2011

The Weil Pairing on Elliptic Curves and Its Cryptographic Applications

Alex Edward Aftuck
University of North Florida

The Weil Pairing on Elliptic Curves and its Cryptographic Applications

Alex Edward Aftuck

A thesis submitted to the Department of Mathematics and Statistics
in partial fulfillment of the requirements for the degree of

Master of Sciences in Mathematical Sciences

UNIVERSITY OF NORTH FLORIDA
COLLEGE OF ARTS AND SCIENCES

July 2011

The thesis of Alex Edward Aftuck is approved: (Date)

Signature Deleted

07/21/2011

Signature Deleted

07/21/2011

Signature Deleted

7/21/2011

Committee Chairperson

Accepted for the Department:

Signature Deleted

7/26/2011

Chairperson

Accepted for the College:

Signature Deleted

28 JULY 2011

Dean

Accepted for the University:

Signature Deleted

8/15/11

Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

This thesis presents the Weil pairing on elliptic curves as a tool to implement a tripartite Diffie-Helman key exchange. Elliptic curves are introduced, as well as the addition operation that creates a group structure on its points. In leading to the definition of the Weil pairing, divisors of rational functions are studied, as well as the Weierstrass $\wp$-function, which shows the complex lattice as isomorphic to elliptic curves. Several important qualities of the Weil pairing are proved, and Miller's algorithm for quick calculation is shown. Next, the bipartite Diffie-Helman key exchange is discussed over finite fields and elliptic curves. Finally an example of a modified Weil pairing is defined, which leads to the tripartite Diffie-Helman key exchange.

0 Introduction

The Diffie-Helman key exchange, created in 1976, is one of the most widely used cryptographic tools, with several different versions. The process allows two people to create a common key to a cypher, even if there are eavesdroppers listening to their conversation. This process is based on the discrete logarithm problem and, later, the elliptic curve discrete logarithm problem. It wasn't until the early 2000's before a process was developed to allow three people to share a key easily. We want the process to not require more than one round of communications, as this would require all parties to be online at once. Anotoine Joux created an effective tripartite Diffie-Helman key exchange, thus allowing three people to easily share a key. His work brought in bilinear pairings on elliptic curves, and used them to create the Diffie-Helman process.

The bilinear pairing on elliptic curves we focus on is the Weil pairing, although there are others. The Weil pairing was given its original abstract definition by Andre Weil, but was not fully realized on elliptic curves until later by other mathematicians. Originally, Weil pairings on elliptic curves were introduced to cryptography not as a constructive mechanism, as Joux used them, but as a way to hopefully break the elliptic curve discrete logarithm problem, and thus unravel the Diffie-Helman key exchange. In the early 1990's the MOV algorithm [1] was designed to take the elliptic curve discrete logarithm problem in a finite field of size p and turn it into a discrete logarithm problem in a finite field of size p^k, for some integer k. Unfortunately, the discrete logarithm problem is still not easily breakable, and thus this method is only

truly effective on very specific curves (curves where the the embedding degree k is very small.) Thus, the elliptic curve discrete logarithm problem was not harmed too much, and eventually it was even strengthened by the work of Joux, who ironically used the pairings originally meant to weaken Diffie-Helman to strengthen it.

Sections 1 and 2 serve as an introduction to elliptic curves. In Section 1, we arrive at our definition of an elliptic curve and view the "addition operation," both in its natural form of drawing a line through two points and its more analytical form. We then see how this operation gives the structure of a group to points on an elliptic curve. In Section 2, we review further aspects of elliptic curves, such as the order of points, as well as view an algorithm used to add a point to itself multiple times.

In section 3, we begin setting up the basis for Weil pairings by reviewing the divisors of rational functions. We also see how elliptic curves are related to complex lattices using the Weierstrass $\wp$-function, which will allow us to prove a theorem concerning divisors. In Section 4, we review bilinear pairings, give our definition of the Weil pairing, and prove several of its qualities. In Section 5, we view Miller's algorithm, which gives us a quick way of finding the functions necessary for the Weil pairing, and will ultimately give us a way to easily calculate the pairing. Section 6 will shift our view from elliptic curves over complex numbers to elliptic curves over finite fields, which is necessary for cryptographic applications.

The final sections of the paper focus on cryptography and serve to link the more abstract ideas of elliptic curves with the very practical ideas of cryptography. In Section 7, we view the Diffie-Helman key exchange, a cryptographic tool based only on modular arithmetic which becomes exceedingly time consuming due to the discrete

logarithm problem (DLP). In Section 8, we view its elliptic curve analog, made even harder by the elliptic curve discrete logarithm problem (ECDLP). Finally, in Section 9, we show the implementation of the Weil pairing as a cryptographic tool. This section will end with an example of the tripartite Diffie-Helman key exchange, made possible by the implementation of the modified Weil pairing.

The Appendices will feature Mathematica programs that implement the algorithms and functions reviewed in the paper. Appendix A features a program that puts a general homogenous cubic polynomial into Weierstrass form, which may be needed for the other programs to correctly function. Appendix B features the algorithms and functions of the paper implemented when the underlying field is $\mathbb{C}$. Appendix C features the same functions when the underlying field is $\mathbb{F}_p$, and Appendix D features the same functions when the underlying field is $\mathbb{F}_{p^2}$, as well as a few functions to easily calculate in said field.

1 Elliptic curves

An elliptic curve, E, is defined as the set of solution points to an equation of the form $Y^2 = X^3 + aX + b$. As it turns out, there is a natural operation upon the points of an elliptic curve that give an abelian group. If we consider two points P and Q, when we draw a line through them, we get a third point on the curve, R. This operation is referred to as the pound ($\#$) operation, giving us $P\#Q = R$ in the above example. In order to obtain an operation which gives us an abelian group, we then reflect our point R over the x-axis (by negating the y value of R). This new point

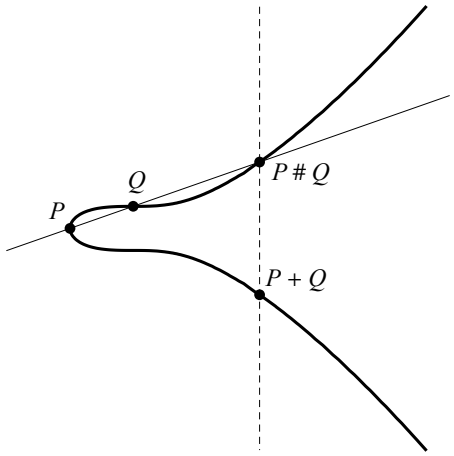

Figure 1: Elliptic curve addition on $E : Y^2 = X^3 + 1$

will be called R', and our operation uses the addition notation, that is, $P + Q = R'$. This is illustrated in Figure 1 on the curve $E : Y^2 = X^3 + 1$.

If we were to add a point P to itself, we do not have a second point on E to draw our line with. Thus, we must consider the tangent line of the curve E at P. In this case, illustrated in Figure 2, we have a line through two points, one of which with multiplicity two. When adding a point to itself, we use slightly modified addition notation, as in: $P + P = [2]P$, $P + P + P = [3]P$, etcetera, so that

$$\underbrace{P + P + \cdots + P}_{n} = [n]P$$

If we add a point $P = (x, y)$ belonging to E to its reflection over the x-axis, $(x, -y)$, (denoted $-P$), the line through the two points is vertical, with no third point of intersection on the XY-plane. Thus, we must include an extra point in E, which can be seen as a point at infinity at the end of every curve (this is discussed further below.) This point is referred to as $\mathcal{O}$. Then $P\#(-P) = \mathcal{O}$, and since $\mathcal{O}$

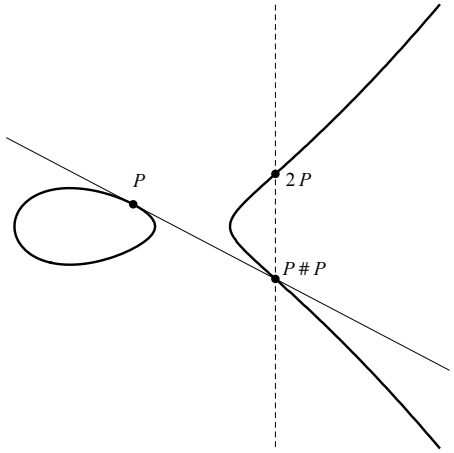

Figure 2: Adding P to itself on $E : Y^2 = X^3 - 3X + 1$

is not on the XY-plane, reflecting it still gives $\mathcal{O}$, thus $P + (-P) = \mathcal{O}$. How does $\mathcal{O}$ behave when added to other points? Drawing a line through $P = (x, y)$ and $\mathcal{O}$ gives a vertical line $X = x$, which intersects E at P,$\mathcal{O}$ and $-P$. Thus $P\#\mathcal{O} = -P$, and negating the y-value, we have $P + \mathcal{O} = P$. This addition is shown below to be commutative, and thus $\mathcal{O}$ acts as the identity.

Now that we have an idea of the structure of elliptic curves, it is time to look at the background of the material before we define explicitly the curves we will study. In its most general form, an elliptic curve is any nondegenerate, nonsingular cubic polynomial. These are of the form

$$aX^3 + bX^2Y + cXY^2 + dY^3 + eX^2 + fX + gXY + hY^2 + jY + k = 0$$

with the coefficients belonging to the field of choice. Obviously, this ten coefficient polynomial is unwieldy to work with. This is where projective coordinates help us.

Definition 1.1. For field K, the set KP^2, the *projective plane*, is the set of equiva-

lence classes on $K^3 - (0,0,0)$ defined by the relation:

$$(x_1, y_1, z_1) \equiv (x_2, y_2, z_2) \text{ if and only if } (x_1, y_1, z_1) = \lambda(x_2, y_2, z_2) \text{ for some } \lambda \neq 0.$$

The representative classes are denoted $[x, y, z]$.

Then K^2, a standard plane, is a subset of KP^2 (as $(x, y) \to [x, y, 1]$) and is referred to as the *affine plane*. Since in the projective plane, equivalence classes are determined by a nonzero constant, $(2, 4, 3) \equiv (2/3, 4/3, 1)$, but $(2, 4, 3) \not\equiv (2, 4, 0)$. Thus, we will consider two groups of classes: $[x, y, 1]$ and $[x, y, 0]$. $[x, y, 1]$ makes up the affine plane, while $[x, y, 0]$ is not covered by the affine plane, and makes up our points at infinity.

An affine curve of the form $aX + bY + c = 0$ might be considered as having two ends, one heading towards positive infinity, one to negative infinity. These points are $(-b, a, 0)$ and $(a, -b, 0)$. But $(-b, a, 0) \equiv (b, -a, 0)$, so we consider the line as having one point at infinity, the representative class $[-b, a, 0]$.

In order to work with affine curves in the projective plane, we need the following definitions:

Definition 1.2. A *homogeneous polynomial* over KP^2 is a polynomial in three variables with coefficients in K such that each monomial is of the same degree.

Example 1.3. The affine curve $aX + bY + c = 0$ is not a homogeneous polynomial, but $aX + bY + cZ = 0$ is. Similarly, the unit circle $X^2 + Y^2 = 1$ is represented as $X^2 + Y^2 = Z^2$ homogeneously.

Then in order to work between the two forms, we need to be able to "homogenize"

and "dehomogenize".

Definition 1.4. Let $f(X, Y)$ be an affine polynomial of degree n. Then

$$(X, Y, Z) = Z^n f({}^X/_Z, {}^Y/_Z)$$

is a homogeneous polynomial in three variables.

Similarly, if $F(X, Y, Z)$ is a homogeneous polynomial in three variables,

$$f(X, Y) = F(X, Y, 1)$$

is an affine polynomial.

Just as the projective plane partitions the points of $K^3 - (0, 0, 0)$, there are equivalent projective curves.

