbb{F}_{q^k}^*$. It is necessary to get a unique value for most applications in cryptography. Therefore, the reduced Tate pairing is defined as

$$e_r(P, Q) = f_{r,P}(D)^{(q^k - 1)/r}.$$

Here, raising the output $f_{r,P}(D)$ to the power of $(q^k-1)/r$ is known as the final exponentiation. This operation is also used for the derivatives of the Tate pairing. The Tate pairing is computed by Miller's algorithm [24], efficiently. For a more detailed background, one can refer to the chapter 9 of [4].

2.2. The Final Powering

The exponentiation needed by the reduced Tate pairing (and its derivatives) $e_r(P,Q) = f_{r,P}(D)^{(q^k-1)/r}$ has been efficiently computed for supersingular elliptic curves with embedding degree $k = 2,4,6$ in [2]. Later, this is carried out in [23] as we explain now : Assume that the Tate pairing value obtained by Miller's algorithm is a. We will now exponentiate a by $\frac{q^k-1}{r}$. To do this, we first write

$$q^k - 1 = \prod_{d|k} \Phi_d(q).$$

Since k is the embedding degree, r has to divide cyclotomic polynomial $\Phi_k(q)$ (not to smaller degree of it). We compute $b = a^c$, where

$$c = \prod_{d|k, d<k} \Phi_d(q).$$

Since $\frac{\Phi_k(q)}{r}$ is an integer, we obtain the output $b^{\Phi_k(q)/r}$ using a standard exponentiation algorithm. We note that this method is faster than the previous approach given for supersingular elliptic curves in [2].

Example 1: Let $\mathbb{F}_{q^2} = \mathbb{F}(\alpha) \cong \mathbb{F}_q[x]/ < x^2 - \delta >$. Then we can write an element $a \in \mathbb{F}_{q^2}$ in the form $a = u + \alpha v$, where $u, v \in \mathbb{F}_q$ and $\alpha^2 = \delta$. It is clear that $q^2 - 1 = \Phi_2(q)\Phi_1(q)$ and $r \mid \Phi_2(q) = q + 1$. We have

$$b^{(q+1)/r} = (a^{q-1})^{(q+1)/r} = (\frac{u - \alpha v}{u + \alpha v})^{(q+1)/r}.$$

Since

$$b = \frac{u - \alpha v}{u + \alpha v} = \frac{u^2 - v^2}{u^2 + v^2} - \alpha \frac{2uv}{u^2 + v^2},$$

the field element b becomes "unitary" ([17], [28]). In other words, $b\bar{b} = 1$, where $\bar{b}$ is the conjugate of b in $\mathbb{F}_{q^2}$. Then, we compute $b^{(q+1)/r}$ using a standard exponentiation algorithm to obtain $a^{(q^2-1)/r}$. As a result, we have effectively halved the size of the final powering using this approach.

Remark 1: For $k = 2d$, the final exponent can be written by

$$\frac{q^k - 1}{r} = (q^d - 1)\frac{(q^d + 1)}{\Phi_d(q)}\frac{\Phi_d(q)}{r}.$$

After raising to the power of $q^d - 1$, the field element becomes unitary. This property gives us two important implications:

(i) squaring of unitary elements is significantly cheaper than squaring of non-unitary elements.
(ii) for unitary elements, any future inversions can be implemented by simple conjugation.

3. Compression in Finite Fields

We first describe a method to represent elements of cyclotomic subgroups in $\mathbb{F}_{q^k}$ with fewer bits. This is so called compressed form of those elements in $\mathbb{F}_{q^k}$. A cyclotomic subgroup $G_{r,q,k}$ in $\mathbb{F}_{q^k}$ is defined to be a subgroup of prime order r with $r \mid \Phi_k(q)$ and $r \nmid k$. We now show that the relationships between coefficients of minimal polynomials for the elements of a cyclotomic subgroup $G_{r,q,k}$ and the corresponding linear recurrence relation. For more details and proofs, we refer the reader to look at [7].

Let α be an element of a cyclotomic subgroup $G_{r,q,k}$, where $k \geq 2$. Let

$$f_\alpha(x) = x^n - a_1 x^{n-1} + \cdots + (-1)^{n-1} a_{n-1} x + (-1)^n a_n$$

be the minimal polynomial of α over $\mathbb{F}_{q^d}$ for some d dividing k with $n = k/d > 1$. Then $a_n = 1$. It is clear that for $1 \leq i \leq n$, a_i's are the elementary symmetric functions in variables $\alpha, \alpha^q, \ldots, \alpha^{q^{n-1}}$, namely

$$a_1 = \sum_{i=0}^{n-1} \alpha^{q^i},\ a_2 = \sum_{i<j} \alpha^{q^i+q^j}, \ldots,\ a_n = \prod_{i=0}^{n-1} \alpha^{q^i}.$$

The polynomial $f_\alpha(x)$ allows us to introduce the n-th order linear recurrence relation $\{s_i\}$ which is defined by

$$s_t = a_1 s_{t-1} - a_2 s_{t-2} + \cdots - (-1)^n s_{t-n}, \quad t \geq n.$$

The sequence $\{s_i\}$ of elements in $\mathbb{F}_{q^d}$ with fixed initial conditions

$$s_i = Tr(\alpha^i) = \alpha^i + \alpha^{iq} + \ldots + \alpha^{iq^{n-1}}$$

for $i = 0, \cdots, n-1$ is called the n-th order characteristic sequence over $\mathbb{F}_{q^d}$ generated by α. Here, $Tr : \mathbb{F}_{q^n} \to \mathbb{F}_q$ is the Trace map.

