ON THE IMPLEMENTATION OF COMPOSITE-ORDER BILINEAR GROUPS IN CRYPTOGRAPHIC PROTOCOLS

BY

SEVERIANO K. SISNEROS

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Adviser:

Professor Richard Blahut

ABSTRACT

In this thesis we present the latest developments in composite-order pairing based cryptography. We first develop the necessary mathematical background. We describe elliptic curves over finite fields, as well as rational functions and divisors on such curves. Bilinear pairings on elliptic curves are explained. We then introduce bilinear groups of composite order and how they are used to create pairing-based cryptosystems. We present two cryptosystems using composite-order groups. We conclude with a discussion of current research.

TABLE OF CONTENTS

1 INTRODUCTION

Elliptic curves were first suggested for use in cryptography independently by Koblitz [1]and Miller [2] over two decades ago and since then there has been an ever increasing amount of research into the development of cryptosystems based on elliptic curves. In particular, researchers have found it possible to use bilinear pairings of the groups formed by the points on an elliptic curve with point addition for the solution of many cryptographic primitives. For example, in 2001 Boneh and Franklin used bilinear pairings to implement the first practical and secure identity based encryption scheme [3], a possibility proposed by Shamir in 1984 [4]. Due to the complexity of bilinear pairings, the elliptic curves used to generate groups for pairing-based protocols must possess certain properties that are not likely to be found in randomly generated elliptic curves [5]. This has led to the definition of pairing-friendly elliptic curves [6]. For the reader interested in the remarkable history and development of elliptic curve cryptography, A. Koblitz, N. Koblitz, and Menezes have a highly readable and interesting exposition [7].

The first applications of pairings focused primarily on pairings of groups with prime order but a powerful new idea has emerged in pairing-based cryptography which uses bilinear groups with composite order rather than prime. The idea was introduced by Boneh et al. [8] for partial homomorphic public key encryption and has since been used in many important applications including non-iterative zero-knowledge proofs, group and ring structures, searching encrypted data, and fully collusion resistant traitor tracing [9], [10], [11], [12], [13], [14], [15]. These works leverage convenient features of composite-order groups that are not shared with prime-order groups. Special consideration must be made when implementing protocols based on bilinear pairings of composite-order groups. The efficient implementation of such schemes is an active area of research.

Our main goal with this thesis is to provide a comprehensive survey of the current research into bilinear groups of composite order and their application to pairing-based cryptography.

Currently the only known groups which admit a bilinear pairing are those derived from the points on elliptic curves. Thus, we first develop the necessary background on elliptic curves, and go over some topics from algebraic geometry such as divisors and rational functions, which we need when defining bilinear pairings.

We then introduce the abstract definition of a bilinear pairing and then show how a pairing can be defined on elliptic curves.

We then show how these concepts can be used in cryptography. The discrete log problem on elliptic curves and the mathematical problems arising from the bilinear pairing are discussed.

Once we have the necessary background, we delve into a study of bilinear groups of composite order. We highlight the useful properties of these groups which are exploited to solve unique problems in cryptography as well as some of the drawbacks of these groups, compared to prime-order bilinear groups. We explain how elliptic curves are used to generate these groups and how to construct elliptic curves admitting the necessary group structure.

We then demonstrate two representative cryptographic protocols using bilinear pairings of composite-order groups.

We conclude with a discussion of the current research.

Much has been written about the implementation of cryptographic protocols based on pairings of groups with prime order, but the same cannot be said for groups of composite order. It is hoped that this text will provide the reader with the necessary background to implement pairing based cryptosystems, in particular those based upon groups of composite order. The reader should also have the necessary background to conduct research in this area.

This text is designed to be somewhat self-contained; however, the reader will need some background in algebra.

2 ELLIPTIC CURVES

The theory of elliptic curves is rich, varied, and vast. Our goal here is not to present a thorough exposition to the theory, building up from the foundations of algebraic geometry. Rather, we prefer a pragmatic approach, by first developing an intuitive “feel” for what an elliptic curve is and then introducing the theory as we need it. In doing so, we will introduce topics from algebraic geometry, such as divisors and rational functions, which will come in handy later when we define bilinear pairings. We’ll save finding particular “pairing-friendly” curves for a later chapter.

For cryptography we are mostly concerned with curves over finite field, however, in this chapter we will define curves over arbitrary (perfect) fields. We do this because later we will want to derive information about curves over finite fields; and to do so we make use of curves over $\mathbb{C}$ and over extensions of the p–adic numbers $\mathbb{Q}_p$.

For the reader who wants a thorough study of elliptic curves, see Silverman’s classic [16].

2.1 THE BASIC DEFINITION

A *plane curve* $\mathcal{X}$ is the set of zeros in the plane F^2 of a bivariate polynomial, *p(x,y)*. We write

$$\mathcal{X} = \{(x, y) \in F^2 : p(x, y) = 0\}.$$

We can define a plane curve to include points appended to but outside of the plane. Such points are called *points at infinity* or *base points*. We will denote a point at infinity by $\mathcal{O}$. Now we can write a plane curve as

$$\mathcal{X} = \{(x, y) \in F^2 : p(x, y) = 0\} \cup \{\mathcal{O}\}.$$

We will only be concerned with plane curves defined by *non-singular polynomials*, known as *non-singular plane curves*. To understand what this means, we define a *singular point* of the bivariate polynomial *p(x,y)* as a point *P* = *(x,y)* such that

$$\frac{\partial p(x,y)}{\partial x} = \frac{\partial p(x,y)}{\partial y} = p(x, y) = 0.$$

The polynomial *p(x,y)* is called a *non-singular polynomial* if it has no singular points in *F* or any finite extension of *F*. A curve $\mathcal{X}$ defined by the zeros of a non-singular polynomial is called a *non-singular (projective) curve*. The curve $\mathcal{X}$ is called a *smooth curve*.

The *genus* of a plane curve is used to describe useful properties of the curve. For a nonsingular curve, the genus is given by $g = \binom{d-1}{2}$.

Definition 2.1.1. An *elliptic curve, E*, over the field *F,* is a plane curve with genus 1 given by the set of zeros of a nonsingular, smooth, bivariate polynomial of the form

$$p(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6$$

together with the point at infinity, $\mathcal{O}$, and where $a_1, a_2, a_3, a_4, a_6 \in F$.

The polynomial in the above definition is said to be in *Weierstrass form*. We see that requiring E to be smooth is essentially requiring that the equations

$$a_1y = 3x^2 + 2a_2x + a_4, \qquad 2y + a_1x + a_3 = 0$$

cannot be satisfied simultaneously by any $(X, Y) \in E(\bar{F})$ (recall that $\bar{F}$ denotes the algebraic closure of F).

Whenever the field characteristic is greater than three, by an appropriate change of variables (specifically $x \to x - \frac{1}{3}a_2$) we can express the elliptic curve E as

$$E: y^2 = x^3 + ax + b.$$

This is known as the *short Weierstrass form* for the elliptic curve. In this case, requiring the curve to be smooth is essentially requiring that the cubic on the right-hand side has no multiple roots. This holds if the *discriminant* of $x^3 + ax + b$, which is $-(4a^3 + 27b^2)$, is nonzero.

2.2 Point addition

We are able to use elliptic curves in cryptography because the points on the curve over F form a group, denoted $E(F)$, under the operation of point addition. We now define this operation.

We denote the operation of point addition by "+". The operation is such that if P and Q are points on the curve E then so is P+Q. As mentioned previously, the points on an elliptic curve form a group under the operation of point addition. The identity of this group is the point at infinity, $\mathcal{O}$, thus $P+\mathcal{O}=P$. The inverse of a point P, is denoted by $-P$so $P+(-P)=\mathcal{O}$.

The operation of point addition is based on the fact that any straight line that intersects an elliptic curve at two points must intersect the curve at three points; a point of tangency is regarded as a double intersection and a “vertical” line through a point and its negative is regarded to also intersect the point at infinity.

We define point addition on the condition that the addition of any three collinear points P, Q, and R of E satisfies

$$P + Q + R = \mathcal{O}.$$

This means that $P + Q = -R$.

Figure 2.1 shows a graphical depiction of this.

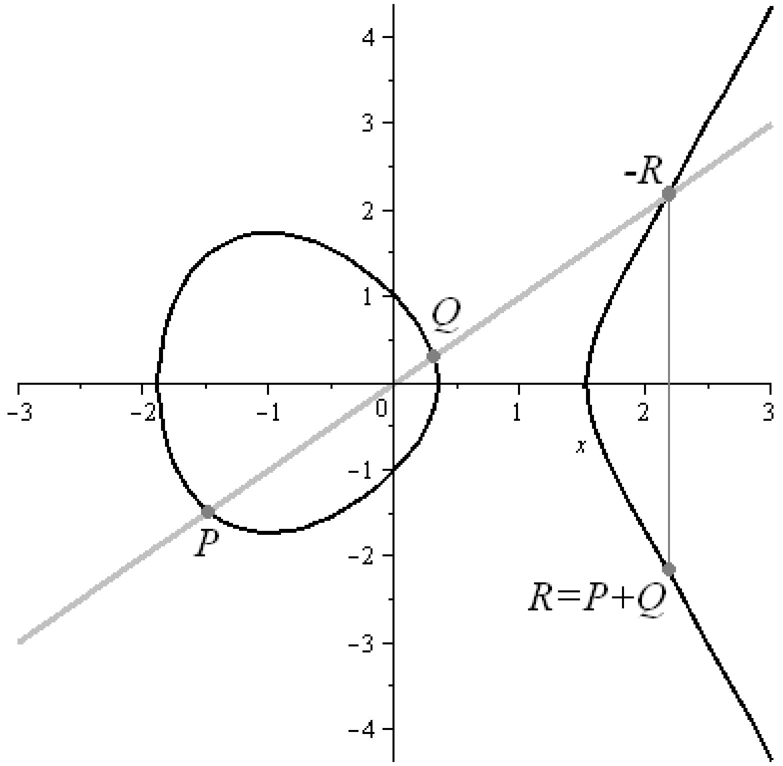

Figure 2.1: Addition of points on an elliptic curve

For the case when the characteristic of the field is neither two nor three, we can describe this process algebraically. Reflecting the point $P = (x, y)$ is done by simply inverting the y coordinate: $-P = (x, -y)$. Addition of two distinct points works as follows. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be distinct points such that $P \neq Q$. The slope of the line through P and Q is given by $\lambda = \frac{y_1 - y_2}{x_1 - x_2}$. Now we can calculate the coordinates of $P + Q = R - (x_3, y_3)$ by reflecting the third point of intersection

$$x_3 = \lambda^2 - x_1 - x_2,$$

$$y_3 = -y_1 + \lambda(x_1 - x_3).$$

Adding $P = (x_1, y_1)$ to itself, known as *point doubling*, is done as follows. If $y_1 = 0$ then $2P = \mathcal{O}$. If $y_1 \neq 0$ then we first need to calculate the slope of the tangent to P, which is given by $\lambda = \frac{3x_1^2+a}{2y_1}$. For the coordinates of $2P = R = (x_3, y_3)$, the reflected third point of intersection is given by

$$x_3 = \lambda^2 - 2x_1,$$

$$y_3 = -y_1 + \lambda(x_1 - x_3).$$

These equations need to be modified for a field of characteristic two or three.