Definition 1.5. Two projective curves $F(X, Y, Z)$ and $G(X, Y, Z)$ are *projectively equivalent* if there exists a nondegenerate 3×3 matrix A which creates the change of variables:

$$\begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix} = A \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

such that $F(\bar{X}, \bar{Y}, \bar{Z}) = G(X, Y, Z)$.

Theorem 1.6. *Let $f(X, Y, Z)$ be a nondegenerate, nonsingular cubic curve. Then by a projective transform, $f(X, Y, Z)$ can be rewritten in the form of*

$$Y^2 Z = X^3 + aXZ^2 + bZ^3,$$

where a, b belong to the field of concern, and the discriminant, $\Delta_f = 4a^3 + 27b^2$, of the cubic is nonzero. After de-homogenizing, the affine cubic version of this curve will have the form

$$Y^2 = X^3 + aX + b,$$

known as the Weierstrass form.

Weierstrass equations are among the most common form of elliptic curves studied. The above theorem is proved in many texts on elliptic curves, such as [3]. Included in Appendix A is a Mathematica program which will take any $f(X, Y, Z)$ in the form of the theorem and transform it to Weierstrass form.

In the introduction, we mentioned the point at infinity $\mathcal{O}$. Projective geometry allows us to describe it in more detail. As X goes to infinity, Y goes to positive and negative infinity, with the slope of our tangent line nearing vertical. Then, our tangent line at infinity is projectively equivalent to the line $X = 0$. Then since a line $aX + bY + c = 0$ has point at infinity $[-b, a, 0]$, our tangent line has point at infinity $[0, 1, 0]$. Then $\mathcal{O} = [0, 1, 0]$.

Definition 1.7. An *elliptic curve*, E, is defined as the set of solution points to the Weierstrass equation $Y^2 = X^3 + aX + b$, in addition to the point $\mathcal{O}$, with the condition that the discriminant of the cubic is nonzero: $\Delta_E = 4a^3 + 27b^2 \neq 0$.

We want the discriminant to be nonzero to avoid any singularities in the curve. If we had a point of singularity P on E, we would have much trouble computing $[2]P$. Since the tangent line at P is not well defined, neither is our addition operation. Now

we formalize the addition operation discussed earlier. Included in the Appendices are Mathematica programs that implement the addition algorithm over several fields.

Theorem 1.8. [1] *Elliptic Curve Addition Algorithm.*

Let E be an elliptic curve and let P_1 and P_2 be points on E.

(a) *If $P_1 = \mathcal{O}$, then $P_1 + P_2 = P_2$.*

(b) *If $P_2 = \mathcal{O}$, then $P_1 + P_2 = P_1$.*

Otherwise, let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$.

(c) *If $x_1 = x_2$ and $y_1 = -y_2$, then $P_1 + P_2 = \mathcal{O}$.*

(d) *Otherwise, let*

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } P_1 = P_2 \end{cases}$$

Then $P_1 + P_2 = (x_3, y_3)$, where:

$$x_3 = \lambda^2 - x_1 - x_2 \text{ and } y_3 = \lambda(x_1 - x_3) - y_1$$

Proof Part (a) is clear when considering that the line through P_2 and $\mathcal{O}$ is a vertical line, thus intersecting E at the third point of $-P_2$. Reflecting over the x-axis, $\mathcal{O}+P_2 = P_2$. Part (b) follows similarly.

If $P_2 = -P_1$, as in part (c), the line through the points is $X = x_1$. This vertical line line intersects E at $\mathcal{O}$ by definition, thus $P_1 \# P_2 = \mathcal{O}$, and so it follows that $P_1 + P_2 = \mathcal{O}$.

In either case for (d), the line through P_1 and P_2 has slope λ . Thus the line has the equation

$$Y = \lambda(X - x_1) + y_1 \tag{1}$$

Substituting Y into E and simplifying, we have:

$$(\lambda(X - x_1) - y_1)^2 = X^3 + aX + b$$

$$X^3 - \lambda^2 X^2 + X(a + 2\lambda^2 x_1) + (b - \lambda^2 x_1^2 - y_1^2) = 0 \tag{2}$$

Since we already know x_1, x_2 are roots of the above cubic, if we let x_3 be the third root, we have:

$$(X - x_1)(X - x_2)(X - x_3) = 0$$

$$X^3 - (x_1 + x_2 + x_3)X^2 + (x_1x_2 + x_1x_3 + x_2x_3)X - x_1x_2x_3 = 0 \tag{3}$$

Equating the coefficients of X^2 in (2) and (3), we have $x_1 + x_2 + x_3 = \lambda^2$. Thus $x_3 = \lambda^2 - x_1 - x_2$. Now, substituting x_3 into (1), we obtain $Y = \lambda(x_3 - x_1) + y_1$. Negating the Y-value, we have $y_3 = \lambda(x_1 - x_3) - y_1$ and our proof is complete. □

Example 1.9. Let $E : Y^2 = X^3 + 1$. Then considering points $P = (-1, 0)$ and

$Q = (0, 1)$, as we did in Figure 1, we compute $P + Q$.

$$\lambda = \frac{1 - 0}{0 + 1} = 1$$

$$x_3 = 1^2 + 1 - 0 = 2$$

$$y_3 = 1(0 - 2) - 1 = -3$$

Thus $P + Q = (2, -3)$.

Theorem 1.10. [1] *Let E be an elliptic curve. Then the addition operation on E has the following properties for all $P, Q, R \in E$:*

(a) *Identity:* $P + \mathcal{O} = \mathcal{O} + P = P$.

(b) *Inverse:* $P + (-P) = \mathcal{O}$.

(c) *Associative:* $(P + Q) + R = P + (Q + R)$.

(d) *Commutative:* $P + Q = Q + P$.

Thus, E with the addition operation is an abelian group.

Proof The proof of (a) and (b) follow from the proof of Theorem 1.8 parts (a) and (c), respectively.

The proof of commutativity, (d), is easily pictured. The line through P and Q is unique, regardless of which point is considered first. Thus, the third point of intersection will be the same, regardless of order. Negating the equal y-values, we have $P + Q = Q + P$.

The proof of associativity, (c), is a matter of following the algebra through a number of cases, and will not be shown here. It is shown in [3]. □

2 Points of finite order, the Double-and-Add algorithm for computing $[n]P$

A natural question to ask is: how many times we can add a point to itself before we end up at our point at infinity $\mathcal{O}$?

Definition 2.1. We define the *order* of a point P on E as a positive integer m such that $[m]P = \mathcal{O}$. If no such integer exists, we say P is of *infinite order*. For positive integer values of m, and some field K, we denote the collection of points on $E(K)$ of order m as

$$E(K)[m] = \{P \in E : [m]P = \mathcal{O}\}.$$

Often, the field K is set throughout an example, and we simply denote the collection as $E[m]$.

Interestingly, by this definition of order, a point of finite order P does not only belong to one $E[m]$, but infinitely many. This can be seen in the case of $\mathcal{O}$. For every positive integer value of m, $[m]\mathcal{O} = \mathcal{O}$. Thus $\mathcal{O} \in E[m]$ for every positive integer m. Similarly, if $P \neq \mathcal{O}$ and $[n]P = \mathcal{O}$, then $[2n]P = [n]P + [n]P = \mathcal{O} + \mathcal{O} = \mathcal{O}$, $[3n]P = \mathcal{O}$, etcetera. Thus P belongs to every $E[km]$ for multiples of m.

Example 2.2. We look at points of order 2. If P were to be of order 2, then

$P+P=\mathcal{O}$, so essentially, $P=-P$. Thus if $P=(x,y)$, we need $y=-y$ for P to have order 2. Then $P=(x,0)$. To find these points, consider $E: Y^2=X^3+aX+b$. We know that if we are working in an algebraically closed field, such as $\mathbb{C}$, we can factor the equation into the form $E: Y^2=(X-\alpha_1)(X-\alpha_2)(X-\alpha_3)$, and since we have demanded in our Weierstrass form for E that the right hand side be nondegenerate, $\alpha_1 \neq \alpha_2 \neq \alpha_3$. Thus, we know three distinct points on E, $P_1=(\alpha_1,0), P_2=(\alpha_2,0)$, and $P_3=(\alpha_3,0)$, each of which is in $E[2]$. Also, by the argument above, $\mathcal{O} \in E[2]$. Since we know that E can have no more than 3 zeros, there are no other points in $E[2]$.

We have an addition algorithm to compute the coordinates for points such as $P+Q$, $[2]P$, etcetera, and we see that computing $[n]P$ to find the order of P will be used. What if we wish to compute, say, $[5400]P$? We could use the addition algorithm of Theorem 1.8 repeatedly, but this would be extremely inefficient. In order to be able to compute $[n]P$ for very large values of n, which we will need to be able to do for applications in cryptography, the Double-and-Add algorithm [1] has been developed.

First, consider n written as its binary expansion:

$$n=n_0+2n_1+4n_2+8n_3+\cdots+2^r n_r \quad \text{with} \quad n_o,n_1,\ldots,n_r \in \{0,1\}.$$

Then, with our addition operation, we can write $[n]P$ as:

$$[n]P=[n_0]P+[2n_1]P+[4n_2]P+[8n_3]P+\cdots+[2^r n_r]P \quad \text{with} \quad n_o,n_1,\ldots,n_r \in \{0,1\}.$$

1. Set $Q = P$ and $R = \mathcal{O}$.
2. Loop while $n > 0$:
    3. If $n \equiv 1 \bmod 2$, set $R = R + Q$.
    4. Set $Q = [2]Q$ and $n = \lfloor {}^{n}/_{2} \rfloor$.
    5. If $n > 0$, continue loop at step 2.
6. Return R which equals $[n]P$

Table 1: The Double-and-Add algorithm

Now, we notice that in each additive term, we are adding P to itself 2^i times, with $i = 0, 1, \ldots, r$. Then if we let $Q_i = [2^i]P$ (thus $Q_0 = P$), we can write

$$[n]P = [n_0]P + [n_1]Q_1 + [n_2]Q_2 + [n_3]Q_3 + \cdots + [n_r]Q_r \quad \text{with} \quad n_o, n_1, \ldots, n_r \in \{0, 1\}$$

where each $Q_i = [2]Q_{i-1}$. Now, with each step of addition, we are only completing one operation of the elliptic curve addition algorithm, as opposed to 2^i operations of it. Table 1 formalizes how the Double-and-Add algorithm will be completed in a program, and Mathematica code is included in the Appendices.

Example 2.3. Table 2 gives an example of a calculation using the algorithm given in Table 1. In the table, n represents the n used at that step, as defined in the algorithm. We calculate $[245]P$, where $P = (-2, -10)$, on the curve:

$$E : Y^2 = X^3 + 13X + 134$$

Our algorithm results, after rounding, in $[245]P = (78.034, -690.160)$.

step i	n	$Q = [2^i]P$	R
0	245	(-2,-10)	$\mathcal{O}$
1	122	(5.563,19.4531)	(-2,-10)
2	61	(-3.727,5.813)	(-2,-10)
3	30	(29.559,-162.314)	(89.597,848.853)
4	15	(6.730,-22.941)	(89.597,848.853)
5	7	(-2.932,-8.408)	(14.351,-57.239)
6	3	(11.184,40.965)	(-3.437,6.981)
7	1	(0.085,11.624)	(-2.344,-9.521)
8	0	(0.144,-11.656)	(78.034,-690.160)

Table 2: Double-and-Add algorithm used to compute $[245]P$

Note that

$$245 = 1 + 2^2 + 2^4 + 2^5 + 2^6 + 2^7$$

$$\text{thus } [245]P = P + [4]P + [16]P + [32]P + [64]P + [128]P$$

and since, in our algorithm, we compute each doubling of P with one point addition, we only need to compute up to 15 point additions (7 doublings, up to 8 additions of $R + Q$), as opposed to 245. In reality, since 245 has 6 nonzero elements in its binary expansion, we computed $[245]P$ in 13 point additions.