For any integer m, the minimal polynomial of α^m is

$$\begin{aligned} f_{\alpha^m}(x) &= x^n - a_{1,m}x^{n-1} + a_{2,m}x^{n-2} - \cdots \\ &\quad +(-1)^{n-1}a_{n-1,m}x + (-1)^n, \end{aligned}$$

whose roots are α^{mq^i} for $i = 0, \cdots, n-1$. Hence, we may represent α^m (and its conjugates) by the set $\{a_{1,m}, a_{2,m}, \cdots, a_{n-1,m}\}$, where the elements are written by

$$a_{i,m} = \sum_{0 \leq j_1 \leq \cdots \leq j_i \leq n-1} \alpha^{m(q^{j_1}+q^{j_2}+\cdots+q^{j_i})}.$$

It follows from the equation above $a_{i,m} = a_{n-i,-m}$. The Newton's Formula [22] tells us that for any $i \in \{1, \cdots, n-1\}$, we can efficiently obtain $\{a_{1,m}, a_{2,m}, \cdots, a_{i,m}\}$ from the set $\{s_m, s_{2m}, \cdots, s_{im}\}$ and vice-versa using the following equalities:

$$\begin{aligned} s_{im} &= a_{1,m}s_{(i-1)m} - a_{2,m}s_{(i-2)m} + \cdots \\ &\quad -(-1)^i i a_{i,m} \\ a_{i,m} &= i^{-1}\big((-1)^{i+1}s_{im} + \cdots + a_{i-1,m}s_m\big) \end{aligned}$$

In this section, our goal is to obtain shorter representation of α^m. Thus, we now descibe two significant cases:

1 If $k = 2l$ is even, then $\alpha \in \mathbb{F}_{q^{2l}}$ has order dividing q^l+1. This implies that $\alpha^{mq^l} = \alpha^{-m}$. Therefore, for $i = 1, \cdots, n-1$, we have

$$\begin{aligned} a_{n-i,m} &= a_{i,-m} \\ &= \sum_{0 \leq j_1 \leq \cdots \leq j_i \leq n-1} \alpha^{-m(q^{j_1}+q^{j_2}+\cdots+q^{j_i})} \\ &= \sum_{0 \leq j_1 \leq \cdots \leq j_i \leq n-1} \alpha^{mq^l(q^{j_1}+q^{j_2}+\cdots+q^{j_i})} \\ &= a_{i,m}^{q^l} \end{aligned}$$

Hence, we may represent α^m (and its conjugates) by the set $\{a_{1,m}, \cdots, a_{(n-1)/2,m}\}$.

2 If $k = 2l$ with $d \mid l$, then we have $a_{n-i,m} = a_{i,m}^{q^l}$ for $i = 1, \cdots, n-1$ from the previous result. Since $d \mid l$, i.e., $n = k/d$ is even, we obtain $a_{i,m}^{q^l} = a_{i,m}$ and the result follows. Hence, we may represent α^m (and its conjugates) by the set $\{a_{1,m}, \cdots, a_{n/2,m}\}$.

Lemma 1: [7] Let $k = de$, with $e > 1$. Then for any element α of a cyclotomic subgroup $G_{r,q,k}$ and for any integer m, α^m can be represented by the following number of elements, $\#R_{\alpha^m}$, in $\mathbb{F}_{q^d}$:

$$\#R_{\alpha^m} = \begin{cases} e-1, & \text{if } de \text{ is odd} \\ \frac{e-1}{2}, & \text{if } d \text{ is even and } e \text{ is odd} \\ \frac{e}{2}, & \text{if } e \text{ is even} \end{cases}$$

3.1. Compression Factor 2

Let α be any element of $G_{r,q,2}$ in $\mathbb{F}^*_{q^2}$ and let

$$f_\alpha(x) = x^2 - a_1 x + 1$$

be the minimal polynomial of α over $\mathbb{F}_q$. The polynomial $f_\alpha(x)$ allows us to introduce the second-order linear recurrence relation $\{s_i\}$ which is defined by

$$s_{t+2} = a_1 s_{t+1} - s_t.$$

For any integer m, the minimal polynomial of α^m over $\mathbb{F}_q$ is

$$f_{\alpha^m}(x) = x^2 - a_{1,m}x + 1,$$

where $a_{1,m} = s_m = Tr(\alpha^m) = \alpha^m + \alpha^{mq}$. The sequence $\{a_{1,m}\}$ which is so called Lucas sequence is defined by the following recurrence relations:

$$a_{1,0} = 2,\ a_{1,1} = a_1,\ a_{1,u+1} = a_1 a_{1,u} - a_{1,u-1}.$$

Smith and Skinner [30] showed that the elements of $G_{r,q,2}$ in $\mathbb{F}^*_{q^2}$ could be identified by $\{a_{1,m}\}$ only $1/2$ as many as in the ordinary case. More precisely, the elements of $G_{r,q,2}$ can be uniquely determined by their traces over $\mathbb{F}_q$. This construction yields a compression factor 2.

The Lucas sequence $\{a_{1,m}\}$ can be efficiently computed in Algorithm 1 depending on the following properties

$$\begin{aligned} a_{1,u+v} &= a_{1,u}a_{1,v} - a_{1,u-v} \\ a_{1,2u} &= a_{1,u}^2 - 2 \end{aligned}$$

for $u, v \in \mathbb{Z}$.