2.3 GROUP ORDER

Consider a curve E defined over a finite field F_q, a field of characteristic q. We would like to know the number of points on the curve. The number of points on the curve E is called the *order* of E and is denoted $\#E(F_q)$. We see that for every $x \in F_q$ there are at most two values of $y \in F_q$ for which (x,y) is a point on the curve. Including the point at infinity we see that there can never be more than $2q + 1$ points on the curve. We note that about half of the field elements are squares; thus, for each fixed $x \in F_q$, half will have two solutions in y and the others will have no solution. Thus we reason that there are roughly $q + 1 + \epsilon$ points on the curve, where ϵ is really small. This is actually a pretty close approximation and is formalized by a famous theorem from Hasse.

Theorem 2.3.1. (Hasse) Let E be an elliptic curve over a finite field F_q. Then the following holds:

$$\left|\#E(F_q) - (q + 1)\right| \leq 2\sqrt{q}.$$

Definition 2.3.2. *The trace of Frobenius of the elliptic curve $E(F_q)$ over the finite field F_q, is the integer*

$$t = q + 1 - \#E(F_q).$$

2.4 TORSION SUBGROUPS

As in all additive groups, the addition of points on the curve induces a scalar multiplication of points. For $n \in \mathbb{Z}$ and a point, $P \in E(F_q)$, we define

$$nP = \begin{cases} P + P + \cdots + P, (n\ times) & if\ n > 0 \\ (-P) + (-P) + \cdots + (-P), (n\ times) & if\ n < 0 \\ \mathcal{O}, & if\ n = 0 \end{cases}$$

The *order* of a point *P* is the smallest positive integer *n* such that $nP = \mathcal{O}$. If the order of a point *P* divides *n* (i.e. if $nP = \mathcal{O}$), then *P* is called an *n-torsion point*. The set of *F*-Rational *n*-torsion points forms a subgroup, the *n-torsion group*, denoted by $\#E(F)[n]$. The following theorem gives us valuable information about the structure of the *n*-torsion group.

Theorem 2.4.1. *Let $E(F_q)$be an elliptic curve over a finite field F_q of prime characteristic p and let $n \in \mathbb{Z}$ be a prime coprime to p. Then the order of $E(F_q)[n]$ is n^2 and $E(F_q)[n] \cong \mathbb{Z}\backslash n\mathbb{Z} \oplus \mathbb{Z}\backslash n\mathbb{Z}$.*

For a special class of curves the *n*-torsion group is trivial when *n* is a power of the characteristic of the field over which the curve is defined. Let *E(F_q)* be an elliptic curve over the finite field F_q, where $q = p^m$ for prime *p* and $m \in \mathbb{Z}$. If $p|t$, where *t* is the trace of Frobenius of *E*, then *E* is called *supersingular.*

2.5 RATIONAL FUNCTIONS ON ELLIPTIC CURVES

Suppose $E(F)\colon y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ is an elliptic curve over an arbitrary field *F*. Observe that every polynomial $p(x, y) \in E(\bar{F})$ (where $E(\bar{F})$ denotes the set of all points on the curve *E* defined on the algebraic closure of *F*) can be written in the form $a(x) + b(x)y$, by rearranging terms and replacing any occurrence of y^2 by $x^3 + a_2x^2 + a_6 - a_1xy - a_3y$. This form can be used to define the degree of *f*, which is needed for determining the orders of zeros and poles of *f* as well as the behavior of *f* at $\mathcal{O}$.

Definition 2.5.1. *Let $f(x, y) = a(x) + b(x)y \neq 0$ be a polynomial function in $E(\bar{F})$. Then the degree of f is defined to be*

$$\deg(f) = \max\{\deg_x(a), 3 + 2\deg_x(b)\},$$

where $\deg_x$ *denotes the usual degree of a polynomial in x, with* $\deg_x(0) = \mathcal{O}$.

We define the behavior of rational functions on *E* in the point at infinity as follows.

Definition 2.5.2. *Let* $f = \frac{g}{h} \in E(\bar{F})$*. If* $\deg(g) < \deg(h)$ *then we define* $f(\mathcal{O}) = 0$*. If* $\deg(g) > \deg(h)$ *we say f is not defined at* $\mathcal{O}$*. If* $\deg(g) = \deg(h)$*, then* $f(\mathcal{O})$ *is defined to be the ratio of the leading coefficients of g and h.*

2.6 Divisors

Here we discuss the notion of divisors from algebraic geometry.

Let *E* be an curve over a field *F*. Consider the set of all points on the curve defined over the algebraic closure of *F*, $E(\bar{F})$.

Definition 2.6.1. *A divisor on E is a formal sum*

$$D = \sum_{P \in E(\bar{F})} n_P(P).$$

where $n_p \in \mathbb{Z}$ *and all but finitely many* n_p *are zero.*

If all n_p= 0, the divisor is denoted 0. The set of all divisors on *E* is denoted $Div_{\bar{F}}(E)$ and has a natural group structure under the operation of addition.

Definition 2.6.2. *The support of a divisor D is the set of all points P such that* $n_p \neq 0$.

Definition 2.6.3. *The degree of a divisor D is*

$$\deg(D) = \sum_P (n_p).$$

For $\sigma \in Gal(\bar{F}/F)$ we define $D^\sigma = \sum_P n_P(\sigma(P))$, where $Gal(\bar{F}/F)$ denotes the absolute Galois group over *F*. We say that a divisor is *defined over F* if $D = D^\sigma \forall \sigma \in Gal(\bar{F}/F)$.

If *f* is a non-zero function on *E*, then $ord_P(f)$ counts the multiplicity of *f* at *P*. Note that $ord_P(f)$ is positive at zeros of *f* and negative at poles of *f*. The *divisor of f*, written (f), is the divisor $\sum_{P \in E(\bar{F})} ord_p(f)(P)$. If follows that $(fg) = (f) + (f)$ and $\left(\frac{f}{g}\right) = (f) - (g)$.

Definition 2.6.4. *A principal divisor on E is a divisor which is equal to* (f) *for some f; i.e., a principal divisor is a divisor that corresponds to poles and zeros, counted with multiplicity, of a rational function on the curve E.*

Theorem 2.6.5. *A divisor of the elliptic curve E(F) is a principal divisor of E(F) iff the following conditions are satisfied:*

1. $\sum_{P \in E(\bar{F})} n_p = 0$,
2. *as a sum of points* $\sum_{P \in E(\bar{F})} [n_P]P = \mathcal{O}$.

3 PAIRINGS

Here we begin our study of the central figure in pairing based cryptography, the bilinear map. First we look at the problems that motivate the use of pairings in cryptography. We will introduce the pairings abstractly and then introduce the original Weil pairing, the more efficient Tate pairings, and the optimized Omega pairing for composite-order groups. The bilinear map is generally the most computationally expensive action in pairing based cryptosystems hence there has been much research into optimizations of the map. We look at some of the latest optimized implementations with a focus on those over composite-order groups.

3.1 THE ABSTRACT DEFINITION

There are two forms of maps, *e*, commonly used in cryptography [17]. The first are of the form

$$e: G_1 \times G_1 \rightarrow G_T,$$

where G_1 and G_T are cyclic groups of order *l*. Pairings of this type are called *Type 1* or *symmetric* pairings. The second form is

$$e: G_1 \times G_2 \rightarrow G_T,$$

where G_1, G_2, and G_T are groups of order *l*. Pairings of this type are called *asymmetric.* Note that the first form can be considered a special case of the second, where $G_1 = G_2$. Also note that when $G_1 \neq G_2$, but there exists a computable homomorphism between the two groups in both directions, the situation can be reinterpreted as a *Type 1* pairing, and so this situation is not considered separately. We call the situation where $G_1 \neq G_2$ and there exists no computable homomorphism between the two groups of a *Type 3* pairing.

The pairings used in cryptosystems also satisfy the following properties:

1. Bilinearity: $\forall P, P' \in G_1, \forall Q, Q' \in G_2$, we have $e(P + P', Q) = e(P, Q)e(P', Q)$ and $e(P, Q + Q') = e(P, Q)e(P, Q')$.
2. Non-degeneracy:
    a. $\forall P \in G_1, with\ P \neq 0, \exists Q \in G_2: e(P, Q) \neq 1.$
    b. $\forall Q \in G_2, with\ Q \neq 0, \exists P \in G_1: e(P, Q) \neq 1.$
3. For practical purposes, *e* has to be efficiently computable.