3 Divisors of rational functions

In order to define our bilinear pairing, we need to study how a rational function on E relates to its zeros and poles. To begin, we look at the simpler example of a rational

function of one variable. A rational function is simply a ratio of two polynomial functions, thus we may state that a general rational function of one variable takes the form:

$$F(X) = \frac{a_0 + a_1 X + \cdots + a_m X^m}{b_0 + b_1 X + \cdots + b_n X^n}.$$

If we allow factorization over the complex numbers, we can find $\alpha_1, \alpha_2, \ldots, \alpha_r$ and $\beta_1, \beta_2, \ldots, \beta_s$ such that our function takes the form:

$$F(X) = \frac{a(X-\alpha_1)^{d_1}(X-\alpha_2)^{d_2} \cdots + (X-\alpha_r)^{d_r}}{b(X-\beta_1)^{e_1}(X-\beta_2)^{e_2} \cdots + (X-\beta_s)^{e_s}}.$$

We may assume that the α's and β's are distinct, otherwise the corresponding factors could be canceled out. We denote any value of X where the numerator vanishes as a zero, and any value of X where the denominator vanishes as a pole.

Then F has *zeros* at $\alpha_1, \alpha_2, \ldots, \alpha_r$ and *poles* at $\beta_1, \beta_2, \ldots, \beta_s$, where each zero α_i has *multiplicity* d_i, and each pole β_j has multiplicity e_j, we define the *divisor* of F, $\mathrm{div}(F)$, as the formal sum

$$\mathrm{div}(F) = d_1(\alpha_1) + d_2(\alpha_2) + \cdots + d_r(\alpha_r) - e_1(\beta_1) - e_2(\beta_2) - \cdots - e_s(\beta_s).$$

This is effectively a way of keeping track of the zeros and poles of a function. Note in our definition the zeros have positive coefficients, the poles have negative. Also note the use of parentheses to distinguish we are not actually computing, say, $e_1\beta_1$ by using multiplication, but that we simply have a pole β_1 with multiplicity e_1.

Just as we have looked at divisors of functions of one variable, we can consider

rational functions of two variables, $f(X, Y) : E \to \mathbb{C}$, and look at their divisors.

Definition 3.1. In its most general form, a *divisor*, D on a curve C is a sum

$$D = \sum_{P \in C} n_P(P)$$

where all but a finite number of n_P are zero. The *degree* of a divisor is the sum of the coefficients:

$$\deg(D) = \sum_{P \in C} n_P.$$

On an elliptic curve, E, the *sum* of a divisor is the result of dropping the parentheses from the divisor, thus using the addition algorithm add each P to itself n_P times, and then sum every resultant point.

$$\text{sum}(D) = \sum_{P \in E} [n_P]P.$$

Example 3.2. Consider the elliptic curve $E : Y^2 = X^3 + aX + b$. As we have done previously, the cubic on the right side may be factorized to give an equation of the form: $Y^2 = (X - \alpha_1)(X - \alpha_2)(X - \alpha_3)$, with each root being distinct.

If we consider the function $f(X, Y) = Y$, as a rational function, we have that f vanishes at three points, $P_1 = (\alpha_1, 0), P_2 = (\alpha_2, 0)$, and $P_3 = (\alpha_3, 0)$, giving us three zeros, each with mutiplicity 1. To find the poles, we must remember that as a rational function, $Y = {}^{Y}/_{1}$. Homogenizing, we have $f(X, Y, Z) = {}^{Y}/_{Z}$. Then to find the poles of f, we must analyze Z as a polynomial. For all affine points on E, $Z = 1$, so $Z = 0$ only at $\mathcal{O}$. As a result of our projective transformation, it is easy to show that $\mathcal{O}$ is

an inflection point and $Z = 0$ is the tangent line at that point. This gives us that the pole $\mathcal{O}$ has multiplicity 3. Thus,

$$\operatorname{div}(Y) = (P_1) + (P_2) + (P_3) - 3(\mathcal{O}).$$

It is obvious that $\deg(\operatorname{div}(Y)) = 0$, and basic calculation using the addition algorithm gives $\operatorname{sum}(\operatorname{div}(Y)) = \mathcal{O}$.

The general definition of a divisor as a sum of arbitrary points is quite loose, and may not actually give us the divisor of a rational function f on E. Thus it is important to characterize which divisors D have corresponding functions f on E such that $D = \operatorname{div}(f)$.

Theorem 3.3. [2] *For an elliptic curve E,*

(a) *$D = \sum n_P P$ is a divisor of a function f on E if and only if $\deg(D) = 0$ and $\operatorname{sum}(D) = \mathcal{O}$.*

(b) *$\operatorname{div}(f) = 0$ if and only if f is a constant function.*

(c) *For functions f and f', $\operatorname{div}(f) = \operatorname{div}(f')$ if and only if f is a nonzero constant multiple of f'.*

To prove this theorem, we now find a group isomorphic to the elliptic curve and prove it for that group, which will hopefully be easier.

Definition 3.4. Given $\omega_1, \omega_2 \in \mathbb{C}$ with ω_2 not a real multiple of ω_1, we define the lattice $\Lambda = \langle \omega_1, \omega_2 \rangle = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$.

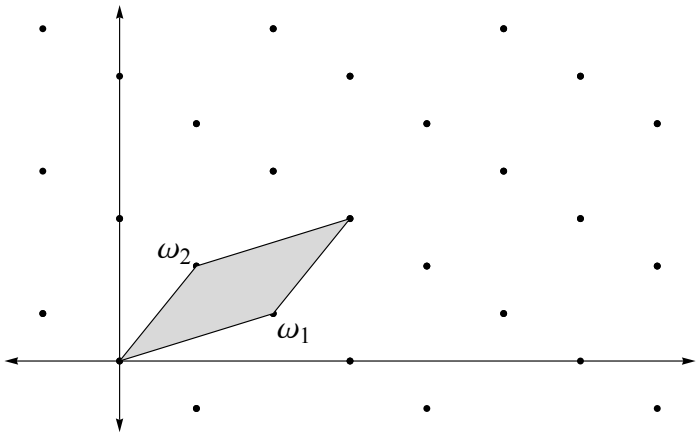

Figure 3: Λ and the period parallelogram for ω_1, ω_2

Theorem 3.5. [2] *Given any elliptic curve $E(\mathbb{C})$, there exists $\omega_1, \omega_2 \in \mathbb{C}$ and a doubly periodic map $\wp : \mathbb{C}/\langle \omega_1, \omega_2 \rangle \to E(\mathbb{C})$ that is a group isomorphism. The map is the Weierstrass $\wp$-function:*

$$\wp(z) = \frac{1}{z^2} + \sum_{\substack{\omega \in \Lambda \\ \omega \neq 0}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right)$$

As mentioned above, $\wp$ is doubly periodic, and its periods are ω_1 and ω_2. That is, for $z \in \mathbb{C}$,

$$\wp(z + \omega_1) = \wp(z) \text{ and } \wp(z + \omega_2) = \wp(z).$$

So, if $\omega \in \Lambda$, that is, $\omega = m\omega_1 + n\omega_2$ for some $m, n \in \mathbb{Z}$, then for $z \in \mathbb{C}$,

$$\wp(z + \omega) = \wp(z + m\omega_1 + n\omega_2) = \wp(z).$$

So, since $\wp$ effectively does modulo arithmetic on $\mathbb{C}$, our quotient group $\mathbb{C}/\Lambda$ can be considered as just the period parallelogram defined by ω_1 and ω_2, shown in Figure 3.

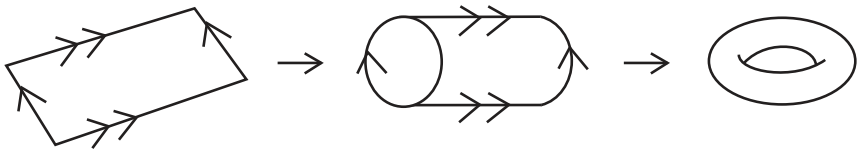

Figure 4: The period parallelogram mapped to a torus under $\wp$.

Points on the lattice, $\omega = m\omega_1 + n\omega_2$ for some $m, n \in \mathbb{Z}$, are mapped to the same point that the origin is mapped to:

$$\wp(\omega) = \wp(0 + \omega) = \wp(0).$$

Corresponding points on the border lines are mapped to the same points. Let z_1 be on the bottom border of the parallelogram, that is, $z_1 = a\omega_1$, where $0 < a < 1$, and $z_2 = a\omega_1 + \omega_2$, which is on the top border.

$$\wp(z_2) = \wp(a\omega_1 + \omega_2) = \wp(a\omega_1) = \wp(z_1)$$

Thus, the top and bottom borders of the period parallelogram are equivalent. By a similar argument, we have that the left and right borders are equivalent, which shows that the period parallelogram is essentially a torus (see Figure 4) which is mapped into $E(\mathbb{C})$.

We moved to looking at the complex lattice to help prove Theorem 3.3, so we now look at rational functions on $\mathbb{C}/\Lambda$. Specifically, we want rational functions $f : \mathbb{C}/\Lambda \to \mathbb{C}$. Due to the discussed doubly periodic structure of $\wp$, we also want our f to be doubly

periodic with periods ω_1, ω_2. Such functions are known as *elliptic functions*. The following definition helps us create elliptic functions.

Definition 3.6. The Weierstrass σ-function for lattice Λ is defined as:

$$\sigma(z) = z \prod_{\substack{\omega \in \Lambda \\ \omega \neq 0}} \left(1 - \frac{z}{\omega}\right) \exp\left(-\left(\frac{z}{\omega}\right) - \frac{1}{2}\left(\frac{z}{\omega}\right)^2\right).$$

Lemma 3.7. [2] *The Weierstrass σ-function satisfies:*

(a) *For all $\omega \in \Lambda$, $\sigma(\omega) = 0$. Specifically, $\sigma(0) = 0$.*

(b) $\dfrac{d^2}{dz^2} \log(\sigma(z)) = -\wp(z)$ *for all $z \notin \Lambda$.*

(c) *For all $\omega \in \Lambda$, there exist constants $a, b \in \mathbb{C}$ such that $\sigma(z + \omega) = \sigma(z)e^{az+b}$.*

Proof For part (a), we can easily see that substituting ω into σ, the first portion of the infinite product will become zero.

For part (b), we consider

$$\begin{aligned} \log(\sigma(z)) &= \log\left(z \prod_{\substack{\omega \in \Lambda \\ \omega \neq 0}} \left(1 - \frac{z}{\omega}\right) \exp\left(-\left(\frac{z}{\omega}\right) - \frac{1}{2}\left(\frac{z}{\omega}\right)^2\right)\right) \\ &= \log(z) + \sum_{\substack{\omega \in \Lambda \\ \omega \neq 0}} \left(\log\left(1 - \frac{z}{\omega}\right) - \left(\frac{z}{\omega}\right) - \frac{1}{2}\left(\frac{z}{\omega}\right)^2\right) \end{aligned}$$

Thus, differentiating term by term,

$$\frac{d}{dz}\log(\sigma(z)) = \frac{1}{z} + \sum_{\substack{\omega\in\Lambda\\ \omega\neq 0}} \left(-\left(\frac{1}{(\omega - z)}\right) - \left(\frac{1}{\omega}\right) - \left(\frac{z}{\omega^2}\right)\right)$$

$$\frac{d^2}{dz^2}\log(\sigma(z) = -\frac{1}{z^2} + \sum_{\substack{\omega\in\Lambda\\ \omega\neq 0}} \left(-\left(\frac{1}{(\omega - z)^2}\right) - \left(\frac{1}{\omega^2}\right)\right)$$

$$= -\wp(z)$$

Since $\frac{d^2}{dz^2}\log(\sigma)$ is equal to $-\wp$, we see that $\frac{d^2}{dz^2}\log(\sigma)$ is doubly periodic. Thus:

$$\begin{aligned}\int\int \frac{d^2}{dz^2}\log(\sigma(z+\omega))dzdz &= \int\int \frac{d^2}{dz^2}\log(\sigma(z))dzdz \\ &= \int \frac{d}{dz}\left(\log(\sigma(z)) + a\right)dz \quad \text{for some } a \in \mathbb{R}. \\ &= \log(\sigma(z)) + az + b \quad \text{for some } a, b \in \mathbb{R}.\end{aligned}$$

Finally, if we set both sides of the equation as the exponent of e, simplification gives:

$$\sigma(z+\omega) = \sigma(z)e^{az+b},$$

which completes the proof of (c). □

Theorem 3.8. [2]*Let* $n_1, \ldots, n_r \in \mathbb{Z}$ *and* $z_1, \ldots, z_r \in \mathbb{C}$. *If* $\sum_{i=1}^r n_i = 0$ *and* $\sum_{i=1}^r n_i z_i \in \Lambda$, *then there is a doubly periodic rational function* $f : \mathbb{C}/\Lambda \to \mathbb{C}$ *such that* $\operatorname{div}(f) = \sum_{i=1}^r n_i(z_i)$. *Additionally, if* $\sum_{i=1}^r n_i z_i = 0$, *which we can force to be*

true, then

$$f(z) = \prod_{i=1}^{r} (\sigma(z - z_i))^{n_i}.$$

Proof First, consider that if $\sum_{i=1}^{r} n_i z_i = \lambda \in \Lambda$ such that $\lambda \neq 0$, then define our divisor as $\sum_{i=1}^{r+2} n_i(z_i) = (0) - (\lambda) + \sum_{i=1}^{r} n_i(z_i)$. Now we have $\sum_{i=1}^{r+2} n_i z_i = 0$, so we can define f as in the theorem, except with a different integer r.