Algorithm 1 Computing Lucas Sequence

```
Require: $a_1 \in \mathbb{F}_q$ and $m = \sum_{j=0}^{t} m_j 2^j \in \mathbb{Z}^+$ with $m_t = 1$
Ensure: $(a_{1,m}, a_{1,m+1})$
  $(a_{1,y}, a_{1,y+1}) \leftarrow (2, a_1)$
  for $j \leftarrow t$ to $0$ do
      if $m_j = 1$ then
          $a_{1,y} \leftarrow a_{1,y}a_{1,y+1} - a_1, \quad a_{1,y+1} \leftarrow a_{1,y+1}^2 - 2$
      else
          $a_{1,y} \leftarrow a_{1,y}^2 - 2, \quad a_{1,y+1} \leftarrow a_{1,y}a_{1,y+1} - a_1$
      end if
  end for
  return $(a_{1,y}, a_{1,y+1})$
```

Yen and Laih [31] developed the algorithms using the property of Lucas sequence given above that could also be derived from the equations 3.8 and 3.9 in [29]. Algorithm 1 is based on the left-to-right scanning approach and the other which is not given in this study is based on the right-to-left scanning approach. The computational cost of both algorithms is $2 + 2\lfloor \log_2 m \rfloor$ multiplications in $\mathbb{F}_q$, where $\lfloor\ \rfloor$ denotes the greatest integer function. However, right-to-left scanning one requires more temporary memories.

3.2. Compression Factor 3/2

Let α be any element of $G_{r,q,3}$ in $\mathbb{F}^*_{q^3}$ and let

$$f_\alpha(x) = x^3 - a_1x^2 + a_2x - 1$$

be the minimal polynomial of α over $\mathbb{F}_q$. The polynomial $f_\alpha(x)$ allows us to introduce the third-order linear recurrence relation $\{s_i\}$ which is defined by

$$s_{t+3} = a_1 s_{t+2} - a_2 s_{t+1} + s_t.$$

For any integer m, the minimal polynomial of α^m over $\mathbb{F}_q$ is

$$f_{\alpha^m}(x) = x^3 - a_{1,m}x^2 + a_{1,-m}x - 1,$$

where $a_{1,m} = s_m = Tr(\alpha^m) = \alpha^m + \alpha^{mq} + \alpha^{mq^2}$.

Gong and Harn [13] showed that the elements of $G_{r,q,3}$ in $\mathbb{F}^*_{q^3}$ could be identified by $\{a_{1,m}, a_{1,-m}\}$ with a compression factor 3/2 using the above procedure. They also obtained an efficient exponentiation algorithm for the compressed form of those elements in Algorithm 2. To do this, they used the following relations

$$\begin{aligned} a_{1,u+v} &= a_{1,u}a_{1,v} - a_{1,u-v}a_{1,-v} + a_{1,u-2v} \\ a_{1,2u} &= a_{1,u}^2 - 2a_{1,-u} \end{aligned}$$

for $u, v \in \mathbb{Z}$.

Algorithm 2 Computing the third-order sequences $\{a_{1,m}\}$ and $\{a_{1,-m}\}$

```
Require: $a_1, a_2 \in \mathbb{F}_q$ and $m = \sum_{j=0}^{t} m_j 2^{t-j} \in \mathbb{Z}^+$ with $T_0 = m_0 = 1$, $T_i = m_i + 2T_{i-1}$ for $1 \leq i \leq t$ and so $T_t = m$
Ensure: $(a_{1,m-1}, a_{1,m}, a_{1,m+1})$
$(a_{1,T_i-1}, a_{1,T_i}, a_{1,T_i+1}) \leftarrow (3, a_1, a_1^2 - 2a_2)$
for $j \leftarrow 0$ to $t$ do
    if $m_j = 1$ then
        $a_{1,T_i-1} \leftarrow a_{1,T_{i-1}}^2 - 2a_{1,-T_{i-1}}$
        $a_{1,T_i} \leftarrow a_{1,T_{i-1}} a_{1,T_{i-1}+1} - a_1 a_{1,-T_{i-1}} + a_{1,-(T_{i-1}-1)}$
        $a_{1,T_i+1} \leftarrow a_{1,T_{i-1}+1}^2 - 2a_{1,-(T_{i-1}+1)}$
    else
        $a_{1,T_i-1} \leftarrow a_{1,T_{i-1}} a_{1,T_{i-1}-1} - a_2 a_{1,-T_{i-1}} + a_{1,-(T_{i-1}+1)}$
        $a_{1,T_i} \leftarrow a_{1,T_{i-1}}^2 - 2a_{1,-T_{i-1}}$
        $a_{1,T_i+1} \leftarrow a_{1,T_{i-1}} a_{1,T_{i-1}+1} - a_1 a_{1,-T_{i-1}} + a_{1,-T_{i-1}-1}$
    end if
end for
return $(a_{1,T_i-1}, a_{1,T_i}, a_{1,T_i+1})$
```

In Algorithm 2, to compute $\{a_{1,-m}\}$, T_i and T_{i-1} should be respectively replaced by $-T_i$ and $-T_{i-1}$. Using Algorithm 2 to calculate a pair of the m-th term $a_{1,m}$ and $a_{1,-m}$ needs $9 \log_2 m$ multiplications in $\mathbb{F}_q$ on the average. This method is more efficient than Fiduccia's algorithm using modulo polynomial in [9]. Gong, Harn and Wu [14] also proposed much more efficient algorithm that utilizes the signed-digit representation. Using this representation to calculate a pair of the m-th term $a_{1,m}$ and $a_{1,-m}$ needs $4 \log_2 m$ multiplications in $\mathbb{F}_q$ on the average. This method is optimized where q is a prime or a power of prime p in [15].

3.3. Compression Factor 3

Let α be any element of $G_{r,q,6}$ in $\mathbb{F}_{q^6}^*$. Then the conjugates over $\mathbb{F}_{q^2}$ of $\alpha \in \mathbb{F}_{q^6}$ are α, α^{q^2} and α^{q^4}. Therefore, we obtain $a_1 = Tr_{\mathbb{F}_{q^6}/\mathbb{F}_{q^2}}(\alpha) = \alpha^m + \alpha^{mq^2} + \alpha^{mq^4}$. The conjugates of $\alpha \in G_{r,q,6}$ are also α, α^{q-1} and α^{-q} because $q^2 \equiv q-1 \pmod{q^2-q+1}$ and $q^4 \equiv -q \pmod{q^2 - q + 1}$. This implies that the second elementary symmetric function is $a_2 = \alpha\alpha^{q^2} + \alpha\alpha^{q^4} + \alpha^{q^2}\alpha^{q^4} = a_1^q$. Therefore, the minimal polynomial of α over $\mathbb{F}_{q^2}$ is

$$f_\alpha(x) = x^3 - a_1 x^2 + a_1^q x - 1.$$

For any integer m, the conjugates of α^m are the roots of the polynomial

$$f_{\alpha^m}(x) = x^3 - a_{1,m} x^2 + a_{1,m}^q x - 1$$

over $\mathbb{F}_{q^2}$. The latter polynomial is fully determined by $\{a_{1,m}\}$.