Following are some consequences of bilinearity, for a bilinear pairing *e*:

1. $e(P, 0) = e(0, Q) = 1$.
2. $e(-P, Q) = e(P, Q)^{-1} = e(P, -Q)$.
3. $e([j]P, Q) = e(P, Q)^j = e(P, [j]Q) \quad \forall j \in \mathbb{Z}$.

3.2 THE TATE PAIRING

The Tate pairing is the most important pairing in elliptic curve cryptography [18]; it was introduced by Tate a general pairing on abelian varieties over local fields. Frey and Ruck [19]considered the pairing over finite fields, thus introducing the Tate pairing to the cryptographic community.

First we consider the Tate pairing over an arbitrary field and then move to the case of a finite field.

3.2.1 DEFINING THE TATE PAIRING

Let E be an elliptic curve over a field F_0 and let $n \in \mathbb{Z}^+$be coprime to the characteristic of the field F_0. The set of the nth-roots of unity is defined to be $\mu_n = \{u \in \bar{F}_0^*: u^n = 1\}$. Define the field $F = F_0(\mu_n)$ to be the extension of F_0 generated by the nth-roots of unity. Define

$$E(F)[n] = \{P \in E(F): [n]P = \mathcal{O}\},$$

and

$$nE(F) = \{[n]p: P \in E(F)\}.$$

Then $E(F)[n]$ is a group of exponent n. Also, $nE(F)$ is a subgroup of $E(F)$ and the quotient group $E(F)/nE(F)$ is a group of exponent n.

Define

$$(F^*)^n = \{u^n: u \in F^*\}.$$

Then $(F^*)^n$ is a subgroup of F^* and the quotient group $F^*/(F^*)^n$ is a group of exponent n. The groups $F^*/(F^*)^n$ and μ_n are isomorphic.

Note that while $\#(E(F)[n] = \#E(F)/nE(F)$ it is not necessarily true that the points of $E(F)[n]$ may be used as representatives for the classes of the quotient group $E(F)/nE(F)$.

Now we are ready to define the Tate pairing. Let $P \in E(F)[n]$ and let $Q \in E(F)$. We think of Q as representing an equivalent class in $E(F)/nE(F)$. Since $[n]P = \mathcal{O}$, there is a function f such that $(f) = n(P) - n(\mathcal{O})$. Let D be any degree zero divisor equivalent to $(Q) - (\mathcal{O})$ such that D is defined over F and the support of D is disjoint from the support of (f). Since f and D are defined over F, the value $f(D) \in F$. Since the supports of (f) and D are disjoint, $f(D) \neq 0$ and so $f(D) \in F^*$.

Definition 3.2.1. *The Tate pairing of P and Q is defined to be*

$$\langle P, Q \rangle_n = f(D)$$

interpreted as an element of $F^*/(F^*)^n$.

Note that the values of the Tate pairing are equivalence classes. Thus, the symbol = is used under this equivalence relation.

Now let us consider the Tate pairing over finite fields.

Suppose that $E(F_q)$ is an elliptic curve over a finite field of characteristic p. Let $n \in \mathbb{Z}$ be coprime to q which divides $\#E(F_q)$. The field $K = F_q(\mu_n)$ is some finite extension F_{q^k}.

Definition 3.2.2. *The number k is called the embedding degree and is defined as the smallest number such that n divides (q*^{k} *-1).*

Since E is defined over F_q, if P and Q are defined over a proper subfield of F_{q^k}, then the Tate pairing $\langle P, Q \rangle_n$ is also defined over the same proper subfield. Suppose n is a prime; since F_{q^k} is the smallest field containing both μ_n and F_q it follows that, for every intermediate field$F_q \subseteq L \subset F_{q^k}$ with $L \neq F_{q^k}$, we have $L \subseteq \left(F_{q^k}^*\right)^n$. These observations lead us to the following result, which tells us when the values of the pairing are trivial.

Lemma 3.2.3. *Let E be an elliptic curve over* F_q*and let n be a prime. Suppose the embedding degree for E and n is k > 1. Let L be a proper subfield of* F_{q^k} *which contains* F_q*. If* $P, Q \in E(L)$*, then* $\langle P, Q \rangle_n \in \left(F_{q^k}^*\right)^n$*.*

From the above definition of the Tate pairing, in a finite field F_q with embedding degree k, the value of the Tate pairing is an equivalence class in $F_{q^k}^*/\left(F_{q^k}^*\right)^n$. For practicality we would like a unique representative of this class. To achieve this we raise the value of the Tate pairing to the power $(q^k - 1)/n$. This leaves exactly the nth roots of unity in F_{q^k}. Thus, for finite groups, we have the bilinear pairing used by cryptographers

$$e(P,Q) = \langle P,Q\rangle_n^{\frac{q^k-1}{n}}$$

which maps into the group $\mu_n \subset F_{q^k}^*$ rather than the group $F_{q^k}^*/\left(F_{q^k}^*\right)^n$.

3.2.2 Computing the Tate pairing using Miller's algorithm

Miller [20] gave an algorithm for computing the Weil pairing in polynomial time and this approach can also be applied to compute the Tate pairing. Miller uses the double-and-add method to construct a function *f* such that $(f) = n(P) - n(\mathcal{O})$.

We define function f_i such that $(f_i) = i(P) - ([i]P) - (i-1)(\mathcal{O})$. Since P is an n-torsion point, it follows that

$$(f_n) = n(P) - (nP) - (n-1)(\mathcal{O}) = n(P) - n(\mathcal{O}) = (f).$$

Note that $(f_1) = (P) - (P) - 0(\mathcal{O}) = 0$, which means that $f_1 = 1$.

Lemma 3.2.4. *Let* $P \in E(F_{q^k})[n]$, *and let i and j be positive integers. Suppose that l is function such that* $l(x,y) = 0$ *is the equation of the line between the points iP and jP and suppose v is a function such that* $v(x,y) = 0$ *is the equation of the vertical line through the point* $(i+j)P$. *Then*

$$f_{i+j} = f_i \cdot f_j \cdot \frac{l}{v}.$$

Proof. We have

$$\left(\frac{l}{v}\right) = (iP) + (jP) - ([i+j]P) - (\mathcal{O}).$$

And so

$$\begin{aligned} f_i \cdot f_j \cdot \frac{l}{v} &= i(P) - ([i]P) - (i-1)(\mathcal{O}) + j(P) - ([j]P) - (j-1)(\mathcal{O}) + \left(\frac{l}{v}\right) \\ &= (i+j)(P) - ([i+j]P) - (i+j-1)(\mathcal{O}). \end{aligned}$$

Q.E.D.

Miller's algorithm uses an addition chain for *nP* to compute f_n. We are interested in the value of $f_n(D)$, thus we evaluate all intermediate quantities at the divisor $D = (Q+S) - (S)$.

Algorithm 3.2.5. Miller's Algorithm

```
INPUT:  $P, Q \in E(F)$where $P$ has order $n$.
OUTPUT: $\langle P, Q \rangle_n$.
Choose a suitable point $S \in E(F)$.
$Q' \leftarrow Q + S$.
$T \leftarrow P$.
$m \leftarrow \lfloor \log_2 n \rfloor - 1, f \leftarrow 1$.
While $m \geq 0$ do:
    Calculate lines $l$ and $v$ for doubling $T$
    $T \leftarrow 2T$.
    $f \leftarrow \frac{f^2\left(l(Q')v(S)\right)}{v(Q')l(S)}$.
    If the $m$th bit of $n$ is one, then:
        Calculate lines $l$ and $v$ for addition of $T$ and $P$.
        $T \leftarrow T + P$.
        $f \leftarrow \frac{f\left(l(Q')v(S)\right)}{v(Q')l(S)}$.
    $m \leftarrow m - 1$
Return $f$.
```

4 CRYPTOSYSTEMS BASED ON THE DISCRETE LOG PROBLEM

In this chapter we focus on the discrete log problem and how it has been used to develop cryptosystems. We start out working in $F_q^{\times}$, then move to elliptic curve groups. We also look at the most effective attack on these problems in both groups. The cryptosystems built from these problems form the foundation from which pairing-based cryptography grew. It is thus useful to examine how these systems are defined.

4.1 THE DISCRETE LOG PROBLEM

We use multiplicative notation simply because the discrete log problem was first defined for multiplicative groups.

Definition 4.1.1. *The discrete log problem (DLP)*

Given (G, g, β) find the least positive integer a such that

$$g^{a} = \beta.$$

We denote this integer a as $log_g(\beta)$; it is called the *discrete logarithm* of β.

This problem is believed to be computationally infeasible. Whenever we compute the discrete log problem with respect to a base $g \in G$, we write DLP_g. When we compute the discrete log problem for any $g \in G$, we write DLP_G.

Definition 4.1.2. *Computational Diffie-Helman (CDH)*

Given an abelian group G, an element $g \in G$ with order n, and two elements $\alpha, \beta \in \langle g \rangle$, find $\delta \in \langle g \rangle$ such that $\log_g \delta \equiv \log_g \beta \times \log_{\mathrm{g}} \alpha \pmod{n}$

(equivalently, given g^{α}, g^{β} find $g^{\alpha\beta}$).

Definition 4.1.3 *Decisional Diffie-Hellman (DDH)*

Given an abelian group G, an element $g \in G$ with order n, and three elements $\alpha, \beta, \gamma \in \langle g \rangle$, is it true that $\log_g \delta \equiv \log_g \beta \times \log_{\mathrm{g}} \alpha \pmod{n}$*?*

(Equivalently, given g^{α}, g^{β}, and g^{γ}, determine if $d \equiv bc \pmod{n}$.)

Note that if we can solve CDH, the DDH follows, and we denote this by $CDH_G \to DDH_G$. Specifically, if there exists an algorithm$\mathcal{A}$ that solves CDH_G in polynomial time, then it is possible to construct an algorithm utilizing $\mathcal{A}$ that solves DDH_G in polynomial time. We say that CDH is a *stronger* problem than DDH.