Since by Lemma 3.7 (a), $\sigma(0) = 0$, our rational function f has the correct zeros and poles. Then we need to show f is doubly periodic, thus $f(z+\omega) = f(z)$ for any $\omega \in \Lambda$. To do this, consider

$$\begin{aligned}
\frac{f(z+\omega)}{f(z)} &= \frac{\prod_{i=1}^{r} (\sigma(z+\omega-z_i))^{n_i}}{\prod_{i=1}^{r} (\sigma(z-z_i))^{n_i}} \\
&= \frac{\prod_{i=1}^{r} (\sigma(z-z_i)\exp(a(z-z_i)+b))^{n_i}}{\prod_{i=1}^{r} (\sigma(z-z_i))^{n_i}} \qquad \text{by Lemma 3.7 (c)} \\
&= \frac{\prod_{i=1}^{r} (\sigma(z-z_i))^{n_i} \exp((a(z-z_i)+b)n_i)}{\prod_{i=1}^{r} (\sigma(z-z_i))^{n_i}} \\
&= \prod_{i=1}^{r} \exp((a(z-z_i)+b)n_i) \\
&= \exp\left(\sum_{i=1}^{r} (a(z-z_i)+b)n_i\right) \\
&= \exp\left(az\sum_{i=1}^{r} n_i - a\sum_{i=1}^{r} n_i z_i + b\sum_{i=1}^{r} n_i\right) \\
&= e^0 = 1
\end{aligned}$$

Thus, $f(z+\omega) = f(z)$, so f is doubly periodic. □

Proof of Theorem 3.3 All results are proven in $\mathbb{C}/\Lambda$, then transferred to E with the

isomorphism provided by the Weierstrass $\wp$-function.

The proof of Theorem 3.8 proves (a).

To prove (b), consider $f = c$, for some constant $c \in \mathbb{C}$. For $c \neq 0$, there are clearly no poles or zeros, and thus the divisor is 0. If $f = 0$, consider that $g = c$ is simply a translation of f, but has no poles or zeros.

Considering a function f with $\text{div}(f) = 0$, there are no poles or zeros. Thus by Picard's Little Theorem, since there are many complex points not in the image of f, f is a constant function.

For (c), we first show a general fact for elliptic functions f and f'.

$$f = \frac{(\sigma(z-\alpha_1))^{a_1} \dots (\sigma(z-\alpha_p))^{a_p}}{(\sigma(z-\beta_1))^{b_1} \dots (\sigma(z-\beta_q))^{b_q}}, \quad f' = \frac{(\sigma(z-\gamma_1))^{c_1} \dots (\sigma(z-\gamma_p))^{c_r}}{(\sigma(z-\delta_1))^{d_1} \dots (\sigma(z-\delta_q))^{d_s}},$$

which gives us

$$\text{div}(f) = \sum_{i=1}^{p} \alpha_i - \sum_{j=1}^{q} \beta_j, \quad \text{div}(f') = \sum_{i=1}^{r} \gamma_i - \sum_{j=1}^{s} \delta_j.$$

Then

$$\frac{f}{f'} = \frac{(\sigma(z-\alpha_1))^{a_1} \dots (\sigma(z-\alpha_p))^{a_p} (\sigma(z-\delta_1))^{d_1} \dots (\sigma(z-\delta_q))^{d_s}}{(\sigma(z-\beta_1))^{b_1} \dots (\sigma(z-\beta_q))^{b_q} (\sigma(z-\gamma_1))^{c_1} \dots (\sigma(z-\gamma_p))^{c_r}}$$

From this, we can see that

$$\text{div}\left(\frac{f}{f'}\right) = \text{div}(f) - \text{div}(f'). \tag{4}$$

Assuming $f = cf'$ for some constant c, $\operatorname{div}\left(\frac{f}{f'}\right) = \operatorname{div}(c) = 0$, by Part (b). Thus, by equation (6), $\operatorname{div}(f) = \operatorname{div}(f')$.

Assuming $\operatorname{div}(f) = \operatorname{div}(f')$, by equation (6), $\operatorname{div}\left(\frac{f}{f'}\right) = 0$. Thus $\frac{f}{f'} = c$ for some constant c, by Part (b). Then $f = cf'$. □

The divisor of a function $\operatorname{div}(f)$ tells us important facts about the evaluation of f at certain points, but what if we wanted to evaluate another function g at $\operatorname{div}(f)$? This leads to an important theorem about divisors: Weil reciprocity, which will be helpful in later proofs.

Definition 3.9. For functions f, g with $\operatorname{div}(g) = \sum n_P(P)$, the evaluation of f at the divisor of g is defined as:

$$f\big(\operatorname{div}(g)\big) = \prod f(P)^{n_P}.$$

Example 3.10. We look at an example of single variable real-valued functions for simplicity. For $X \in \mathbb{R}$, define

$$f(X) = \frac{(X-1)^3(X-3)^2}{(X-4)^2(X+2)^2} \text{ and } g(X) = \frac{(X-2)^3(X+3)^4}{X^2(X+1)}.$$

Then clearly,

$$\operatorname{div}(f) = 3(1) + 2(3) - 2(4) - 2(-2)$$

and

$$\operatorname{div}(g) = 3(2) + 4(-3) - 2(0) - (-1)$$

By definition,

$$\begin{aligned} f\big(\operatorname{div}(g)\big) &= \frac{f(2)^3 f(-3)^4}{f(0)^2 f(-1)} \\ &= \frac{\left(\frac{1}{64}\right)^3 \left(-\frac{2304}{49}\right)^4}{\left(-\frac{9}{64}\right)^2 \left(-\frac{128}{25}\right)} \\ &= -\frac{1061683200}{5764801} \end{aligned}$$

and

$$\begin{aligned} g\big(\operatorname{div}(f)\big) &= \frac{g(1)^3 g(3)^2}{g(4)^2 g(-2)^2} \\ &= \frac{(-128)^3 (36)^2}{\left(\frac{2401}{10}\right)^2 (16)^2} \\ &= -\frac{1061683200}{5764801} \end{aligned}$$

We notice above that $f\big(\operatorname{div}(g)\big) = g\big(\operatorname{div}(f)\big)$. The following theorem states that this is generally true, an important fact for later proofs.

Theorem 3.11. [2] *Weil Reciprocity*

For functions f, g mapping curve C into the complex numbers,

$$f\big(\operatorname{div}(g)\big) = g\big(\operatorname{div}(f)\big).$$

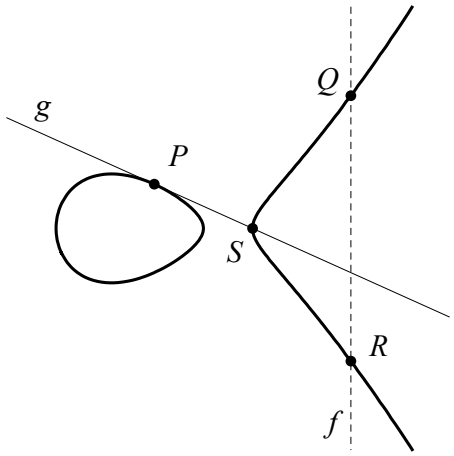

Figure 5: $E : Y^2 = X^3 - 2X^2 - 3X$, with our selected functions and their zeros.

Example 3.12. Looking at $Y^2 = X^3 + 2X^2 - 3X$, (which could easily be put into Weierstrass form, if desired) we choose four points $P = (-1, 2), Q = (3, 6), R = (3, -6)$ and $S = 1, 0)$. Two simple functions on E are the lines $f : X - 3 = 0$ and $g : Y + X - 1$. By viewing the graph of these in Figure 5. we see that f has roots at Q, R, and $\mathcal{O}$, since all vertical lines intersect $\mathcal{O}$. We also see that g has a root at S and a double root at P, since it is tangent at P. Thus

$$\operatorname{div}(f) = (Q) + (R) + (\mathcal{O}) \text{ and } \operatorname{div}(g) = 2(S) + (P).$$

Importantly, when we wish to evaluate,say, $f\big(\operatorname{div}(g)\big)$, we need to work in our homogeneous forms: $f = X - 3Z$, $g = Y + X - Z$. So, by definition:

$$\begin{aligned} f\big(\mathrm{div}(g)\big) &= f(S)^2 f(P) \\ &= (-4)^2(-2) \\ &= -32 \end{aligned}$$

and

$$\begin{aligned} g\big(\mathrm{div}(f)\big) &= g(Q)g(R)g(\mathcal{O}) \\ &= (8)(-4)(1) \\ &= -32 \end{aligned}$$

4 The Weil pairing on points of an elliptic curve

We quickly review the defining qualities of a bilinear pairing before moving on to our definition of the Weil pairing.

Definition 4.1. For abelian groups G, H a *bilinear pairing* is a map $b : G \times G \to H$ with the qualities:

$$b(g_1 + g_2, g_3) = b(g_1, g_3) + b(g_2, g_3)$$

$$b(g_1, g_2 + g_3) = b(g_1, g_2) + b(g_1, g_3)$$

for all $g_1, g_2, g_3 \in G$.

Examples of common bilinear pairings include the dot product $\bullet : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$,

the cross product $\times : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$, and the map from $\mathbb{R}^2 \times \mathbb{R}^2$ to $\mathbb{R}$ defined as:

$$\left(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} c \\ d \end{bmatrix} \right) \mapsto \det \begin{bmatrix} a & c \\ b & d \end{bmatrix} = ad - bc.$$

It can be easily shown that each of these satisfy the conditions of bilinearity.

Through the Weierstrass $\wp$-function, there is a clear relationship between $E[m]$ and a vector space with m^2 elements. On $\mathbb{C}/\Lambda$, the set

$$\left\{ \frac{i\omega_1}{m} + \frac{j\omega_2}{m}, \;\; 0 \leq i \leq m-1, \;\; 0 \leq j \leq m-1 \right\}$$

is mapped directly into $E[m]$. These points are easily verified as of order m, since multiplying $\dfrac{i\omega_1}{m}$ and $\dfrac{j\omega_2}{m}$ by m clearly gives a multiple of ω_1 and ω_2, respectively. There are m^2 elements in the set, and thus there are m^2 elements in $E[m]$. In fact, this also illustrates that $E[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$.