Lenstra and Verheul introduced XTR cryptosystem in [21] using the above procedure. They showed that the elements of $G_{r,q,6}$ in $\mathbb{F}_{q^6}^*$ could be identified by $\{a_{1,m}\}$ over $\mathbb{F}_{q^2}$ with a compression factor 3.

The XTR exponentiation $\{a_{1,m}\}$ can be efficiently computed in Algorithm 3 ([21], Algorithm 2.3.7) using the following relations

$$\begin{aligned} a_{1,u+v} &= a_{1,u} a_{1,v} - a_{1,v}^q a_{1,u-v} + a_{1,u-2v} \\ a_{1,2u} &= a_{1,u}^2 - 2a_{1,u}^q \end{aligned}$$

for $u, v \in \mathbb{Z}$.

Algorithm 3 Computing XTR exponentiation

```
Require: $a_1 \in \mathbb{F}_{q^2}$ and $m = \sum_{j=0}^{t} m_j 2^j \in \mathbb{Z}^+$ with $m_t = 1$
Ensure: $(a_{1,2m}, a_{1,2m+1}, a_{1,2m+2})$
$(a_{1,y-1}, a_{1,y}, a_{1,y+1}) \leftarrow (3, a_1, a_1^2 - 2a_1^q)$
for $j \leftarrow t$ to $0$ do
    if $m_j = 1$ then
        $a_{1,y-1} \leftarrow a_{1,y}^2 - 2a_{1,y}^q$
        $a_{1,y} \leftarrow a_{1,y+1} a_{1,y} - a_{1,y}^q a_1 + a_{1,y-1}^q$
        $a_{1,y+1} \leftarrow a_{1,y+1}^2 - 2a_{1,y+1}^q$
    else
        $a_{1,y-1} \leftarrow a_{1,y-1}^2 - 2a_{1,y-1}^q$
        $a_{1,y} \leftarrow a_{1,y-1} a_{1,y} - a_{1,y}^q a_1^q + a_{1,y+1}^q$
        $a_{1,y+1} \leftarrow a_{1,y}^2 - 2a_{1,y}^q$
    end if
end for
return $(a_{1,y-1}, a_{1,y}, a_{1,y+1})$
```

Lenstra and Verheul [21] showed that the computational cost of the XTR exponentiation $\{a_{1,m}\}$ was $8\log_2 m$ multiplications in $\mathbb{F}_q$.

3.4. Compression Factor 5/2

Let α be any element of $G_{r,q,10}$ in $\mathbb{F}^*_{q^{10}}$ and let

$$f_\alpha(x) = x^5 - a_1x^4 + a_2x^3 - a_2^px^2 + a_1^px - 1$$

be the minimal polynomial of α over $\mathbb{F}_{q^2}$. The polynomial $f_\alpha(x)$ allows us to introduce the fifth-order linear recurrence relation $\{s_i\}$ which is defined by

$$s_{t+5} = a_1s_{t+4} - a_2s_{t+3} + a_2^ps_{t+2} - a_1^ps_{t+1} + s_t.$$

For any integer m, the minimal polynomial of α^m over $\mathbb{F}_{q^2}$ is

$$f_{\alpha^m}(x) = x^5 - a_{1,m}x^4 + a_{2,m}x^3 - a_{2,m}^px^2 + a_{1,m}^px - 1,$$

where

$$a_{1,m} = s_m = \mathrm{Tr}_{\mathbb{F}_{q^{10}}/\mathbb{F}_{q^2}}(\alpha^m) = \sum_{i=0}^{4} \alpha^{mq^{2i}},$$

with the initial conditions $a_{1,-1} = a_1^p$, $a_{1,0} = 5$, $a_{1,1} = a_1$, $a_{1,2} = a_1^2 - 2a_2$, $a_{1,3} = a_1^3 - 3a_1a_2 + 3a_2^p$, and

$$\begin{aligned} a_{2,m} &= \mathrm{Tr}_{\mathbb{F}_{q^{10}}/\mathbb{F}_{q^2}}\left(\alpha^{m(q^2+1)} + \alpha^{m(q^4+1)}\right) \\ &= \sum_{0\le i<j\le 4} \alpha^{mq^{2i}+mq^{2j}}. \end{aligned}$$

Giuliani and Gong [11] showed that the elements of $G_{r,q,10}$ in $\mathbb{F}^*_{q^{10}}$ could be identified by $\{a_{1,m}, a_{2,m}\}$ over $\mathbb{F}_{q^2}$ with a compression factor 5/2 using the above procedure. They obtained an algorithm, Algorithm 4, to compute the compressed form of those elements using the following fact.