We also see that $DLP_G \to CDH_G$ holds. The converse is not known to be true, no one knows if it is possible to obtain g^{ab} from g^a and g^b without first determining a or b.

Diffie and Hellman [21] first used the these problems decades ago to demonstrate the security of their key agreement protocol. These problems hold for any cyclic group, e.g., the finite multiplicative group $\mathbb{Z}_p^*$ where p is a prime, or the group formed by the elliptic curve *E(F$_q$)* over the field F_q with prime characteristic under the operation of point addition.

4.2 THE ELGAMAL CRYPTOSYSTEM IN $\mathbb{Z} \times p\mathbb{Z}^*$

The ElGamal cryptosystem was the first and most well-known of the cryptographic protocols based on the discrete log problem. The protocol is as follows [22].

Let p be a prime such that the DLP in ($\mathbb{Z}/p\mathbb{Z}^*$,×) is infeasible, and let $\alpha \in \mathbb{Z}/p\mathbb{Z}^*$ be a primitive element. Let $= \mathbb{Z}/p\mathbb{Z}^*$, $\mathcal{C} = \mathbb{Z}/p\mathbb{Z}^* \times \mathbb{Z}/p\mathbb{Z}^*$, and define

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a (mod\ p)\}.$$

The values $p, \alpha, and\ \beta$ are the public key, and a is the private key.

For $K = (p, \alpha, a, \beta)$, and for a secret random number $k \in \mathbb{Z}/(p-1)\mathbb{Z}$, define the ciphertext

$$\mathcal{C}(x, k) = (y_1, y_2),$$

where

$$y_1 = \alpha^k (mod\ p)\ and,$$

$$y_2 = x\beta^k (mod\ p).$$

For $y_1, y_2 \in \mathbb{Z}/p\mathbb{Z}^*$, to decrypt define

$$M_k(y_1, y_2) = y_2 (y_1^{\alpha})^{-1} (mod\ p).$$

4.3 THE DISCRETE LOG PROBLEM ON ELLIPTIC CURVES

The discrete log problem holds for any abelian group, including the group of points on an elliptic curve. We use additive notation.

Definition 4.3.1. *Elliptic curve discrete log problem (ECDLP)*

Let E be an elliptic curve. For $P, Q \in E$, find $x \in \mathbb{Z}$ such that $P = xQ$if such x exists.

5 BILINEAR GROUPS OF COMPOSITE ORDER

Bilinear groups of composite order are being used to solve many problems in cryptography. They were first introduced by Boneh, Goh, and Nissim [8] in 2005 to solve problems for private information retrieval, online voting, and universally verifiable computation. They have since found a wealth of applications throughout cryptography. These applications leverage properties of bilinear groups of composite order not shared with their prime-order counterparts. Most notably, composite-order groups contain orthogonal subgroups of coprime-order. Up to the notion of isomorphism, a composite-order group has the structure of the direct product of prime-order subgroups. Thus every group element can be decomposed as a product of elements from orthogonal subgroups; when the group order is hard to factor, this decomposition is also hard to compute. Orthogonality means the subgroups can act as separate entities, allowing the system designer to utilize the subgroups independently without the worry of any crossover interactions. Additionally, Freeman [23] and Seo and Cheon [24] have explicitly defined properties of bilinear composite-order groups not known to be shared with the prime-order variants (although there is current research into how to construct prime-order groups with the properties enjoyed by composite-order groups, we discuss this in greater detail later).

The security of most of the protocols using bilinear composite-order groups is based upon the *subgroup decision assumption* defined by Boneh et al. in [8]. Essentially, the assumption is, given any element $g \in G$ where G is a bilinear composite-order group of order $n = p_1 p_2 \dots p_m$, it is not possible to determine if g has order $p_i, i \in [1, \dots, m]$. This assumption implies it is infeasible to factor n .The security of protocols being based on the infeasibility of factoring the order of the group leads to some disadvantages which will be discussed later.

5.1 THE BILINEAR PAIRING ON COMPOSITE-ORDER GROUPS

Let G be a bilinear group of composite with order $N = p_1 p_2 \dots p_m$, where $p_1, \dots, p_m$ are distinct primes. G admits a bilinear pairing,

$$e: G \times G \to G_T$$

where G_T also has order N and e satisfies the conditions for a bilinear pairing defined in Chapter 3.

Note that for each p_i, G has a subgroup with order p_i, denoted G_{p_i}. We denote the generators of each G_i as $g_i \in G_i \ for \ i = 1,2, \dots, m$. Thus, each element $g \in G$ can be represented as $g = g_1^{a_1} g_2^{a_2} \dots g_m^{a_m}, \ a_i \in \mathbb{Z}/N\mathbb{Z}$. We say $g_i^{a_i}$ is the G_{p_i} component of g. When $a_i \equiv 0 \ (mod \ N)$ we say g has no G_{p_i} component. The subgroups are orthogonal under e, meaning that if $g \in G_{p_i}$ and $h \in G_{p_j}, i \neq j$, then $e(g, h) = 1$, where 1 is the identity element in G_T.

5.2 The subgroup decision problem

The problem of determining whether an element $g \in G$ belongs to a proper subgroup G_i has been used as a security assumption for cryptosystems since before Boneh, Goh, and Nissim defined their *subgroup decision problem*. Gjosteen presents a survey of such problems in [25]. Here we'll present a general definition of the subgroup decision problem from Freeman's work in [23] and later we will see how this is used in the development of cryptographic protocols.

Definition 5.2.1. A *bilinear group generator* is an algorithm $\mathcal{G}$ that takes as input a security parameter λ and outputs a description of five abelian groups G, G_1, H, H_1, G_t, with $G_1 \subset G$ and $H_1 \subset H$. We assume that this description permits efficient (polynomial-time in λ) group operations and random sampling in each group. The algorithm also outputs and efficiently computable map

$$e: G \times H \to G_t,$$

where e is:

- Bilinear: $e(g_1, g_2, h_1, h_2) = e(g_1, h_1)e(g_1, h_2)e(g_2, h_1)e(g_2, h_2)$, for all$g_1, g_2 \in G, \ h_1, h_2 \in H$; and
- Nondegenerate: for any $g \in G$, if $e(g, h) = 1$ for all $h \in H$, then $g = 1$ (and vice-versa with H).

Freeman's generalized subgroup decision problem says that it is impossible to distinguish an element from G_1 from a randomly chosen element from G, and vice-versa for H. This statement is made precise in the following definition. (The notation $x \overset{R}{\leftarrow} X$ means x is chosen at random from X.)

Definition 5.5.2. Let $\mathcal{G}$ be a bilinear group generator. We define the following distribution:

$$\mathbb{G} = (G, G_1, H, H_1, G_t, e) \overset{R}{\leftarrow} \mathcal{G}(\lambda), T_0 \overset{R}{\leftarrow} G, T_1 \overset{R}{\leftarrow} G_1.$$

We define the *advantage* of an algorithm $\mathcal{A}$ in solving the *subgroup decision problem on the left* to be

$$SDP_L - Adv[\mathcal{A}, \mathcal{G}] = |\Pr[\mathcal{A}(\mathbb{G}, T_0) = 1] - \Pr[\mathcal{A}(\mathbb{G}, T_1) = 1]|.$$

We say that $\mathcal{G}$ *satisfies the subgroup decision assumption on the left* if $SDP_L - Adv[\mathcal{A}, \mathcal{G}](\lambda)$ is a negligible function of λ for any polynomial-time algorithm $\mathcal{A}$.

We define the *subgroup decision problem on the right* analogously, with $T_0 \overset{R}{\leftarrow} H, T_1 \overset{R}{\leftarrow} H_1$.

We say $\mathcal{G}$ *satisfies the subgroup decision assumption* if it satisfies both the left and the right assumptions.

5.3 Paring-friendly curves

When we demonstrated the ElGamal encryption scheme earlier, we did not care what elliptic curve we used: any randomly generated elliptic curve would do. This is not the case for pairing-based cryptosystems. The elliptic curves used in pairing-based systems are required to have certain properties that randomly generated curves are unlikely to have. Fortunately there has been research into the elliptic curve requirements for pairing-based cryptosystems and recently Freeman et al. [26] published a comprehensive survey covering many of the known techniques for constructing pairing-friendly curves as well as recommendations on when and how to use them; however, their survey was lacking in methods for constructing curves of composite order.

Early cryptosystems based on composite-order bilinear pairings focused on using supersingular curves; we will see later that under some circumstances this is insufficeint. More recently Boneh et al. [27] presented a method for finding composite-order ordinary elliptic curves of prescribed embedding degree.

To understand these methods of curve construction we must first determine what a “pairing-friendly” elliptic curve is. We will follow Freeman et al.’s approach to this subject in [6]. Given an input k, our goal is to produce a “suitable” composite integer N and an elliptic curve over the finite field F_q such that the order $|E(F_q)|$is a multiple of N and the embedding degree of E with respect to N is k. What do we mean by “suitable”? In order for a pairing-based cryptosystem to be secure, the discrete log problem on the elliptic curve E and the discrete log problem on the extension field F_{q^k} must be infeasible. The Pollard-Rho algorithm [28], [29] for computing the discrete log problem on $E(F_q)$ has a running time $O(\sqrt{N})$. In the extension field F_{q^K} the index calculus attack [30] has subexponential running time. Thus the size of the extension field must be significantly larger than *N.* These sizes are related by two parameters: the embedding degree *k,* which is the degree *k* of the extension field the pairing maps into, and the ratio of the sizes of the base field F_q and *N* given by $\rho = \frac{\log q}{\log N}$. ρ-values have been shown to be the most desirable in order to speed up arithmetic on the elliptic curve. Freeman et al. [26] presented the following definition for a "paring-friendly" curve of composite order.