Now, just as the determinant of a matrix can be used to show whether or not a number of vectors are linearly independent of each other, it would be advantageous to determine whether or not $P, Q \in E[m]$ are both constant multiples of each other or some other point $R \in E[m]$. This was one of the original motivations of developing the Weil pairing long before it was used in cryptography.

Now we give our definition of the Weil pairing.

Definition 4.2. For $P, Q \in E[m]$, let f_P, f_Q be rational functions on E such that

$$\operatorname{div}(f_P) = m(P) - m(\mathcal{O}) \;\text{ and }\; \operatorname{div}(f_Q) = m(Q) - m(\mathcal{O}).$$

The *Weil pairing* of P and Q is defined as

$$e_m(P,Q) = \frac{f_P(Q+S)}{f_P(S)} \Big/ \frac{f_Q(P-S)}{f_Q(-S)} ,$$

where $S \notin \{\mathcal{O}, P, -Q, P-Q\}$ to ensure the pairing is defined and nonzero.

Theorem 4.3. [1] *The Weil pairing has the following qualities:*

(a) $e_m(P,Q)$ *is independent of choice of the functions and the point* S.

(b) *The value of the Weil pairing is an m-th root of unity, that is:* $e_m(P,Q)^m$=1.

(c) *The Weil pairing is bilinear in a multiplicative manner: for all* $P_1, P_2, Q \in E[m]$,

$$e_m(P_1+P_2,Q) = e_m(P_1,Q)e_m(P_2,Q) \ \textit{and}$$

$$e_m(Q,P_1+P_2) = e_m(Q,P_1)e_m(Q,P_2).$$

(d) *The Weil pairing is alternating, that is,*

$$e_m(P,P) = 1 \ \textit{for all } P \in E[m].$$

This implies

$$e_m(P,Q) = e_m(Q,P)^{-1} \ \textit{for all } P,Q \in E[m].$$

(e) *The Weil pairing is non-degenerate:*

$$\textit{if } e_m(P,Q) = 1 \ \textit{for all } Q \in E[m], \ \textit{then } P = \mathcal{O}$$

Proof

(a) To show that $e_m(P,Q)$ is independent of choice of functions, consider that in the definition, for f_P to be used, it must satisfy $\text{div}(f_P) = m(P) - m(\mathcal{O})$. Then if we use function f'_P, where $f'_P \neq f_P$, we must have $\text{div}(f'_P) = m(P) - m(\mathcal{O})$. Then by Theorem 3.3 (c), for some constant c,

$$cf_P = f'_P.$$

Then, if we denote the Weil pairing acquired from using f'_p as $e'_m(P,Q)$, we see that

$$\begin{aligned} e'_m(P,Q) &= \frac{f_{P'}(Q+S)}{f_{P'}(S)} \Big/ \frac{f_Q(P-S)}{f_Q(-S)} \\ &= \frac{cf_P(Q+S)}{cf_P(S)} \Big/ \frac{f_Q(P-S)}{f_Q(-S)} \\ &= \frac{f_P(Q+S)}{f_P(S)} \Big/ \frac{f_Q(P-S)}{f_Q(-S)} \\ &= e_m(P,Q). \end{aligned}$$

Clearly, the same is true for choice of the function f_Q, thus the Weil pairing is independent of choice of functions.

Consider $F(S) : E(\mathbb{C}) \to \mathbb{C}$, a function defined as

$$F(S) = \frac{f_P(Q+S)}{f_P(S)} \Big/ \frac{f_Q(P-S)}{f_Q(-S)}\ .$$

We show that $F(S)$ has no zeros or poles, which is equivalent, by Picard's Little Theorem, to showing that $F(S)$ is constant. By the divisors of f_P and f_Q, the possible zeros would occur when

(1) $Q + S = P$, which implies $S = P - Q$.

(2) $-S = Q$, which implies $S = -Q$.

(3) $S = \mathcal{O}$.

(4) $P - S = \mathcal{O}$, which implies $S = P$.

Checking case (1), $S = P - Q$,

$$F(P-Q) = \frac{f_P(P))}{f_P(P-Q)} \Big/ \frac{f_Q(Q)}{f_Q(Q-P)} \, .$$

By the divisors of f_P and f_Q, $f_P(P) = f_Q(Q) = 0$, which gives a removable singularity, and since both $f_P(P-Q)$ and $f_Q(Q-P)$ are nonzero, $F(P-Q) \neq 0$. Checking cases (2)-(4) gives the same result. Then $e_m(P, Q)$ is independent of choice of S.

(b) Consider the numerator of the Weil pairing raised to the m-th power,

$$\frac{f_P(Q+S)^m}{f_P(S)^m} = f_P(Q+S)^m f_P(S)^{-m}$$

Note that this product is of the form of f_P evaluated at a divisor:

$$= f_P\big(m(Q+S) - m(S)\big)$$

Next note that $m(Q+S) - m(S)$ is the divisor of $f_Q(X - S)$. Thus, using Weil

reciprocity,

$$f_P\Big|_{\mathrm{div}(f_Q(X-S))} = f_Q(X-S)\Big|_{\mathrm{div}(f_P)}$$

Since $\mathrm{div}(f_P) = m(P) - m(\mathcal{O})$, we have

$$\begin{aligned}\frac{f_P(Q+S)^m}{f_P(S)^m} &= f_Q\big(m(P-S) - m(-S)\big) \\ &= \frac{f_Q(P-S)^m}{f_Q(-S)^m}\end{aligned}$$

Then the numerator and denominator, when raised to the m-th power, equal each other. Thus $e_m(P,Q)^m = 1$.

(c) We now prove bilinearity in the first term, (the proof in the second term will be identical) that is, we show:

$$e_m(P_1+P_2,Q) = e_m(P_1,Q)e_m(P_2,Q)$$

$$\begin{aligned}\frac{f_{P_1+P_2}(Q+S)}{f_{P_1+P_2}(S)}\Big/\frac{f_Q(P_1+P_2-S)}{f_Q(-S)} &= \left(\frac{f_{P_1}(Q+S)}{f_{P_1}(S)}\Big/\frac{f_Q(P_1-S)}{f_Q(-S)}\right)\left(\frac{f_{P_2}(Q+S)}{f_{P_2}(S)}\Big/\frac{f_Q(P_2-S)}{f_Q(-S)}\right) \\ &= \frac{f_{P_1}(Q+S)f_{P_2}(Q+S)}{f_{P_1}(S)f_{P_2}(S)}\Big/\frac{f_Q(P_1-S)f_Q(P_2-S)}{f_Q(-S)f_Q(-S)}\end{aligned}$$

Rearranging gives:

$$\frac{f_{P_1+P_2}(Q+S)}{f_{P_1}(Q+S)f_{P_2}(Q+S)}\Big/\frac{f_{P_1+P_2}(S)}{f_{P_1}(S)f_{P_2}(S)} = \frac{f_Q(P_1+P_2-S)f_Q(-S)}{f_Q(P_1-S)f_Q(P_2-S)} \qquad (5)$$

Define a new function, F_{P_1,P_2} as:

$$F_{P_1,P_2}(X) = \frac{f_{P_1+P_2}(X)}{f_{P_1}(X)f_{P_2}(X)}.$$

Then

$$\begin{aligned}\operatorname{div}(F_{P_1,P_2}) &= m(P_1+P_2) - m(\mathcal{O}) - \big(m(P_1) - m(\mathcal{O})\big) - \big(m(P_2) - m(\mathcal{O})\big) \\ &= m\big((P_1+P_2) - (P_1) - (P_2) + (\mathcal{O})\big) \\ &= m\operatorname{div}(G_{P_1,P_2})\end{aligned}$$

where G_{P_1,P_2} is some function such that $\operatorname{div}(G_{P_1,P_2}) = (P_1+P_2) - (P_1) - (P_2) + (\mathcal{O})$. Thus $F_{P_1,P_2}(X) = (G_{P_1,P_2}(X))^m$.

Now, the left side of (5) becomes

$$\begin{aligned}\frac{f_{P_1+P_2}(Q+S)}{f_{P_1}(Q+S)f_{P_2}(Q+S)} \Big/ \frac{f_{P_1+P_2}(S)}{f_{P_1}(S)f_{P_2}(S)} &= \frac{F_{P_1,P_2}(Q+S)}{F_{P_1,P_2}(S)} \\ &= \frac{G_{P_1,P_2}(Q+S)^m}{G_{P_1,P_2}(S)^m}\end{aligned}$$

Note that this product is of the form of G_{P_1,P_2} evaluated at a divisor, and that this is the divisor of $f_Q(X-S)$.

$$\begin{aligned}&= G_{P_1,P_2}\big(m(Q+S) - m(S)\big) \\ &= G_{P_1,P_2}\big(\operatorname{div}(f_Q(X-S))\big)\end{aligned}$$

Using Weil reciprocity and evaluating $f_Q(X-S)$ at $\operatorname{div}(G_{P_1,P_2}) = (P_1+P_2) - (P_1) - (P_2) + (\mathcal{O})$ gives:

$$= \frac{f_Q(P_1+P_2-S)f_Q(-S)}{f_Q(P_1-S)f_Q(P_2-S)}$$

This is exactly the right hand side of (5), and thus

$$e_m(P_1 + P_2, Q) = e_m(P_1, Q)e_m(P_2, Q).$$

(d) To prove that $e_m(P, P) = 1$, consider the definition:

$$e_m(P, P) = \frac{f_P(P + S)}{f_P(S)} \Big/ \frac{f_P(P - S)}{f_P(-S)}$$

Now, by part (a), we may choose any value of S, so we will let S converge to $\mathcal{O}$. Then we have:

$$e_m(P, P) = \frac{f_P(P)}{f_P(\mathcal{O})} \Big/ \frac{f_P(P)}{f_P(\mathcal{O})} = 1.$$

Now, using bilinearity from (c), we have

$$e_m(P + Q, P + Q) = e_m(P, P)e_m(P, Q)e_m(Q, P)e_m(Q, Q).$$

Then since $e_m(X, X) = 1$, we have

$$1 = e_m(P, Q)e_m(Q, P).$$

(e) Consider $P \in E[m]$ such that $e_m(P, Q) = 1$ for all $Q \in E[m]$. Then if

$$e_m(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \Big/ \frac{f_Q(P - S)}{f_Q(-S)},$$

we can consider two divisors (not necessarily of functions),

$$D_P = (P - S) - (-S) \text{ and } D_Q = (Q + S) - (S).$$

Then

$$e_m(P, Q) = f_P(D_Q)/f_Q(D_P)$$

Then if $e_m(P, Q) = 1$, substituting backwards gives:

$$\begin{aligned} f_P(D_Q) &= f_Q(D_P) \\ \frac{f_P(Q + S)}{f_P(S)} &= f_Q(D_P) \\ f_P(Q + S) &= f_Q(D_P) f_P(S). \end{aligned} \tag{6}$$

Since the Weil pairing is independent of choice of S, we can replace S with $Q + S$.

$$f_P([2]Q + S) = f_Q(D_P) f_P(Q + S)$$

replacing $f_P(Q + S)$ by (6), and repeating m times:

$$f_P([2]Q + S) = f_Q(D_P)^2 f_P(S)$$

$$\vdots$$

$$f_P([m]Q + S) = f_Q(D_P)^m f_P(S)$$

But $[m]Q = \mathcal{O}$, thus

$$f_P(S) = f_Q(D_P)^m f_P(S)$$

and so, $f_Q(D_P)^m = 1$.

Raising equation (6) to the m-th power,

$$f_P(Q+S)^m = f_Q(D_P)^m f_P(S)^m$$

and

$$f_P(Q+S)^m = f_P(S)^m$$

for all $S \in E$ and $Q \in E[m]$.

Since ${f_P}^m$ is unchanged by a translation by Q, [2] tells us that $f_P(S)^m = (h \circ [m])(S)$ for some function h, where $[m]$ is the map of adding P to itself m times to result in $[m]P$. As we know, $\mathrm{div}(f_P) = m(P) - m(\mathcal{O})$, and thus $\mathrm{div}(h \circ [m]) = \mathrm{div}({f_P}^m) = m^2(P) - m^2(\mathcal{O})$.