Lemma 2: For all integers u and v, we have the following:

(1) $a_{1,2u} = a_{1,u}^2 - 2a_{2,u}$

(2) $a_{2,2u} = a_{2,u}^2 + 2a_{1,u}^p - 2a_{1,u}a_{2,u}^p$

(3) $a_{1,3u} = a_{1,u}^3 - 3a_{1,u}a_{2,u} + 3a_{2,u}^p$

(4) $a_{2,3u} = a_{2,u}^3 - 3a_{1,u}^pa_{2,u} - 3a_{1,u}a_{2,u}a_{2,u}^p$
$\qquad +3a_{1,u}^2a_{1,u}^p + 3a_{2,u}^{2p} - 3a_{1,u}$

(5) $a_{1,u+v} = a_{1,u}a_{1,v} - a_{1,u-v}a_{2,v} + a_{1,u-2v}a_{2,v}^p$
$\qquad -a_{1,u-3v}a_{1,v}^p + a_{1,u-4v}$

(6) $3a_{2,u+v} = a_{1,u}^pa_{2,u-v} - a_{2,u}a_{2,v} + a_{1,u}a_{1,v}a_{1,u+v}$
$\qquad -a_{1,u-2v}a_{1,u-v} + a_{1,2u-3v}$
$\qquad -a_{1,u+2v}a_{1,u} - a_{1,2u+v}a_{1,v} + a_{1,u+v}^2$

Algorithm 4 Computing the fifth-order sequences $\{a_{1,m}\}$ and $\{a_{2,m}\}$

```
Require: $a_1, a_2 \in \mathbb{F}_{q^2}$ and $m = \sum_{j=0}^{n} c_j 3^j$ with
    $c_j = \{-1, 0, 1\}$
Ensure: $(a_{1,m}, a_{2,m})$
$a_{1,y} \leftarrow (a_{1,-1}, a_{1,0}, a_{1,1}, a_{1,2}, a_{1,3})$
$a_{2,y} \leftarrow (a_{2,-1}, a_{2,0}, a_{2,1}, a_{2,2}, a_{2,3})$
$l \leftarrow 1$
for $i \leftarrow n-1$ to $0$ do
  if $c_i = -1$ then
    $a_{1,y} \leftarrow (a_{1,3l-3}, a_{1,3l-2}, a_{1,3l-1}, a_{1,3l}, a_{1,3l+1})$
    $a_{2,y} \leftarrow (a_{2,3l-3}, a_{2,3l-2}, a_{2,3l-1}, a_{2,3l}, a_{2,3l+1})$
  end if
  if $c_i = 0$ then
    $a_{1,y} \leftarrow (a_{1,3l-2}, a_{1,3l-1}, a_{1,3l}, a_{1,3l+1}, a_{1,3l+2})$
    $a_{2,y} \leftarrow (a_{2,3l-2}, a_{2,3l-1}, a_{2,3l}, a_{2,3l+1}, a_{2,3l+2})$
  end if
  if $c_i = 1$ then
    $a_{1,y} \leftarrow (a_{1,3l-1}, a_{1,3l}, a_{1,3l+1}, a_{1,3l+2}, a_{1,3l+3})$
    $a_{2,y} \leftarrow (ta_{2,3l-1}, a_{2,3l}, a_{2,3l+1}, a_{2,3l+2}, a_{2,3l+3})$
  end if
  $l \leftarrow 3l + c_i$
end for
return $(a_{1,y}, a_{2,y})$
```

In Algorithm 4, they showed that a pair of the m-th term $a_{1,m}$ and $a_{2,m}$ needs $108.5\log_2 m$ multiplications in $\mathbb{F}_q$ on the average. Later, Quoos and Mjølsnes gave an algorithm for computing these sequences in [26]. The computational cost of their algorithm was found to be $102\log_2 m$ multiplications in $\mathbb{F}_q$. Giuliani and Gong [12] also proposed a new algorithm called the Diagonal Double-Add (DDA) algorithm to calculate the m-th term of the sequences $\{a_{1,m}\}$ and $\{a_{2,m}\}$. This algorithm is more efficient than the previous one and Fiduccia's algorithm that needs $74\log_2 m$ multiplications in $\mathbb{F}_q$.

3.5. Compression Factor 4 and 6

Let $q = 3^t$ for any odd integer t, i.e. $t = 2l + 1$. Then $\sqrt{3q} = 3^{l+1}$ is an integer and

$$q^2 - q + 1 = (q + \sqrt{3q} + 1)(q - \sqrt{3q} + 1).$$

Shirase et al. introduced the improved version of XTR [27]. They considered that the elements of $G_{r,q,6}$ with $r \mid q - \sqrt{3q} + 1$ in $\mathbb{F}^*_{q^6}$ and showed that those elements can be uniquely represented $\{a_{1,m}\}$ (up to conjugation over $\mathbb{F}_q$) with a compression factor 6. They also presented an exponentiation algorithm of those elements using an analogue of the Lenstra-Verheul algorithm and the following equalities

$$\begin{aligned} a_{1,m} &= Tr_{\mathbb{F}_{q^6}/\mathbb{F}_q}(\alpha^m) = \alpha^m + \alpha^{mq} + \cdots + \alpha^{mq^5}, \\ b_{1,m} &= Tr_{\mathbb{F}_{q^6}/\mathbb{F}_{q^2}}(\alpha^m) = \alpha^m + \alpha^{mq^2} + \alpha^{mq^4}, \end{aligned}$$

where $a_{1,m} = b_{1,m} + b^q_{1,m}$. They showed that computing $a_{1,m}$ takes about $8\log_2 m + 2\lfloor \log_2(t-1) \rfloor + HW(t-1) + 2$ multiplications in $\mathbb{F}_q$, where $\lfloor\ \rfloor$ and HW denote the greatest integer function and Hamming weight function, respectively. If $a_{1,m}$ is given, then $b_{1,m}$ can be obtained efficiently using the following polynomial

$$x^2 - a_{1,m}x + a^{\sqrt{3q}}_{1,m}.$$

In fact, $b_{1,m}$ and $b^q_{1,m}$ are the roots of the above polynomial.