Definition 5.3.1 Suppose *E* is an elliptic curve defined over a finite field F_q. We say that *E* is *paring-friendly* if the following two conditions hold:

1. there is a composite $r > \sqrt{q}$ dividing $\#E(F_q)$, and
2. the embedding degree of *E* with respect to *r* is less than $\log_2(r)/8$.

The bound in 1 comes from the findings of Luca and Shparlinski [31]. Essentially, they found that curves with small embedding degree with respect to *r* are common if $r > \sqrt{q}$ and are rare if $r < \sqrt{q}$. The bound in 2 is based on the rationale presented in [26], that embedding degrees of practical interest depend on the desired security level of the application, for which *r* has been shown to be a clear measure [32].

Elliptic curves satisfying this definition will be guaranteed to have a large enough embedding degree so that the discrete log problem on $E\left(F_{q^k}^*\right)$ is computationally infeasible and yet is small enough so that the bilinear pairing is easy to compute.

Menezes et al. [33] demonstrated that supersingular curves over prime fields F_p with $p > 5$ have embedding degree 2 and embedding degree of at most 6 in any case. Moreover, supersingular curves may be insufficient for some cryptosystems. In order to achieve high levels of security, we want to be able to vary the embedding degree; thus we must be able to construct pairing-friendly curves.

There has been a lot of research into this problem and several methods have been developed (although most methods are for curves with prime order). At a high level all these methods share a common structure [26]:

1. Fix k, and compute integers t, N, q such that there is an elliptic curve E over F_q that has trace t, a subgroup of order N, and embedding degree k.
2. Use the *complex multiplication* method to find the equation of the curve E over F_q.

We desire to construct elliptic curves which admit groups of composite order, thus the following conditions must hold:

1. q is a prime or prime power.
2. N is a composite number.
3. t is relatively prime to q.
4. N divides $q + 1 - t$.
5. N divides $q^k - 1$, and k is the smallest integer satisfying this condition.
6. $4q - t^2 = Dy^2$ for some sufficiently small positive integer D and some integer y.

Condition (1) ensures that E is defined on a finite field with q elements. Condition (2) together with condition (4) ensures we have a group of composite order N. Condition (6) along with (3) ensure the order $\#E(F_q) = q - t + 1$ from Theorem 4.1 in [34]. Recalling our definition of embedding degree, condition (5) tells us that k is the embedding degree with respect to r.

The methods we will use to construct elliptic curves will make extensive use of the theory of cyclotomic polynomials and cyclotomic fields. We will introduce only what we need to know and the ambitious reader is referred to Lidl and Niederreiter's book [35].

Denote by ζ_n the complex number $e^{\frac{2\pi i}{n}}$; thus, the n roots of the polynomial $x^n - 1 \in \mathbb{C}$ are the n distinct powers of ζ_n. Moreover, they form a cyclic subgroup of order n of the multiplicative group of $\mathbb{C}$, denoted by μ_n.

A *primitive* n-th root of 1 is a generator of μ_n. Thus ζ^n is a primitive, μ_n being a cyclic group implies that ζ_n^m is a primitive n-th root of 1 iff m is relatively prime to n. Specifically, there are $\phi(n)$ primitive roots of 1, where ϕ is Euler's totient function. Thus we define

$$\Phi_n(x) \coloneqq \prod_{\zeta primitive\ root\ of\ 1} (x - \zeta) = \prod_{1 \le m \le n, gcd(m.n)=1} (x - \zeta_n^m)$$

and the degree of the polynomial is $\phi(n)$.

$\Phi_n(x)$ is a monic polynomial and has rational (integer) coefficients and is *irreducible* over $\mathbb{Q}$. $\Phi_n(x)$ is called the *n-th cyclotomic polynomial.*

Given this definition, for all positive integers n,

$$x^n - 1 = \prod_{1 \leq d | n} \Phi_n(x).$$

We are now ready to dive into actual algorithms for constructing elliptic curves. The methods we will use were presented by Boneh, Rubin, and Silverberg in [27] and are the most popular method found in the literature.

5.3.1 BRS METHOD FOR CONSTRUCTING COMPOSITE-ORDER ELLIPTIC CURVES

First we see how to construct a supersingular curve with composite-order group [8].

Constructing supersingular curve with embedding degree 2 and composite order N

Step 1: Choose a square-free integer $N > 3$ that is not divisible by 3.
Step 2: Find the smallest positive integer w such that $q = 3wN - 1$ is a prime number.
Step 3: The elliptic curve $y^2 = x^3 + 1$ over F_q has $q + 1 = 3wN$ points over F_q and embedding degree 2 with respect to N.

Ordinary composite-order groups with embedding degree 1 [27]

Input: a positive integer N
Output: a prime q;
an elliptic curve E over F_q such that $E[n] \subseteq E(F_q)$.

Step 1: Choose a positive integer D suitable for the CM method.

Step 2:

$$\text{Let } q = \begin{cases} 1 + DN^2 & if\ D \equiv 0{,}4\ (mod\ 6), \\ 1 + 4DN^2 & if\ D \equiv 1{,}3\ (mod 6)\ , \\ (1 - N)^2 & if\ D \equiv 5\ (mod\ 6), \\ (1 - 2N)^2 + DN^2 & if\ D \equiv 2\ (mod\ 6). \end{cases}$$

Step 3: If q is prime, use the CM method to obtain an elliptic curve over F_q that has $q - 1 = DN^2$ points when $D \equiv 0{,}4\ (mod\ 6), q - 1 = 4DN^2$points when $D \equiv 1{,}3\ (mod\ 6), q - 1 + 2N = (D + 1)N^2$ points when $D \equiv 5\ (mod\ 6)$, and

$q - 1 + 4N = (D + 4)N^2$ points when $D \equiv 2\ (mod 6)$. If q is not prime, start over with a new D.

Notes:

- Since $N|(q - 1)$, the embedding degree is 1.
- The CM method produces a curve E such that $E[N] \subseteq E(F_q)$. Thus the pairing is computed entirely in the ground field, which is the optimal efficiency case.
- Knowledge of N's factors was not used and thus no information about N's factorization was leaked.
- When N and D are odd, then $1 + DN^2$ is even. When $D \equiv 2\ (mod\ 3)$ and $3 \nmid N$, then $1 + DN^2$ is divisible by 3, so it is not prime (unless $D = 2\ and\ N = \pm 1$). Hence the need to adjust q above.

Ordinary composite oreder groups (version 2)

Input: a positive integer k such that either $4|k$ or k has a prime divisor that is congruent to 3 modulo 4,
distinct primes $p_1, \dots p_r$ congruent to 1 (mod k), and
positive integers $\alpha_1, \dots \alpha_r$.
Let $N = \prod_{i=1}^{r} p_i^{\alpha_i}$.

Output: a prime q;
an elliptic curve E over F_q of embedding degree k with respect to N

Step 1: Choose an integer X with order k in$(\mathbb{Z}/p_i^{\alpha_i}\mathbb{Z})^* \ \forall i$.

Step 2: Choose a positive square-free divisor D of k such that if k is a multiple of 4 then D divides $k/4$, if k is not a multiple of 4 then $D \equiv 3\ (mod\ 4)$.

Step 3: With $\left(-\frac{D}{a}\right)$ denoting the Jacobi symbol, let

$$s = \begin{cases} \sum_{a=1}^{2D-1} \left(-\frac{D}{a}\right) X^{\frac{ak}{D}} \quad (mod\ N) \quad if\ D \equiv 3 \quad (mod\ 4), \\ \frac{1}{2} \sum_{a=1}^{4D-1} \left(-\frac{D}{a}\right) X^{\frac{ak}{4D}} \quad (mod\ N) \qquad otherwise. \end{cases}$$

Step 4: Take an integer Y congruent to $\pm(X-1)s^{-1}$ $(mod\ N)$.

Step 5: Let $q = \frac{(X+1)^2 + DY^2}{4} \in \mathbb{Q}$.

Step 6: If q is a prime number, use CM method to obtain an elliptic curve E over F_q with trace $t = X + 1$, so

$$\left|E(F_q)\right| = q + 1 - t = q - X.$$

If q is not prime, start again with difference X and/or Y.

5.3.2 CHOOSING WHICH CURVES TO IMPLEMENT

Performance and security requirements vary amongst protocols. We provide recommendations for designers to aid with the selection of curves for a given protocol. Because we are using composite-order groups, the factorization of the order $\#E(F_q)$ must be infeasible. There exist sub-exponential time factorization techniques (the Number Field Sieve) but only exponential-time elliptic curve discrete log algorithms, thus the size of the elliptic group will depend on the required security level. To be secure against algorithms like the Number Field Sieve, we should choose parameters for curve construction such that $\#E(F_q) \cong q^k$ [26].

In regards to security, Koblitz [36] showed that composite-order groups used in a pairing-based cryptosystem with embedding degree $k > 2$ are vulnerable. Specifically, he proved that an attacker who observes two independent implementations of a protocol (with same N and $k > 2$ but different E and q) has a probability of at least $1 - \phi(k)^{1-r} \geq 1 - 2^{1-r}$of factoring N, where $r \geq 2$ is the number of distinct prime factors of N and $\phi$is Euler's totient function as described earlier. Thus, for protocols in which it is necessary to protect the factorization of N, it is best to use embedding degree 1 or 2. This would be severely limiting if we were using prime-order groups however, because when using composite-order groups this is not a very restrictive. Indeed, we already have shown that we want $\#E(F_q) = N \cong q^k$, so we would ideally choose embedding degree 1 or 2 anyway; this result only reinforces our decision.

5.3.3 A NOTE ABOUT FAMILIES OF PAIRING-FRIENDLY CURVES

It is shown in [6] that many of the curves constructed using this general framework can be classified into families, in which the parameters N, t, q can be represented as polynomials. This allows us to specify curves with certain bit sizes for applications. It also provides a framework for discovering previously unknown pairing-friendly curves. There has been extensive research to classify families of elliptic curves. Some of the more popular families were found by Miyaji, Nakabayashi, and Takano [37]; Barreto, Lynn, and Scott [38]; Scott and Barretto [39]; and Brezing and Weng [40].