If h has a zero, say X, such that $h(X) = 0$, then $h \circ [m]$ will have zeros at the points $\{X' + Q : Q \in E[m]\}$ where $mX' = X$. Then if h has a zero, $h \circ [m]$ will have m^2 distinct zeros. But by its divisor, ${f_P}^m$ only has one zero, P. Thus h has no zeros, so it must be constant. Then

$$\mathrm{div}(h \circ [m]) = m^2(P) - m^2(\mathcal{O}) = 0,$$

which will imply $(P) = (\mathcal{O})$, further implying $P = \mathcal{O}$. □

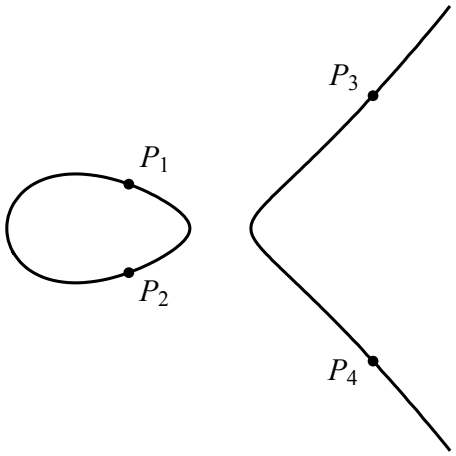

Figure 6: Four "nice" points of order 4 on $E : Y^2 = X^3 - 2X^2 - 3X$

Example 4.4. Looking at $Y^2 = X^3 + 2X^2 - 3X$, pictured in Figure 6, four easy-to-work-with points of order 4 are $P_1 = (-1, 2), P_2 = (-1, -2), P_3 = (3, 6)$, and $P_4 = (3, -6)$. We will also use $Q = (1, 0)$, $S = (-2, \sqrt{6})$, and of course $\mathcal{O}$. To calculate Weil pairings among these four points,we will need the correct functions. Taking P_1 as an example, we need a function f_{P_1} with $\text{div}(f_{P_1}) = 4(P_1) - 4(\mathcal{O})$. Without any better algorithm, we would be forced to look at lines drawn through our points of interest and use the properties of divisors to create a function. Listed below are the functions used to find f_{P_i} for $i = 1, 2, 3, 4$.

Function	Divisor
$g_1 : \quad X - 3 = 0$	$\text{div}(g_1) = (P_3) + (P_4) + (\mathcal{O})$
$g_2 : Y + 3X - 3 = 0$	$\text{div}(g_2) = 2(P_4) + (Q)$
$g_3 : Y - 3X + 3 = 0$	$\text{div}(g_3) = 2(P_3) + (Q)$
$g_4 : \quad Y + X - 1 = 0$	$\text{div}(g_4) = 2(P_1) + (Q)$
$g_5 : \quad Y - X + 1 = 0$	$\text{div}(g_5) = 2(P_2) + (P_3)$
$g_6 : \quad Z = 0$	$\text{div}(g_6) = 3(\mathcal{O})$

Now, we use the divisors above to create a set of coefficients c_k such that

$$4(P_1) - 4(\mathcal{O}) = \sum_{k=1}^{6} c_k \text{div}(g_k).$$

These coefficients and functions will give

$$f_{P_1} = \prod_{k=1}^{6} {g_k}^{c_k}.$$

Solving for the coefficients, $c_1 = 2, c_2 = -1, c_3 = 0, c_4 = 2, c_5 = 0, c_6 = -2$ and thus, in homogenized form, we end up with:

$$f_{P_1}(X, Y, Z) = \frac{(X - 3Z)^2(Y + X - Z)^2}{(Y + 3X - 3Z)(Y - 3X + 3Z)Z^2}.$$

Similarly, after solving:

$$f_{P_2}(X, Y, Z) = \frac{(X - 3Z)^2(Y - X + Z)^2}{(Y + 3X - 3Z)(Y - 3X + 3Z)Z^2}$$
$$f_{P_3}(X, Y, Z) = \frac{(X - 3Z)^2(Y - 3X + 3Z)}{(Y + 3X - 3Z)Z^2}$$
$$f_{P_4}(X, Y, Z) = \frac{(X - 3Z)^2(Y + 3X - 3Z)}{(Y - 3X + 3Z)Z^2}.$$

Then to calculate the pairing of P_1 and P_3,

$$e_4(P_1, P_3) = \frac{f_{P_1}(P_3 + S)}{f_{P_1}(S)} \Big/ \frac{f_{P_3}(P_1 - S)}{f_{P_3}(-S)} = -1$$

after using the addition algorithm to find $P_3 + S = (-2.496, -2.047), P_1 - S =$

e_4	P_1	P_2	P_3	P_4
P_1	1	1	-1	-1
P_2	1	1	-1	-1
P_3	-1	-1	1	1
P_4	-1	-1	1	1

Table 3: Weil pairing values for four "nice" points in $E[4]$ on $Y^2 = X^3 + 2X^2 - 3X$.

$(20.798, -98.990)$ and using f_{P_1}, f_{P_3} as above. Note that

$$e_4(P_3, P_1) = e_4(P_1, P_3)^{-1} = -1.$$

To avoid computing all of these, we use the bilinearity and alternating qualities of e_4: Note that $[4]P_1 = \mathcal{O}$ and $-P_1 = P_2$ implies $P_2 = [3]P_1$. Thus,

$$e_4(P_1, P_2) = e_4(P_1, [3]P_1) = e_4(P_1, P_1)^3 = 1^3 = 1$$

and by the alternating quality,

$$e_4(P_2, P_1) = e_4(P_1, P_2)^{-1} = 1$$

Using this and the qualities of the Weil pairing, we attain all values, recorded in Table 3. It is of note that there are 12 other points in $E[4]$, $\mathcal{O}$ and eleven more. Each of these has complex coordinates, and computing the Weil pairing would have taken us into the complex fourth roots of unity.

5 Miller's algorithm

Although we were able to calculate 16 Weil pairings in Example 4.4 with only one true (by the definition) calculations, it is clear that the process was painstaking, and more importantly, not computer friendly. If a computer is to quickly evaluate a Weil pairing after being given the curve and two points, this system will not work. If a point R not in our four "nice" points was given to us, the process would have to be completely redone. Thankfully, Miller[4], in 1986, gave an algorithm that quickly finds functions with desired divisors.

Theorem 5.1. [1] *For points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ on curve E,*

(a) *Let λ be the slope of the line through P and Q (where $\lambda = \infty$ if the line is vertical), or the slope of the tangent line through P if $P = Q$. Define function $h_{P,Q}$ on E as:*

$$h_{P,Q} = \begin{cases} \dfrac{Y - y_P - \lambda(X - x_P)}{X + x_P + x_Q - \lambda^2} & \text{if } \lambda \neq \infty \\ X - x_P & \text{if } \lambda = \infty \end{cases}$$

Then

$$\mathrm{div}(h_{P,Q}) = (P) + (Q) - (P + Q) - (\mathcal{O}).$$

(b) **(Miller's Algorithm)** *Let $m \geq 1$, with binary expansion*

$$m = m_0 + 2m_1 + 4m_2 + 8m_3 + \cdots + 2^{n-1}m_{n-1} \quad \text{with} \quad m_o, m_1, \ldots, m_{n-1} \in \{0, 1\}.$$

The following algorithm, using $h_{P,Q}$ defined as in part (a), results in a function

f_P with

$$\mathrm{div}(f_P) = m(P) - ([m]P) - (m-1)(\mathcal{O}).$$

1. Set $T = P$ and $f = 1$.
2. Loop $i = n - 2$ down to $i = 0$:
    3. Set $f = f^2 h_{T,T}$.
    4. Set $T = [2]T$.
    5. If $m_i = 1$
        6. Set $f = f h_{T,P}$.
        7. Set $T = T + P$.
    8. End If.
9. End i-loop.
10. Return f.

Then if $P \in E[m]$, $\mathrm{div}(f_P) = m(P) - m(\mathcal{O})$.

Proof To prove that the function $h_{P,Q}$ has the desired divisor, consider the line through P and Q. The line has the form

$$Y = \lambda(X - x_P) + y_P$$

and intercepts E at P, Q and $-(P+Q)$. Then

$$\mathrm{div}(Y - y_P - \lambda(X - x_P)) = (P) + (Q) + (-P - Q) - 3(\mathcal{O}).$$

This is the divisor of the numerator of $h_{P,Q}$. Next note that, by the addition algorithm, $x_{P+Q} = \lambda^2 - x_P - x_Q$. Then the denominator of $h_{P,Q}$ is:

$$X + x_P + x_Q - \lambda^2 = X - x_{P+Q}.$$

Since this is a vertical line, it intercepts E at $P+Q$ and $-P-Q$, and has a pole of duplicity 2 at $\mathcal{O}$. Thus.

$$\operatorname{div}(X - x_{P+Q}) = (P+Q) + (-P-Q) - 2(\mathcal{O}).$$

Then

$$\operatorname{div}(h_{P,Q}) = \operatorname{div}(Y - y_P - \lambda(X - x_P)) - \operatorname{div}(X - x_{P+Q}) = (P) + (Q) - (P+Q) - (\mathcal{O}).$$

Finally, if $\lambda = \infty$, then the line is vertical, and thus $Q = -P$. Then we want

$$\operatorname{div}(h_{P,Q}) = (P) + (Q) + (-P-Q) - (\mathcal{O}) = (P) + (-P) - 2(\mathcal{O}).$$

The function $X - x_P$ has exactly this divisor. This completes the proof of (a).

Part (b) will not be proven in entirety, but the first few cases are shown here:

Let $m = 2 = 0 + 1(2)$, then we have $i = 0$ as our only use of step (2) of the algorithm. Thus we will return f after going through step (3) only once, skipping (5) since $m_0 = 0$, thus giving

$$f = f^2 h_{P,P} = h_{P,P}.$$

Thus,

$$\begin{aligned} \operatorname{div}(f) &= \operatorname{div}(h_{P,P}) \\ &= 2(P) - ([2]P) - (\mathcal{O}) \text{ by (a)} \end{aligned}$$

which is the desired divisor.

Let $m = 3 = 1 + 1(2)$, then again we have $i = 0$ as our only use of step (2) of the algorithm. However, $m_0 = 1$, so we will use step (5) in this case. Thus we return f as

$$f = h_{P,P} h_{[2]P,P}$$

Thus,

$$\begin{aligned}
\mathrm{div}(f) &= \mathrm{div}(h_{P,P}) + \mathrm{div}(h_{[2]P,P}) \\
&= 2(P) - ([2]P) - (\mathcal{O}) + ([2]P) + (P) - ([2]P + P) - (\mathcal{O}) \text{ by (a)} \\
&= 3(P) + ([3]P) - 2(\mathcal{O})
\end{aligned}$$

which is the desired divisor.

If we let $m = 4 = 0 + 0(2) + 1(4)$, we loop through (2) twice. Both times we will skip (5), since $m_1 = m_0 = 0$, thus we return f as

$$f = (h_{P,P})^2 h_{[2]P,[2]P}.$$

Then

$$\begin{aligned}
\mathrm{div}(f) &= 2\mathrm{div}(h_{P,P}) + \mathrm{div}(h_{[2]P,[2]P}) \\
&= 2\Big(2(P) - ([2]P) - (\mathcal{O})\Big) + \Big(([2]P) + [2]P) - ([2]P + [2]P) - (\mathcal{O})\Big) \text{ by (a)} \\
&= 4(P) + ([4]P) - 3(\mathcal{O})
\end{aligned}$$

as desired.