Later, using the same trick in [27], Karabina [18] showed that the elements of $G_{r,q,6}$ with $r \mid q \mp \sqrt{3q} + 1$ in $\mathbb{F}^*_{q^6}$, where $q = 3^{2l+1}$ can be uniquely represented by their traces over $\mathbb{F}_q$ with a compression factor 6. He presented six exponentiation algorithms and compared them. The first works directly using the following polynomial

$$\begin{aligned} f(x) \;=\; & x^6 - a_{1,m}x^5 + (a^t_{1,m} + a_{1,m})x^4 \\ & -(a^2_{1,m} + a^t_{1,m} + 2)x^3 + (a^t_{1,m} + a_{1,m})x^2 \\ & -a_{1,m}x + 1, \end{aligned}$$

where $t = \mp 3^{l+1}$ is the trace of the Frobenius. This algorithm is 59% faster than the algorithm proposed by Shirase et al. in [27].

He also achieved compression factor 4 for the subgroups $G_{r,q,6}$ with $r \mid q \mp \sqrt{2q} + 1$ in $\mathbb{F}^*_{q^4}$, where $q = 2^t$ for some odd t, i.e., $t = 2l + 1$. He presented five exponentiation algorithms for compression factor 4 and compared them. His first algorithm works directly with the polynomial

$$f(x) = x^4 + a_{1,m}x^3 + a^t_{1,m}x^2 + a_{1,m}x + 1,$$

where $t = \mp 2^{l+1}$.

3.6. Compression Factor 7/3

We consider the elements of $G_{r,q,14}$ in $\mathbb{F}^*_{q^{14}}$ and showed that any positive power m of those elements could be identified by $\{a_{1,m}, a_{2,m}, a_{3,m}\}$ over $\mathbb{F}_{q^2}$ with a compression factor 7/3 in [19]. Namely, let α be any element of $G_{r,q,14}$ in $\mathbb{F}^*_{q^{14}}$ and let

$$f_\alpha(x) = x^7 - a_1x^6 + a_2x^5 - a_3x^4 + a^p_3x^3 - a^p_2x^2 + a^p_1x - 1$$

be the minimal polynomial of α over $\mathbb{F}_{q^2}$. For any integer m, the minimal polynomial of α^m over $\mathbb{F}_{q^2}$ is

$$\begin{aligned} f_{\alpha^m}(x) \;=\; & x^7 - a_{1,m}x^6 + a_{2,m}x^5 - a_{3,m}x^4 \\ & + a^p_{3,m}x^3 - a^p_{2,m}x^2 + a^p_{1,m}x - 1, \end{aligned}$$

where

$$a_{1,m} = s_m = \mathrm{Tr}_{\mathbb{F}_{q^{14}}/\mathbb{F}_{q^2}}(\alpha^m) = \sum_{i=0}^{6} \alpha^{mq^{2i}},$$

$$\begin{aligned} a_{2,m} &= \mathrm{Tr}_{\mathbb{F}_{q^{14}}/\mathbb{F}_{q^2}}\Big(\alpha^{m(q^2+1)}+\alpha^{m(q^4+1)}+\alpha^{m(q^6+1)}\Big) \\ &= \sum_{0\le i<j\le 6}\alpha^{mq^{2i}+mq^{2j}}, \end{aligned}$$

and

$$\begin{aligned} a_{3,m} &= \mathrm{Tr}_{\mathbb{F}_{q^{14}}/\mathbb{F}_{q^2}}\Big(\alpha^{m(q^4+q^2+1)}+\alpha^{m(q^6+q^2+1)} \\ &\quad +\alpha^{m(q^8+q^2+1)}+\alpha^{m(q^8+q^4+1)} \\ &\quad +\alpha^{m(q^{10}+q^2+1)}\Big) \\ &= \sum_{0\le i<j<k\le 6}\alpha^{mq^{2i}+mq^{2j}+mq^{2k}}. \end{aligned}$$

We have the following recurrence relations related to the sequences $\{a_{1,m}\}$, $\{a_{2,m}\}$ and $\{a_{3,m}\}$, but we could not find yet any efficient polynomial time algorithm to compute the m-th term of these sequences.

Lemma 3: For all integers u and v, we have the following:

(1) $a_{1,2u} = a_{1,u}^2 - 2a_{2,u}$

(2) $a_{2,2u} = a_{2,u}^2 + 2a_{3,u}^p - 2a_{1,u}a_{3,u}$

(3) $a_{3,2u} = a_{3,u}^2 - 2a_{1,u}^p + 2a_{1,u}a_{2,u}^p - 2a_{2,u}a_{3,u}^p$

(4) $a_{1,3u} = a_{1,u}^3 - 3a_{1,u}a_{2,u} + 3a_{3,u}$

(5) $$\begin{aligned} a_{1,u+v} &= a_{1,u}a_{1,v} - a_{1,u-v}a_{2,v} + a_{1,u-2v}a_{3,v} \\ &\quad -a_{1,u-3v}a_{3,v}^p + a_{1,u-4v}a_{2,v}^p \\ &\quad -a_{1,u-5v}a_{1,v}^p + a_{1,u-6v} \end{aligned}$$

(6) $$\begin{aligned} a_{2,u+v} &= a_{2,u}a_{2,v} - a_{3,v}^p a_{2,u-v} - a_{1,v}^p a_{2,u-2v} \\ &\quad +(a_{1,u-2v}a_{1,u-v} - a_{1,2u-3v})a_{2,v}^p \\ &\quad +a_{1,u-v}a_{1,u-4v} + a_{1,u-2v}a_{1,u-3v} \\ &\quad +(a_{1,2u-4v} - a_{1,u-v}a_{1,u-3v})a_{1,u}^p \\ &\quad -a_{1,u}a_{1,v}a_{1,u+v} + a_{1,v}a_{1,2u+v} \\ &\quad +a_{1,u}a_{1,u+2v} - 2a_{1,2u+2v} \\ &\quad +a_{1,u+v}^2 - 2a_{1,2u-5v} \end{aligned}$$