The ability to classify families of elliptic curves is great; it allows us to generate curves optimized for the cryptosystem we would like to implement. Unfortunately, no such families exist for composite-order elliptic curves [27]. Classifying composite-order curves into families, as is done for prime-order curves, would involve making public $N(x)$, where specific values $N(x_0)$ correspond to the order composite-order groups. Thus it seems likely that this would lead to the factorization of the group order, thereby compromising security. It remains an open problem to obtain parametrized families in which the prime factors of the composite integer N will be random, unguessable primes of the desired size [27].

5.4 PROPERTIES OF COMPOSITE-ORDER GROUPS

When we were looking for composite-order, pairing-friendly elliptic curves, we noted that it is important that $\#E(F_q) \cong q^k$. This means that the pairing computation will be considerably slower for a given level of security than for a pairing-friendly prime-order curve. As we will see there is current research into how to address this issue. So, other than being slower to use at a given security level, what do composite-order curves offer to the designer of a cryptosystem? Freeman [23] recently presented two definitions for properties of pairings between composite-order groups used by designers, *projecting* and *cancelling*. Very recently Seo and Cheon [24] defined an addition structure, *translating*.

5.4.1 PROJECTING PAIRINGS

Boneh, Goh, and Nissim's cryptosystem [8] takes elements $g \in G$ and $h \in G_1$, where g has order $N = p_1 p_2$ and h has order p_1, and encrypts a message M as $C = g^M h^r$, r is a random number. In this system h is used as a "blinding" term which adds randomness to the ciphertext. Decrypting the message is performed by first computing C^{p_1}, thus getting rid of h; then it is just a matter of taking the discrete log with base g^{p_1} to get M. The functionality of the system requires that this operation can be performed either before or after the pairing is computed; in other words we must be able to construct and remove blinding terms in G_t. Freeman defines this concept as *projecting*.

Definition 5.4.1 Let $\mathcal{G}$ be a bilinear generator. We say that $\mathcal{G}$ is *projecting* if it also outputs a group $G_t' \subset G_t$ and three group homomorphisms π_1, π_2, π_t mapping $G, H\ G_t$ to themselves, respectively, such that

1. $G_1, H_1, G_t{}'$ are contained in the kernels of π_1, π_2, π_t respectively, and
2. $e\big(\pi_1(g), \pi_2(h)\big) = \pi_t\big(e(g,h)\big)\ \forall g \in G, h \in H$.

5.4.2 CANCELLING PAIRINGS

In Boneh, Sahai, and Waters' traitor-tracing scheme [10] and several others, they use the fact that if two elements g, h have relatively prime order, then $e(g,h) = 1$. This essentially means that we could use the two subgroups generated by g and h to encode different information and they will remain distinct after the pairing operation. This was defined explicitly by Freeman as *r-cancelling*.

Definition 5.4.2. Let $\mathcal{G}$ be a bilinear group generator. We say that $\mathcal{G}$ is *r-cancelling* if it also outputs groups $G_2, \dots, G_r \subset G$ and $H_2, \dots, H_r \subset H$, such that

1. $G \cong G_1 \times \dots \times G_r$ and $H \cong H_1 \times \dots \times H_r$.
2. $e(g_i, h_i) = 1$ whenever $g_i \in G_i, h_j \in H_j$, and $i \neq j$.

5.4.3 TRANSLATING PAIRINGS

Very recently, Seo and Cheon [41] formally presented a new property of bilinear composite-order groups, which they called *translating*. Basically, the translating property says that given $g_1, g_1^a \in G_1, g_2 \in G_2$, where G_1 and G_2 are distinct subgroups of G, there exists a map $\mathcal{T}$ outputting g_2^a.

Definition 5.4.3. A bilinear group generator $\mathcal{G}$ is $(i, j) translating$ if there exist efficiently computable (polynomial time in λ) maps $\mathcal{T}_{i,j}\colon G_i^2 \times G_j \to G_j$ defined by $(g_i, g_i^a, g_j) \longmapsto g_j^a$ and $\tilde{\mathcal{T}}_{i,j}\colon H_i^2 \times H_j \to H_j$ defined by $\left(h_i, h_i^a, h_j\right) \mapsto h_j^a$ for an integer $a \in \mathbb{Z}$. If $\mathcal{G}$ is a symmetric bilinear group generator, then set $\widetilde{\mathcal{T}_{\iota,J}} = \mathcal{T}_{i,j}$.

6 IMPLEMENTATIONS

It is useful to see how all these concepts are used to develop cryptosystems. In this chapter we'll focus on a few of the major cryptosystems that have been developed using composite-order groups. Boneh, Goh, and Nissim's system [8] was the first system to use composite-order groups and initiated this branch of research; therefore, we look at their system first. Boneh, Sahai, and Waters [10] used composite-order groups as the basis for a fully collusion-resistant traitor tracing system, the first of its kind. These two systems highlight the properties of composite-order groups that are not shared with their prime-order counter parts.

6.1 HOMOMORPHIC PUBLIC-KEY SYSTEM

Homomorphic encryption enables the computation of encrypted data. Specifically it allows one to compute additions and multiplications on encrypted data and obtain an encrypted result, which can then be decrypted by the intended receiver. Applications such as electronic voting schemes, computational private information retrieval, and private matching are all possible using homomorphic encryption. Before Boneh, Goh, and Nissim's scheme there was an open problem as to whether an encryption scheme were possible in which one may both add and multiply; their scheme proved it was possible. In particular, their scheme allows one to compute arbitrary additions and one multiplication.

In their paper, they chose to use bilinear groups of composite order taken from supersingular elliptic curves generated as in Chapter 5. They also gave the first explicit definition of the subgroup decision problem for bilinear pairings.

We describe the three algorithms, $KeyGen$, $Encrypt$, $Decrypt$.

$KeyGen(\tau)$: Given a security parameter $\tau \in \mathbb{Z}^{+}$, $\mathcal{G}(\tau)$ to obtain a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$. Let $n = q_1, q_2$. Pick random generators $g, u \overset{R}{\leftarrow} \mathbb{G}$ and set $h = u^{q_2}$. Then h is a random generator of the subgroup of $\mathbb{G}$ of order q_1. The public key is $\mathcal{PK} = (n, \mathbb{G}, \mathbb{G}_1, e, g, h)$. The private key is $\mathcal{SK} = q_1$.

$Encrypt(\mathcal{PK}, M)$: Assume the message space consists of integers in the set $\{0,1, \dots, T\}$

with $T < q_2$. To encrypt a message m using $\mathcal{PK}$, pick a random $r \overset{R}{\leftarrow} \{0,1,\dots,n-1\}$ and compute

$C = g^m h^r \in \mathbb{G}$.

Output C as the ciphertext.

$Decrypt(\mathcal{SK}, C)$: Decrypt using $\mathcal{SK}$, observe that

$C^{q_1} = (g^m h^r)^{q_1} = (g^{q_1})^m$.

Let $\hat{g} = g^{q_1}$. To recover m, it suffices to compute the discrete log of C^{q_1} base $\hat{g}$. Because $0 \leq m \leq T$ this takes expected time $O(\sqrt{T})$ using Pollard's lamda method.

The decryption takes polynomial time in T, and thus the system above can only be used to encrypt short messages. In [8], they note that it is possible to encrypt longer messages using any mode of operation that converts cipher on a short block into a cipher on an arbitrary long block. In order to speed up the decryption, one could compute a polynomial-sized table of powers of $\hat{g}$ allowing decryption in constant time.

6.1.1 SECURITY

We observe how this system is secure under the subgroup decision assumption. The general subgroup decision assumption presented earlier has been applied to this system as follows (note that the definitions are equivalent).

Definition 6.1.1. The *bilinear subgroup generator* is an algorithm $\mathcal{G}$ that given a security parameter $\tau \in \mathbb{Z}^+$ outputs a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ where $\mathbb{G}, \mathbb{G}_1$ are groups of order $n = q_1 q_2$ and $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ is a bilinear map. On input τ, the algorithm works as follows:

1. Generate two random $\tau -$bit primes q_1, q_2 and set $n = q_1 q_2 \in doubleZ$.
2. Generate a bilinear group $\mathbb{G}$ of order n using one of the methods in Chapter 5. Let g be a generator of $\mathbb{G}$ and $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$be the bilinear map.
3. Output $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$.

Definition 6.1.2. The *subgroup decision problem* is stated as such. For an algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ in solving the subgroup decision problem $SD - Adv_{\mathcal{A}}(\tau)$ is defined as:

$$SD - Adv_{\mathcal{A}(\tau)} = |\Pr[\mathcal{A}(n, \mathbb{G}, \mathbb{G}_1, e, x) = 1 : (q_1, q_2, \mathbb{G}, \mathbb{G}_1, e) \leftarrow \mathcal{G}(\tau), n = q_1, q_2, x \leftarrow \mathbb{G})]$$
$$- |\Pr[\mathcal{A}(n, \mathbb{G}, \mathbb{G}_1, e, x^{q_2}) = 1 : (q_1, q_2, \mathbb{G}, \mathbb{G}_1, e) \leftarrow \mathcal{G}(\tau), n = q_1, q_2, x \leftarrow \mathbb{G})]|.$$

Definition 6.1.3. We say that $\mathcal{G}$ satisfies the subgroup decision assumption if for any polynomial time algorithm $\mathcal{A}$, $SD - Adv_{\mathcal{A}}(\tau)$ is a negligible function in τ.