As can be seen, using the relation given by (a) and following the algorithm can lead to an inductive proof. Also, note that for each of these cases, if $P \in E[m]$ for the selected m, the divisor becomes:

$$\operatorname{div}(f) = m(P) - ([m]P) - (m-1)(\mathcal{O}) = m(P) - m(\mathcal{O}). \qquad \square$$

Another important aspect of Miller's algorithm is that it can be used to actually evaluate the function f, simply by evaluating the function at the desired point whenever the function is adjusted (steps (3) and (6).) An example where this is done is contained in the next section, and the Mathematica code is included in Appendix.

6 Elliptic curves over finite fields

In the first section, we worked in $E(\mathbb{R})$, a subset of $E(\mathbb{C})$. As we move to a finite field with prime p elements, $\mathbb{F}_p$, we lose the geometric interpretation of our addition operation. That is, we cannot easily talk about lines in $E(\mathbb{F}_p)$. However, we may still use our explicit formulas of Theorem 1.8 to add two points, as long as we use operations of the field $\mathbb{F}_p$. Thankfully, these additions are simply modular arithmetic.

Example 6.1. Consider the elliptic curve

$$E : Y^2 = X^3 + 4 \text{ over the field } \mathbb{F}_5.$$

Since the field $\mathbb{F}_5$ only contains five elements, it is relatively easy to check whether

$+$	$\mathcal{O}$	$(1,0)$	$(0,2)$	$(0,3)$	$(3,1)$	$(3,4)$
$\mathcal{O}$	$\mathcal{O}$	$(1,0)$	$(0,2)$	$(0,3)$	$(3,1)$	$(3,4)$
$(1,0)$	$(1,0)$	$\mathcal{O}$	$(3,4)$	$(3,1)$	$(0,3)$	$(0,2)$
$(0,2)$	$(0,2)$	$(3,4)$	$(0,3)$	$\mathcal{O}$	$(1,0)$	$(3,1)$
$(0,3)$	$(0,3)$	$(3,1)$	$\mathcal{O}$	$(0,2)$	$(3,4)$	$(1,0)$
$(3,1)$	$(3,1)$	$(0,3)$	$(1,0)$	$3,4)$	$(0,2)$	$\mathcal{O}$
$(3,4)$	$(3,4)$	$(0,2)$	$(3,1)$	$(1,0)$	$\mathcal{O}$	$(0,3)$

Table 4: Addition table for $E : Y^2 = X^3 + 4$ over the field $\mathbb{F}_5$

or not, for a certain $(x, y) \in \mathbb{F}_5 \times \mathbb{F}_5$, $(x, y) \in E(\mathbb{F}_5)$. We just compare the values $y^2 \bmod 5$ and $x^3 + 4 \bmod 5$ for $x, y = \{0, 1, 2, 3, 4\}$. Choosing the values of x and y that equate these quantities, we have

$$E(\mathbb{F}_5) = \{(1,0), (0,2), (0,3), (3,1), (3,4), \mathcal{O}\}.$$

Let $P = (0, 2)$ and $Q = (3, 4)$. Then to compute $P + Q$, we have

$$\lambda = \frac{4-1}{2-0} \bmod 5 = \frac{3}{2} \bmod 5 = 4$$

$$x_3 = 4^2 - 0 - 3 \bmod 5 = 3$$

$$y_3 = 4(0-3) - 2 \bmod 5 = 1$$

Thus, $P + Q = (3, 1)$. Note that $P + Q \in E(\mathbb{F}_5)$. The results of summing every pair of points in $E(\mathbb{F}_5)$ are shown in Table 4.

Since all of the work we have done to this point only required the E to be over

a field, all of the theorems we have proved up to this point hold when working with $\mathbb{F}_p$. Only the work in Section 5 dealing with Weierstrass $\wp$ and σ functions require the structure that $\mathbb{F}_p$ lies within is algebraically closed, which is true, despite being beyond the scope of this paper. All examples from this point forward will take place in $E(\mathbb{F}_p)$, or some similar finite field.

Example 6.2. We now use Miller's algorithm to compute two examples of the Weil pairing on the curve

$$E : Y^2 = X^3 + 37X \text{ over the field } \mathbb{F}_{1009}.$$

It is a matter of verifying that the points

$$P = (8, 703), Q = (49, 20), P' = (417, 952), Q' = (561, 153)$$

are all of order 7 on E. It is also verifiable that $P' = [2]P$ and $Q' = [3]Q$. We use the point $S = (0, 0)$, which is clearly of order 2. Using the mathematica program included in Appendix C, using Miller's algorithm we see that

$$e_7(P, Q) = 105 \text{ and } e_7(P', Q') = 394.$$

Verifying that these are both 7th roots of primitive unity, $105^7 \equiv 1 \bmod 1009$ and $394^7 \equiv 1 \bmod 1009$. Also, verifying other properties of the Weil pairing,

$$e_7(P', Q') = e_7([2]P, [3]Q) = e_7(P, Q)^6 = 105^6 \equiv 394 \bmod 1009.$$

7 The discrete logarithm problem (DLP) and Diffie-Hellman key exchange

Now we detour to view a problem which we will later rephrase using elliptic curves.

Definition 7.1. Let g be a primitive root for $\mathbb{F}_p$, and h be a given nonzero element of $\mathbb{F}_p$. We define the *discrete logarithm problem* (DLP) as: Given g, h as above, find the integer $x \in \{1, 2, \ldots, p-1\}$ such that

$$g^x \equiv h \bmod p.$$

The integer x is called the *discrete logarithm of h to the base g* and is denoted by $\log_g(h)$.

One may note that there if there is a solution to the DLP, then there are infinite solutions. This is due to Fermat's little theorem, which tells us $g^{p-1} \equiv 1 \bmod p$. Then for x that satisfies $g^x \equiv h \bmod p$, we have

$$g^{x+k(p-1)} = g^x (g^{p-1})^k \equiv h(1)^k \equiv h \bmod p$$

for any integer value of k. This is why we define x as lying between 0 and $p-1$.

While seemingly simple, this problem becomes exceedingly difficult dependent on the prime p chosen, which will become the basis for the security of the Diffie-Hellman key exchange. Although some methods, such as index calculus [1], have been developed to solve the DLP faster than brute force, none are quick enough to

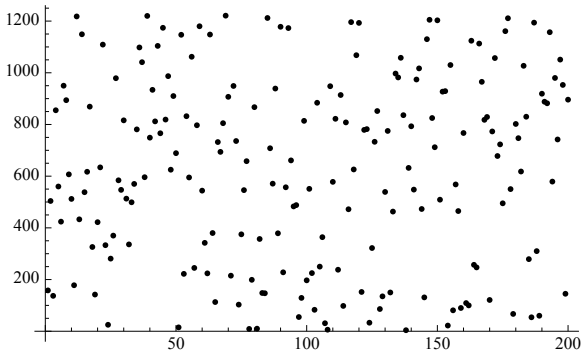

Figure 7: Plot of $158^i \bmod 1223$ for $i = 1, 2, \ldots, 200$

allow fast breaking of any cryptographic system based on the DLP.

Example 7.2. Consider $p = 1223$, $g = 158$. Listing the first 20 powers of g, we have:

i	1	2	3	4	5	6	7	8	9	10
158^i	158	504	137	855	560	424	950	894	607	512
i	11	12	13	14	15	16	17	18	19	20
158^i	178	1218	433	1149	538	617	869	326	142	422

By viewing the above table, there is no clear pattern to the powers of 158 modulo 1223. This is further exemplified by Figure 7, with the powers of up to 200 plotted. As it turns out, the order of 158 in $\mathbb{F}_{1223}$ is 1222. Thus, to solve the DLP in the form of

$$158^x \equiv h \bmod p,$$

a brute force method may require one to compute up to 1221 powers modulo 1223. Even with a relatively small prime number, the DLP can be difficult. In most cryptographical applications, the prime will have hundreds of digits [1], thus creating an exceedingly difficult problem.

Now, consider the case of Alice and Bob, who are trying to share a private key to a cipher. Unfortunately, their only means of communication is insecure, and thus

anything sent is observed by their adversary Eve. While this seems difficult, the Diffie-Hellman key exchange offers a solution.

In the first step, one of Alice or Bob communicates a prime p and a nonzero integer g modulo p. The integer g is best to have a large order, so g is typically a primitive root. That is, the order of g is $p-1$. With this common information, Alice chooses a secret integer a, Bob chooses a secret integer b, and they both raise g to that power. This gives:

$$A \equiv g^a \bmod p \quad \text{and} \quad B \equiv g^b \bmod p.$$

A and B are now communicated between the two, with Eve observing both values. Now, Alice raises B to the power of her secret integer a, and Bob does the same with A to his secret integer b, giving:

$$B' \equiv B^a \bmod p \quad \text{and} \quad A' \equiv A^b \bmod p.$$

These values are the secret key, as they are equal (shown below) and have not been broadcast to Eve.

$$B' \equiv B^a \equiv (g^b)^a \equiv (g^a)^b \equiv A^b \equiv A' \mod p$$

If Eve, knowing A and B, can determine one of the secret integers a and b, she can reconstruct the key. Thus, without loss of generality, given A, g and p, she must be able to find a such that

$$A \equiv g^a \bmod p.$$

Common Parameters Communicated $p = 1223, g = 158$ $\longleftrightarrow$	
Secret Integers	
Alice: $a = 125$	Bob: $b = 421$
First Computations	
$A = 322 \equiv 158^{125} \bmod 1223$	$B = 873 \equiv 158^{421} \bmod 1223$
Communication	
$A = 322$ $\longrightarrow$ $B = 873$ $\longleftarrow$	
Common Key Computation	
$A' = 199 \equiv 873^{125} \bmod 1223$	$B' = 199 \equiv 322^{421} \bmod 1223$

Table 5: Diffie-Hellman key exchange over $\mathbb{F}_{1223}$

In other words, she must solve the discrete logarithm problem.

Example 7.3. Table 5 illustrates a potential Diffie-Hellman key exchange over the field $\mathbb{F}_{1223}$.

Since this example is based on a relatively small field, the solution to the DHP is not unpractical to compute by brute force. Again, any real cryptographical application would use a much larger prime.

8 The elliptic curve discrete logarithm problem (ECDLP) and elliptic curve Diffie- Hellman key exchange

We now define an analogue to the DLP, defined on the additive group $E(\mathbb{F}_p)$.

Definition 8.1. Let P and Q be points in $E(\mathbb{F}_p)$ such that $Q = [n]P$ for some integer n. The *elliptic curve discrete logarithm problem* is the problem of determining the integer $n \in \{1, 2, \ldots, p-1\}$ from given points P and Q. Then n is the *elliptic curve discrete logarithm of Q with respect to P*, denoted

$$n = \log_P Q.$$

As with the DLP, if there exists one integer n that satisfies $P = [n]Q$, there exists an infinite amount of such integers, due to the fact that $E(\mathbb{F}_p)$ is a finite group. This is why we define the solution to the ECDLP as being between 0 and $p-1$.

Another potential issue with the ECDLP is that its solution is not necessarily defined. That is, there exist $P, Q \in E(\mathbb{F}_p)$ such that $Q \neq [n]P$ for any integer n. However, in practical applications, this will not interfere: in the Diffie-Helman key exchange, Alice, Bob, (and Eve) all know the point P from the first communication of parameters. Then both Alice and Bob's points are multiples of P, thus guaranteeing a solution.

Now, considering that the DLP is inefficient to solve, it is logical that the ECDLP

would be as hard or harder to solve. In fact, despite the existence of slightly faster methods to solving the DLP, no method exists for solving the ECDLP that is faster than brute force [1]. Also, thanks to the Double-and-Add algorithm for computing $[n]P$, we can compute $[n]P$ relatively quickly, while the difficulty remains for Alice to find n.

Example 8.2. Now, we create an example Diffie-Helman key exchange, using $E(\mathbb{F}_{1223})$ instead of $\mathbb{F}_{1223}$. Our common point is $P = (583, 599)$ on the curve $Y^2 = X^3 + 3X + 1$ over $\mathbb{F}_{1223}$. Note that Alice and Bob only need to share the x-value of the points they communicate to each other. This is due to the fact that each x-value corresponds to two y-values, since the y is squared in the equation for E (if $y = 0$, there is clearly only one y-value.) This duplicity would only serve to complicate matters, so the x-value is the only number transmitted.