(7) $$\begin{aligned} a_{3,u+v} &= a_{3,u}a_{3,v} - a_{1,v}^p a_{3,u-v} + a_{1,u-2v}a_{2,u-v} \\ &\quad +(2a_{1,u+v} - a_{1,u}a_{1,v})a_{2,u+v} \\ &\quad +(a_{1,v}a_{1,u+2v} - a_{1,u+3v})a_{2,u} \\ &\quad +(a_{1,u}a_{1,2u+v} - a_{1,3u+v})a_{2,v} \\ &\quad -a_{1,u-v}a_{1,2u-3v} + a_{1,3u-4v} \\ &\quad +2a_{1,u}a_{1,2u+3v} + 2a_{1,v}a_{1,3u+2v} \\ &\quad +2(a_{1,u+v} - a_{1,u}a_{1,v})a_{1,2u+2v} \\ &\quad +a_{1,u}a_{1,v}a_{1,u+v}^2 - a_{1,u}a_{1,u+v}a_{1,u+2v} \\ &\quad -a_{1,v}a_{1,u+v}a_{1,2u+v} + a_{1,u+2v}a_{1,2u+v} \\ &\quad -a_{1,u+v}a_{2,u}a_{2,v} - 3a_{1,3u+3v} - a_{1,u+v}^3 \end{aligned}$$

Proof.

$$\begin{aligned} a_{1,u}^2 &= (\sum_{i=0}^{6}\alpha^{uq^i})^2 = (\sum_{i=0}^{6}\alpha^{2uq^i}) + 2(\sum_{0\le i<j\le 6}\alpha^{u(q^i+q^j)}) \\ &= a_{1,2u} + 2a_{2,u} \\ a_{2,u}^2 &= (\sum_{0\le i<j\le 6}\alpha^{u(q^i+q^j)})^2 \\ &= a_{2,2u} + 2\Big(\sum_{0\le i<j<k\le 6}(\alpha^{u(2q^i+q^j+q^k)}+\alpha^{u(q^i+2q^j+q^k)} \\ &\quad +\alpha^{u(q^i+q^j+2q^k)}) + 3\sum_{0\le i<j<k<l\le 6}(\alpha^{u(q^i+q^j+q^k+q^l)})\Big) \\ &= a_{2,2u} + 2\Big(\sum_{i=0}^{6}\alpha^{uq^i}\sum_{0\le i<j<k\le 6}\alpha^{u(q^i+q^j+q^k)} \\ &\quad -\sum_{0\le i<j<k<l\le 6}\alpha^{u(q^i+q^j+q^k+q^l)}\Big) \\ &= a_{2,2u} + 2a_{1,u}a_{3,u} - 2a_{3,u}^p, \end{aligned}$$

which prove (1) and (2). The rest can be similarly proven. ■

4. Compressed Pairings

The compressed reduced Tate pairing (and its derivatives) $\epsilon_r(P,Q)$ is defined by $Tr(e_r(P,Q)) = Tr(f_{r,P}(D)^{(q^k-1)/r})$ in [28]. This corresponds to the first elementary symmetric function $a_{1,m}$ of the minimal polynomial for any elements α^m of the cyclotomic subgroup $G_{r,q,k}$ in $\mathbb{F}_{q^k}^*$. It is convenient to extend the definition $\epsilon_r(P,Q)$ by considering Lemma 1 when computing pairing values represented more than one element. This is done by using the Newton's Identity.

Some pairing-based cryptographic protocols have been used to take a profit from compressed pairings. The classical example is the BLS short signature scheme that was given in Section 2. We will now give the modified signature scheme for compressed pairings as follows [28]: Let $P \in E(\mathbb{F}_{q^k})$ such that $G_1 =< P >$ of order n.

- **Key Generation :** Pick a random $c \in \mathbb{Z}_n^*$ and compute cP. The secret key is c and the public key ξ is the x coordinate of the point cP.
- **Sign :** Let $H : \{0,1\}^* \rightarrow G_1^*$ be a cryptographic hash function. Given a secret key c and a message $m \in \{0,1\}^*$, compute $S = cH(m) \in E(\mathbb{F}_{q^k})$. The signature σ is the x coordinate of the point $S = cH(m)$, which is an element of $\mathbb{F}_{q^k}$.
- **Verify :** Given a public key ξ, a message m and a signature σ, verify $e(P, \pm S) = e(\pm cP, H(m))$ or $e(P, \pm S) = e(\pm cP, H(m))^{-1}$.

Using the property that any pairing value is unitary, one can simply check whether $Tr(e(P, \pm S)) = Tr(e(\pm cP, H(m)))$ to verify BLS signature scheme.

5. Conclusions

In this paper, we showed explicitly how the final exponentiation is related to the linear recurrence relations, and studied the work done in the literature. Moreover, for embedding degree $k = 14$, we developed several recurrence relations related to the sequences $\{a_{1,m}\}$, $\{a_{2,m}\}$ and $\{a_{3,m}\}$; however, we could not get any polynomial time algorithm to compute the m-th term of them which is left as an open problem.

6. Acknowledgment

I would like to express my sincere gratitude to the anonymous referees for their valuable comments, to Professor Ersan Akyıldız for his diligent guidance. An extended abstract of this paper appears in the Proceedings of 4th International Information Security and Cryptology Conference, pages 145-149, 2010.

References

[1] P. S. L. M. Barreto, S. D. Galbraith, C. Eigeartaigh, and M. Scott. “Efficient pairing computation on supersingular abelian varieties”. *Designs, Codes and Cryptography, 42(3)*, pages 239-271, 2007.

[2] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. “Efficient algorithms for pairing-based cryptosystems”. *Advances in Cryptology - Crypto 2002, LNCS 2442, Springer-Verlag*, pages 354-368, 2002.