Theorem 6.1.4. *The public key system in 6.1 is semantically secure assuming $\mathcal{G}$ satisfies the subgroup decision assumption.*

Proof. Suppose a polynomial time algorithm $\mathcal{B}$ breaks the semantic security of the system with advantage $\epsilon(\tau)$. We construct an algorithm $\mathcal{A}$ $t$hat breaks the subgroup decision assumption with the same advantage. Given $(n, G, G1, e, x)$ as input, algorithm $\mathcal{A}$ works as follows:

1. A picks a random generator $g \in \mathbb{G}$ and gives algorithm $\mathcal{B}$ the public key $(n, G, G1, e, g, x)$.

2. Algorithm $\mathcal{B}$ outputs two messages $m_0, m_1 \in \{0, 1, \ldots, T\}$ to which $\mathcal{A}$ responds with the ciphertext $C = g^{m_b} x^r \in \mathbb{G}$ for a random $b \overset{R}{\leftarrow} \{0,1\}$ and random$r$ $\overset{R}{\leftarrow} \{0,1, \ldots, n-1\}$.

3. Algorithm outputs its guess $b' \in \{0,1\}$. If $b = b'$, algorithm $\mathcal{A}$ outputs 1 (meaning *x* is uniform in a subgroup of $\mathbb{G}$); otherwise $\mathcal{A}$ outputs 0 (meaning *x* is uniform in $\mathbb{G}$).

When *x* is uniform in $\mathbb{G}$, the challenge cipher text *C* is uniformly distributed in $\mathbb{G}$ and is independent of the bit *b*. In this case $\Pr[b = b'] = 1/2$. When *x* is uniform in the q_1-subgroup of $\mathbb{G}$, then the public key and challenge *C* given in $\mathcal{B}$ are as in a real semantic security game. By definition of $\mathcal{B}$, we know that $\Pr[b = b'] > \frac{1}{2} + \epsilon(\tau)$. If follows that $\mathcal{A}$ satisfies $SD - Adv_{\mathcal{A}}(\tau) > \epsilon(\tau)$ and hence $\mathcal{A}$ breaks the subgroup decision assumption with advantage $\epsilon(\tau)$.

6.2 Fully collusion resistant traitor tracing

Traitor tracing systems are used primarily to help content distributors identify pirates. As an example, consider a broadcast system which sends encrypted data to *N* intended recipients, e.g. a satellite radio broadcast system (like SiriusXM) in which the broadcasted data is only to be played on certified receivers. Recipient *i* has secret key SK_i. The broadcast is encrypted using a public key *BK*. Any certified receiver can decrypt the data using an embedded secret key SK_i. The certified player could also enforce restrictions such as "play once" or "do not copy".

A pirate could hack a certified player, extracting its secret key. The pirate could then build a pirate decoder that could ignore any of the restrictions intended by the distributor. Even more, the pirate could make the system widely available so anyone could have unauthorized access to the data.

Enter traitor tracing. When a pirate decoder has been discovered, the distributor could run a *tracing* algorithm interacting with the pirate decoder which outputs the secret keys, SK_i, used by the pirate to create the pirate decoder. This information could be used to identify the owner of the original systems.

A traitor tracing system consists of four algorithms, *Setup, Encrypt, Decrypt, and Trace.* At a high level, the setup algorithm generates the broadcaster's key *BK*, tracing key *TK*, and the *N* recipient keys $K_1, \ldots, K_N$. *Encrypt* encrypts using *BK*, and *Decrypt* decrypts with one of the K_i. *Trace* is more complicated, interesting. We detail the formal definitions and syntax in the Appendix, as what we are really concerned with is how they used bilinear composite-order groups.

Boneh, Sahai, and Waters' traitor tracing is fully collusion resistant, meaning even a pirate with only one secret key can be traced. The ciphertext length is $O\sqrt{N}$ for *N* users. They built the system from a new primitive called *private linear broadcast encryption* (PBLE). Boneh et al. show that a traitor tracing scheme can be reduced to the construction of a PBLE scheme. They then devise a PBLE scheme and prove its security under three assumptions in bilinear groups.

A PBLE system consists of four algorithms, *Setup, Encrypt, TrEncrypt,* and *Decrypt* [23]:

$Setup(\lambda, n)$: Takes as input a security parameter λ and a positive integer *n* that is the number of users in the system. The algorithm outputs a public key *PK*, a secret key *SK*, and private keys $K_1, \ldots, K_n$, where K_i is given to user *i*.

$Encrypt(PK, M)$: Takes input *PK* and a message *M* and outputs a ciphertext *C*. This algorithm is used to encrypt a message to all *N* users.

$TrEncrypt(SK, i, M)$: Ta kes as input a *SK*, an integer *i* with $1 \leq i \leq n + 1$, and a message *M*, and outputs a ciphertext *C*. This algorithm encrypts a message to a set $\{i, \ldots, N\}$ and is primarily used for traitor tracing.

$Decrypt(j, K_j, C, PK)$: Takes input a private key K_j for user *j*, a ciphertext *C*, and the public key *PK*. The algorithm outputs a message *M* or the symbol $\perp$.

The system must satisfy the following consistency property for all $i, j \in \{1, .., N + 1\}$, with $j \leq N$, and all messages *M*:

Let $(PK, SK, (K_1, \ldots, K_n)) \overset{R}{\leftarrow} Setup(\lambda, n)$, and let $C \overset{R}{\leftarrow} TrEncrypt(SK, i, M)$.
If $j \leq i$ then $Decrypt(j, K_j, C, PK) = M$.

6.2.1 BONEH, SAHAI, WATERS PBLE

The key algebraic fact that underlies the scheme presented in [10] is that if g_p is any element from the order *p* subgroup $\mathbb{G}_p$ and g_q is any element from the order *q* subgroup $\mathbb{G}_q$, then we have: $e(g_p, g_q) = 1$. The four algorithms defined for Boneh et al.'s system are defined below.

$Setup_{LBE}(N = m^2, 1^\kappa)$:

The setup algorithm takes as input the number of users *N* and a security parameter κ. It first generates an integer $n = pq$ where p, q are random primes (whose size is determined by the security parameter). The algorithm creates a bilinear group $\mathbb{G}$ of composite order *n*. It next creates random generators $g_p, h_p \in \mathbb{G}_p$ and $g_q, h_q \in \mathbb{G}_q$ and sets $g = g_p g_q, h = h_p, h_q \in \mathbb{G}$. Next it chooses random exponents $r_1, \ldots, r_m, c_{1,} \ldots, c_m, \alpha_1, \ldots \alpha_m \in \mathbb{Z}$ and $\beta \in \mathbb{Z}_q$.

The public key include the description of the group and the following:

$$\begin{bmatrix} g, h, E = g_q^{\beta}, E_1 = g_q^{\beta r_1}, \ldots, E_m = g_q^{\beta r_m}, F_1 = f_q^{\beta r_1}, \ldots, F_m = h_q^{\beta r_m}, \\ G_1 = e(g_q, g_q)^{\beta \alpha_1}, \ldots, G_m = e(g_q, g_q)^{\beta \alpha_m}, H_1 = g^{c_1}, \ldots, H_m = g^{c_m} \end{bmatrix}$$

The private key for user (x, y) is generated as $K_{x,y} = g^{\alpha_x} g^{r_x c_y}$. The authorities secret key *K* includes factors q, p along with exponents used to generate the public key.

$TrEncrypt_{LBE}\big(K, M, (i, j)\big)$:

This algorithm is a secret key algorithm used by the tracing authority. The algorithm encrypts a message *M* to the subset of receivers that have row values greater than *i* or both row value equal to *i* and column values greater than or equal to *j*.

The encryption algorithm first chooses random

$$t \in \mathbb{Z}_n, w_1, \dots, w_m, s_1, \dots s_m \in \mathbb{Z}_n, z_{p,1}, \dots, z_{p,j-1} \in \mathbb{Z}_p,$$

$$\text{and } (v_{1,1}, v_{1,2}, v_{1,3}), \dots, \big(v_{i-1,1}, v_{i-1,2}, v_{i-1,3}\big) \in \mathbb{Z}_n^3.$$

For each row *x* we create a four ciphertext components $(R_x, \overline{R_x}, A_x, B_x)$ as follows:

$$\text{If } x > i: \quad R_x = g_q^{s_x r_x} \qquad \overline{R_x} = h_q^{s_x r_x} \qquad A_x = g_q^{s_x t} \qquad B_x = Me\big(g_q, g\big)^{a_x s_x t}$$

$$\text{If } x = i: \quad R_x = g^{s_x r_x} \qquad \overline{R_x} = h^{s_x r_x} \qquad A_x = g^{s_x t} \qquad B_x = Me(g\ \ , g)^{a_x s_x t}$$

$$\text{If } x < i: \quad R_x = g^{v_{x,1}} \qquad \overline{R_x} = h^{v_{x,1}} \qquad A_x = g^{v_{x,2}} \qquad B_x = Me(g\ \ , g)^{v_{x,t}}$$

For each column *y* the algorithm creates values $C_y, \overline{C_y}$ as:

$$\text{If } y \geq j: \quad C_y = g^{c_y t} h^{w_y} \qquad \overline{C_y} = g^{w_y}$$

$$\text{If } y < j: \quad C_y = g^{c_y t} g_p^{z_{p,y}} h^{w_y} \qquad \overline{C_y} = g^{w_y}$$

$Encrypt_{LBE(PK,M)}$:

This algorithm is used by an encryptor to encrypt a message such that all the recipients can receive it. This algorithm is used during normal (non-tracing) operation to distribute content to all receivers.

The algorithm first chooses random $t \in \mathbb{Z}_n, w_1, \dots, w_m, s_1, \dots s_m \in \mathbb{Z}_n$. For each row *x* the algorithm creates the four ciphertext components $(R_x, \bar{R}_x, A_x, B_x)$ as follows:

$$R_x = E_x^{s_x} \qquad \bar{R}_x = F_x^{s_x} \qquad A_x = E^{s_x t} \qquad B_x = MG_x^{s_x t}$$

For each column y the algorithm creates $C_y, \bar{C}_y$ as:

$$C_y = H_y^t h^{w_y} \qquad\qquad \bar{C}_y = g^{w_y}$$

$Decrypt_{LBE}\left((x, y), K_{x,y}, C\right)$:

User (x, y) uses key $K_{x,y}$ to decrypt by computing:

$$B_x \left(\frac{e\left(K_{x,y}, A_x\right) e\left(\bar{R}_x, \bar{C}_y\right)}{e\left(R_x, C_y\right)} \right)^{-1}.$$

7 Converting composite-order groups to prime-order

There is a current research thrust to identify the properties of composite-order groups which are used by cryptosystems and to create elliptic curves that generate prime-order groups with the same properties. Freeman [23] developed such a framework. In particular, he shows how it is possible that the subgroup decision problem is an analogue of the decision Diffie-Hellman assumption. He then shows how it is possible to generate product groups of prime-order curves to simulate the properties of composite-order curves, and how these can be used in existing cryptosystems. Freeman notes that his framework is not a black box and that the security proof of each cryptosystem would need to be checked to ensure that it is still valid in his framework. He gave the example of the Lewko-Waters ID-based encryption scheme and how it uses in an essential way the fact that G has two subgroups of relatively prime order; his framework is thus not valid in this example.

Lewko [42] furthered the work of Freeman by presenting a set of tools to convert composite-order bilinear systems relying on the cancelling property to prime-order systems. Using these tools she was able to expand the set of applicable cryptosystems.

Meiklejohn, Shacham, and Freeman [43] presented limitations on converting pairing-based cyptosystems from composite- to prime-order groups. In particular they proposed that it is not possible for prime-order groups to simultaneously obtain the cancelling and projecting properties.

Recently Seo and Cheon [41] presented evidence that, despite the limitations suggested by Miekeljon et al., it is indeed possible to have prime-order groups simultaneously obtain the cancelling and projecting properties. They presented a new mathematical framework for the conversion, answering several open problems from [23] and [43]. Having successfully created a framework capable of converting known composite- to prime-order groups, they posed the question whether there exist cryptosystems that are based on composite-order groups that are not realizable using prime-order groups.

8 CONCLUSIONS

We have defined elliptic curves over finite fields and shown how the set of points on elliptic curves form a group with point addition.

The Tate pairing is an efficiently computable bilinear map defined on elliptic curves. We have shown how to compute the pairing using Miller's algorithm. There are other variations of the Tate pairing that have a shorter Miller loop and are therefore more efficient from a computational standpoint.

It is possible to define the discrete log problem on elliptic curves. We examined attacks that can be used to solve the DLP on elliptic curves.

We studied pairings and how they can be used a basis for cryptosystems. We discussed finding pairing-friendly elliptic curves of composite order and the security concerns when using such curves. We discussed the properties which make composite-order groups desirable for creating unique cryptosystems.

We examined two existing protocols to see how they use the properties of composite-order bilinear groups.

We then presented the current research on attempting to convert cryptosystems which use composite-order groups to cryptosystems using prime-order groups.

Two major problems remain open concerning composite-order groups. The first is whether it is possible to classify composite-order pairing-friendly curves into families as in the prime-order case. The second is whether there exist cryptosystems using composite-order groups which are not realizable using prime-order groups.

REFERENCES

[1] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation,* vol. 48, no. 177, pp. 203-209, 1987.

[2] V. Miller, "Use of elliptic curves in cryptography," *Crypto,* vol. 85, pp. 417-426, 1985.

[3] D. Boneh and M. Franklin, "Identity based encryption from the Weil Pairing," *SIAM Journal on Computing,* vol. 32, no. 3, pp. 586-615, 2003.

[4] A. Shamir, "Identity-based cryptosystems and signature schemes," *Lecture Notes in Computer Science (Advances in Cryptology: Proceedings of CRYPTO 84),* vol. LNCS, no. 7, pp. 47-53, 1984.

[5] R. Balasubramanian and N. Koblitz, "The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm," *Journal of Cryptology 11,* pp. 141-145, 1998.

[6] D. Freeman, M. Scott and E. Teske, "A taxonomy of pairing-friendly curves," *Journal of Cryptology,* pp. 224-280, 2010.

[7] A. Koblitz, N. Koblitz and A. Menezes, "Elliptic curve cryptography: The serpentine course of a paradigm shift," *Journal of Number Theory,* vol. 131, no. 5, pp. 781-814, 2011.

[8] D. Boneh, E. Goh and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *TCC2005, LNCS 3378*, pp. 325-342, 2005.

[9] J. Groth, R. Ostrovsky and A. Sahai, "Perfect non-interactive zero knowledge for NP," *Lect. Notes in Comp. Sci.,* pp. 339-358, 2006.

[10] D. Boneh, A. Sahai and B. Waters, "Fully collusion resistant traitor tracing with short ciphertexts and private keys," in *Advances in Cryptology*, pp. 573-592, Berlin, 2006.

[11] X. Boyen and B. Waters, "Compact Group signatures wihout random oracles," *Lect. Notes in Comp. Sci,* pp. 1-15, 2007.

[12] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," *Lect. Notes in Comp. Sci,* pp. 535-554, 2007.

[13] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," *Lect. Notes in Comp. Sci.,* pp. 415-432, 2008.

[14] H. Shacham and B. Waters, "Efficient ring signatures without random oracles," *Lect. Notes in Comp. Sci,* pp. 166-180, 2007.

[15] N. Chandran, J. Groth and A. Saha, "Ring signatures of sub-linear size without random oracles," *Lect. Notes in Comp. Sci,* pp. 423-434, 2007.

[16] J. Silverman, The Arithmetic of Elliptic Curves, New York: Springer-Verlag, 1986.

[17] S. Galbraith, K. Paterson and N. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics,* vol. 156, no. 16, pp. 3113-3121, 2008.

[18] I. Blake, G. Seroussi and N. Smart, Advances in elliptic curve cryptography, Cambridge University Press, 2005.

[19] G. Frey, M. Muller and H. Ruck, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," *IEEE Tans. Inf. Theory,* vol. 45, p. 1717–1719, 1999.

[20] V. Miller, "Short programs for functions on curves," unpublished manuscript, 1986.

[21] W. Diffie and M. Hellman, "Multiuser cryptographic techniques," *IEEE Trans. on Info. Theory,* vol. 45, pp. 109-112, 1976.

[22] D. Stinson, Cryptography: Theory and Practice, Boca Raton, FL: Chapman & Hall/CRC, 2006.

[23] D. Freeman, "Converting pairing-based cryptosystems from composite order groups to prime-order groups," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. LNCS, no. 6110, pp. 44-61, 2010.

[24] J. Seo and J. Cheon, "Beyond the limitation of prime-order bilinear groups, and round optimal blind signatures," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. LNCS, no. 7194, pp. 133-150, 2012.

[25] K. Gjosteen, "Subgroup membership problems and public key cryptosystems," Ph.D. dissertation, Norweigan University of Science and Technology, 2004.

[26] D. Freeman, M. Scott and E. Teske, "A taxonomy of pairing-friendly elliptic curves," *Journal of Cryptology,* vol. 23, no. 2, pp. 224-280, 2010.

[27] D. Boneh, K. Rubin and A. Silverberg, "Finding composite order ordinary elliptic curves using the Cocks-Pinch method," *Journal of Number Theory,* vol. 131, no. 5, pp. 832-841, 2011.

[28] J. Pollard, "Monte Carlo methods for index computation (mod p)," *Mathematics of Computation,* vol. 32, pp. 918-924, 1978.

[29] P. van Oorschot and M. Wiener, "Parallel collision search with cryptanalytic applications," *Journal of Cryptology,* vol. 12, pp. 1-18, 1999.

[30] A. Odlyzko, "Discrete logarithms in finite fileds and their cryptographic significance," *Advances in Cryptology - Eurocrypt 1984,* vol. LNCS, no. 209, pp. 224-314, 1985.

[31] F. Luca and I. Shparlinski, "Elliptic curves with low embedding degree," *Journal of Cryptology,* vol. 19, pp. 553-562, 2006.

[32] M. Scott, "Computing the Tate pairing," *Topics in Cryptology,* vol. LNCS, no. 3376, pp. 293-304, 2005.

[33] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Trans. Inf. Theory 39,* pp. 1639-1646, 1993.

[34] W. Waterhouse, "Abelian varieties over finite fields," *Ann. Sci. Ecole Norm. Sup,* vol. IV, no. 2, pp. 521-560, 1969.

[35] R. Lidl and H. Niederreiter, Finite Fields, Cambridge University Press, 1997.

[36] N. Koblitz, "A security weakness in composite order pairing-based protocols with embedding degree k>2," Unpublished manuscript, 2010.

[37] M. Miyaji, M. Nakabayashi and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals,* Vols. E84-A, no. 5, pp. 1234-1243, 2001.

[38] P. Barreto, B. Lynn and M. Scott, "Constructing elliptic curves with prescribed embedding degree," *Security in Communication Networks,* vol. 2576, pp. 263-273, 2002.

[39] M. Scott and P. Barreto, "Generating more MNT elliptic curves," *Designs, Codes, and Cryptography,* vol. 38, pp. 209-217, 2006.

[40] F. Brezing and A. Weng, "Elliptic curves suitable for pairing based cryptography," *Designs, Codes, and Cryptography,* vol. 37, pp. 133-141, 2005.

[41] J. Seo and J. Cheon, "Beyond the limitation of prime-order bilinear groups, and round optimal blind signatures," *Lecture Notes in Computer Science,* vol. LNCS, no. 7194, pp. 133-150, 2012.

[42] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. LNCS, no. 7237, pp. 318-335, 2012.

[43] S. Meiklejohn, H. Shacham and D. Freeman, "Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. LNCS, no. 6477, pp. 519-538, 2010.