[3] P. S. L. M. Barreto, B. Lynn, and M. Scott. “On the selection of pairing-friendly groups”. *Selected Areas in Cryptography - SAC 2003, LNCS 3006*, pages 17-25, 2004.

[4] I.F. Blake, G. Seroussi, and N.P. Smart. “Advances in Elliptic Curve Cryptography”. *London Math. Soc. Lec. Note S., 317, Cambridge*, 2005.

[5] D. Boneh, and M. Franklin. “Identity-based encryption from the Weil pairing”. *SIAM Journal of Computing, 32(3)*, pages 586-615, 2003.

[6] D. Boneh, B. Lynn, and H. Shacham. “Short signatures from the Weil pairing”. *Advances in Cryptology - Asiacrypt 2001, LNCS 2248, Springer-Verlag*, pages 514-532, 2002.

[7] W. Bosma, J. Hutton, and E. Verheul. “Looking beyond XTR”. *Advances in Cryptology - Asiacrypt 2002, LNCS 2501, Springer-Verlag*, pages 46-63, 2002.

[8] A. Brouwer, R. Pellikaan, and E. Verheul. “Doing more with fewer bits”. *Advances in Cryptology - Asiacrypt 1999, LNCS 1716*, pages 321-332, 1999.

[9] C. M. Fiduccia. “An efficient formula for linear recurrences”. *SIAM J. Comput., 14*, pages 106-112, 1985.

[10] S. Galbraith, K. Harrison, and D. Soldera. “Implementing the Tate pairing”. *Algorithmic Number Theory Symposium - ANTS V, LNCS 2369*, pages 324-337, 2002.

[11] K. Giuliani, and G. Gong. “Efficient Key Agreement and Signature Schemes Using Compact Representations in $GF(p^{10})$”. *IEEE International Symposium on Information Theory - ISIT 2004*, pages 13-13, 2004.

[12] K. Giuliani, and G. Gong. “A New Algorithm to Compute Remote Terms in Special Types of Characteristic Sequences”. *Sequences and Their Applications - SETA 2006, LNCS 4086*, pages 237-247, 2006.

[13] G. Gong, and L. Harn. “Public-key cryptosystems based on cubic finite field extensions”. *IEEE Transactions on Information Theory 45, no. 7*, pages 2601-2605, 1999.

[14] G. Gong, L. Harn, and H. Wu. “The GH Public-key Cryptosystems”. *Selected Areas in Cryptography - SAC 2001, LNCS 2259*, pages 284-300, 2001.

[15] G. Gong, A. Hassan, H. Wu, and A. Youssef. “An Efficient Algorithm for Exponentiation in DH Key Exchange and DSA in Cubic Extension Fields”. *Research report at Faculty of Math., University of Waterloo*, 2002.

[16] F. Hess, N. Smart, and F. Vercauteren. “The eta pairing revisited”. *IEEE Transactions on Information Theory, 52(10)*, pages 4595-4602, 2006.

[17] K. Hoffman, and R. Kunze. “Linear Algebra”. *Prentice Hall, New Jersey, USA, 2nd edition*, 1971.

[18] K. Karabina. “Factor-4 and 6 compression of cyclotomic subgroups of $\mathbb{F}^*_{2^{4m}}$ and $\mathbb{F}^*_{3^{6m}}$”. *J. Math. Crypt., 4(1)*, pages 1-42, 2010.

[19] B. B. Kırlar. “Elliptic Curve Pairing-Based Cryptography”, PhD Thesis, 2010.

[20] E. Lee, H. Lee, and C. Park. “Efficient and generalized pairing computation on abelian varieties”. *Cryptology ePrint Archive.*

[21] A. Lenstra, and E. Verheul. “The XTR public key system”. *Advances in Cryptology - Crypto 2000, LNCS 1880*, pages 1-19, 2000.

[22] R. Lidl, and H. Niederreiter. “Finite Fields”. *Cambridge University Press, UK, 2nd edition*, 1997.

[23] B. Lynn. “On The Implementation of Pairing-Based Cryptosystems”, PhD Thesis, 2007.

[24] V. Miller. “The Weil pairing, and its efficient calculation”. *Journal of Cryptology, 17(4)*, pages 235-262, 2004.

[25] A. Miyaji, M. Nakabayashi, and S. Takano. “New explicit conditions of elliptic curve traces for FR-reduction”. *IEICE Trans. Fund. Electron. Comm. Comput. Sci., E84-A(5)*, pages 1234-1243, 2001.

[26] L. Quoos, and S.-F. Mjølsnes. “Public Key Systems Based on Finite Field Extensions of Degree Five”. *Presented at Fq7 conference*, 2003.

[27] M. Shirase, D. Han, Y. Hibin, H. Kim, and T. Takagi. “A more compact representation of XTR cryptosystem”. *IEICE Trans. Fund. Electron. Comm. Comput. Sci., E91-A(10)*, pages 2843-2850, 2008.

[28] M. Scott, and P. Barreto. “Compressed pairings”. *Advances in Cryptology - Crypto 2004, LNCS 3152, Springer-Verlag*, pages 140-156, 2004.

[29] P. Smith, and M. Lennon. “LUC: A new public key system”. Proceedings of the 9th IFIP Symp. - IFIP/Sec 1993, pages 103-117, 1993.

[30] P. Smith, and C. Skinner. “A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms”. *Advances in Cryptology - Asiacrypt 1994, LNCS 917*, pages 357-364, 1994.

[31] S.-M. Yen, and C.-S. Laih. “Fast algorithms for LUC digital signature computation”. *IEE Proc.Comput. Tech. 142(2)*, pages 165-169, 1995.

[32] C. Zhao, F. Zhang, and J. Huang. “A note on the ate pairing”. *Cryptology ePrint Archive.*