The example shows that choosing the wrong y-value will not disturb the process: Bob sent the value $B = 447$, taken from $[421]P = (447, 94)$. Alice chooses the point $(447, 1129) = -(447, 94)$, yet ends up with the same key as Bob.

9 Modified Weil pairings and the tripartite Diffie-Helman key exchange

We now review a key exchange analogous to Diffie-Helman, except with three members trying to share a private key. We will show that this fails if we use the same method as above. There are a few restraints which we would wish to place on the process. If

Common Parameters Communicated $\xleftrightarrow{p = 1223, \;\; Y^2 = X^3 + 3X + 1, \;\; P = (583, 599)}$	
Secret Integers	
Alice: $a = 125$	Bob: $b = 421$
First Computations	
$A = x\{[125]P\} = x\{(994, 372)\} = 994$	$B = x\{[421]P\} = x\{(447, 94)\} = 447$
Communication	
$\xrightarrow{A = 994}$ $\xleftarrow{B = 447}$	
Common Key Computations	
$A' = x\{[125](447, 1129)\} = x\{(835, 304)\} = 835$	$B' = x\{[421](994, 372)\} = x\{(835, 919)\} = 835$

Table 6: Diffie-Hellman key exchange over $E(\mathbb{F}_{1223})$

we wanted to, we could have two of the parties share a key with Diffie-Helman, then the other two share a key, and communicate through each other. This works, but is cumbersome. We also do not want to require more than one round of communications, as this would require all parties to be online at once, making the process less practical. The solution to this problem, given by Antione Joux in the early 2000's [5], is to use bilinear pairings (in our case the Weil pairing.)

Example 9.1. Alice, Bob, and Carlos would all like to share a private key. As above, any information sent between them before forming this key is suspect to eavesdropping by Eve. If they try to use a point P on elliptic curve over a finite field $E(\mathbb{F}_p)$, each with their own private integers, the first round of calculations would look like:

$$A = [a]P, \quad B = [b]P, \quad C = [c]P.$$

Without loss of generality, let's put ourselves in Carlos' shoes. After these points are published, Carlos knows $A = [a]P, B = [b]P$ and his secret integer c. In order to have the common key, he would need to calculate $[abc]P$. Since he knows c, he is basically computing $[c]([ab]P)$. Thus, he needs to glean $[ab]P$ from knowing $[a]P$ and $[b]P$. This is exactly the problem that Eve faced in the two person E.C. Diffie-Helman key exchange. That is, in order for Alice, Bob, and Carlos to share a common key, they each need to be able to solve the Elliptic Curve Diffie-Helman problem efficiently. This clearly does not work.

We can create a modified version of the Weil pairing in order to solve this issue. Generally, the practice only consists of a prime m, an elliptic curve E, and two points $P_1, P_2 \in E$ where $P_1 = [a]P$, $P_2 = [b]P$ for some integers a, b. This is where we see a limitation of the original Weil pairing, as

$$e_m(P_1, P_2) = e_m([a]P, [b]P) = e_m(P, P)^{ab} = 1^{ab} = 1.$$

This will always give a trivial result. Thus we would like a pairing such that $\hat{e}_m(P, P) \neq 1$.

This new pairing, $\hat{e}_m$ is defined as

$$\hat{e}_m(P, Q) = e_m(P, \phi(Q))$$

where ϕ is a distortion map on E.

Definition 9.2. Let $m \geq 3$ be a prime, E an elliptic curve, and P be a point in

$E[m]$. The map $\phi : E \to E$ is an *m-distortion map* on E if:

(a) $\phi([n]P) = [n]\phi(P)$ for all integer $n \geq 1$.

(b) $e_m(P, \phi(Q))$ is a primitive m-th root of unity, that is, if $e_m(P, \phi(Q))^r = 1$, then $m \mid r$.

Proposition 9.3. [1] *Let $m \geq 3$ be a prime, E an elliptic curve, P be a point in $E[m]$, and $\phi : E \to E$ be an* m-distortion map *on E. If Q, Q' are multiples of P, then*

$$\hat{e}_m(Q, Q') = 1 \text{ if and only if } Q = \mathcal{O} \text{ or } Q' = \mathcal{O}.$$

Proof Assume, for integers a, b, that $Q = [a]P$ and $Q' = [b]P$. Then

$$\begin{aligned}\hat{e}_m(Q, Q') &= \hat{e}_m([a]P, [b]P) = e_m([a]P, \phi([b]P)) = e_m([a]P, [b]\phi(P)) \\ &= e_m(P, \phi(P))^{ab}\end{aligned}$$

Then if $\hat{e}_m(Q, Q') = 1$, $e_m(P, \phi(P))^{ab} = 1$. Thus by the second part of Definition 9.4, $m \mid ab$. Thus $m \mid a$ or $m \mid b$, and so $Q = \mathcal{O}$ or $Q' = \mathcal{O}$. □

Then by Proposition 7.2, we only need a distortion map to create a modified Weil pairing which will work in our Tripartite Diffie-Helman Key Exchange. There are several common examples of distortion maps, some of which can be found in [1] and [5]. In application, of course, the specified field would be much larger than in the example created here. Examples of the primes used, many are 200 to 300 digits long, can be found in [6].

Proposition 9.4. *Let E be the curve $Y^2 = X^3 + 1$ over the field K, and let β be a primitive cube root of unity in K. That is, $\beta \neq 1$, with $\beta^3 = 1$. Define a map ϕ on E as $\phi(P) = \phi(x, y) = (\beta x, y)$ and $\phi(\mathcal{O}) = \mathcal{O}$. Then:*

(a) *For $P \in E(K)$, $\phi(P) \in E(K)$.*

(b) *ϕ respects the addition law on E:*

$$\phi(P_1) + \phi(P_2) = \phi(P_1 + P_2) \text{ for all } P_1, P_2 \in E.$$

Proof Using the assumptions above,

(a) If $P = (x, y) \in E(K)$, then $y^2 = x^3 + 1$. If $\phi(P) = (\beta x, y)$, this still holds true: $y^2 = (\beta x)^3 + 1 = \beta^3 x^3 + 1 = x^3 + 1$.

(b) Now, considering $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E$, we have $\phi(P_1) = (\beta x_1, y_1), \phi(P_2) = (\beta x_2, y_2)$. We look at three cases:

(i) $P_2 = -P_1$, that is: $P_1 + P_2 = \mathcal{O}$. Clearly, $\phi(P_1 + P_2) = \phi(\mathcal{O}) = \mathcal{O}$.

If $P_2 = -P_1$, then $x_1 = x_2$ and $y_1 = -y_2$. Then

$$\phi(P_1) = (\beta x_1, y_1) = (\beta x_2, -y_2) = -\phi(P_2).$$

Thus $\phi(P_1) + \phi(P_2) = \mathcal{O}$, and the equality is true.

(ii) If $P_1 = P_2$, then using the addition algorithm,

$$P_1 + P_2 = \left(\lambda^2 - 2x_1, \lambda(x_1 - \lambda^2 + 2x_1) - y_1\right)$$

and

$$\phi(P_1 + P_2) = \big(\beta(\lambda^2 - 2x_1), \lambda(x_1 - \lambda^2 + 2x_1) - y_1\big)$$

where $\lambda = \dfrac{3{x_1}^2}{2y_1}$.

Again using the addition algorithm, for $\phi(P_1) + \phi(P_2)$, we will have a new slope, $\lambda' = \lambda = \dfrac{3\beta{x_1}^2}{2y_1} = \beta^2\lambda$. Then we have:

$$\begin{aligned}
\phi(P_1) + \phi(P_2) &= \big((\beta^2\lambda)^2 - 2\beta x_1, \beta^2\lambda(\beta x_1 - (\beta^2\lambda)^2 + 2\beta x_1) - y_1\big) \\
&= \big(\beta^4(\lambda)^2 - 2\beta x_1, \lambda(\beta^3 x_1 - \beta^6\lambda^2 + 2\beta^3 x_1) - y_1\big) \\
&= \big(\beta(\lambda^2 - 2x_1), \lambda(x_1 - \lambda^2 + 2x_1) - y_1\big) \\
&= \phi(P_1 + P_2)
\end{aligned}$$

(iii) Finally, if $P_1 \neq P_2$, by the addition algorithm,

$$P_1 + P_2 = \big(\lambda^2 - x_1 - x_2, \lambda(2x_1 - \lambda^2 + x_2) - y_1\big)$$

and

$$\phi(P_1 + P_2) = \big(\beta(\lambda^2 - x_1 - x_2), \lambda(2x_1 - \lambda^2 + x_2) - y_1\big)$$

where $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$.

Also, for $\phi(P_1) + \phi(P_2)$, our slope will be $\lambda' = \dfrac{y_2 - y_1}{\beta(x_2 - x_1)} = \dfrac{\lambda}{\beta}$, so:

$$\phi(P_1) + \phi(P_2) = \left(\left(\frac{\lambda}{\beta}\right)^2 - \beta x_1 - \beta x_2, \frac{\lambda}{\beta}(2\beta x_1 - \left(\frac{\lambda}{\beta}\right)^2 + 2\beta x_2) - y_1\right)$$

interjecting a $b^3 = 1$ into the x-coordinate,

$$\begin{aligned}
&= \left(\beta^3\left(\frac{\lambda^2}{\beta^2}\right) - \beta x_1 - \beta x_2, \lambda(2x_1 - \left(\frac{\lambda^2}{\beta^3}\right)\lambda^2 + x_2) - y_1\right) \\
&= \left(\beta(\lambda^2 - x_1 - x_2), \lambda(2x_1 - \lambda^2 + x_2) - y_1\right) \\
&= \phi(P_1 + P_2)
\end{aligned}$$

□

Proposition 9.4 in fact shows that ϕ is a distortion map. Since ϕ respects addition on E,

$$\phi([n]P) = \underbrace{\phi(P) + \cdots + \phi(P)}_{n} = [n]\phi(P).$$

Now, since the Weil pairing will be a root of unity as long as both points are in $E[m]$, in order to satisfy condition (b) of Definition 7.1 we just need $\phi(Q)$ to be in $E[m]$ for all $Q \in E[m]$. So if $Q \in E[m]$,

$$\mathcal{O} = \phi(\mathcal{O}) = \phi([m]Q) = [m]\phi(Q).$$

Thus $\phi(Q) \in E[m]$, and the ϕ defined above is indeed a distortion map.

Now, we develop fields in which such an element β exists, before creating an example of a modified Weil pairing.

Proposition 9.5. *For prime p with $p \equiv 2 \bmod 3$, $\mathbb{F}_p$ does not contain a primitive cube root of unity, but $\mathbb{F}_{p^2}$ does.*

Proof Let prime p with $p \equiv 2 \bmod 3$. Assume $\beta \in \mathbb{F}_p$ such that $\beta^3 = 1$ and $\beta \neq 1$. Then the order of β in $\mathbb{F}_p$ must be 3. So, by Lagrange's Theorem, 3 divides the order

of the multiplicative group $\mathbb{Z}/p\mathbb{Z}^* = \mathbb{F}_p - \{0\}$. Then $p - 1 = 3k$ for some integer k, or: $p \equiv 1 \bmod 3$. This is a contradiction.

Now, assume there exists a $\beta \neq 1$ such that $\beta^3 = 1$. If we can show that the set $\{a + b\beta : a, b \in \mathbb{F}_p\}$ is closed under multiplication and has p^2 elements (which it obviously does), then it is $\mathbb{F}_{p^2}$. Now, to show it is closed under multiplication, let $a + b\beta$ and $c + d\beta$ be two arbitrary elements.

$$(a + b\beta)(c + d\beta) = ac + \beta(ad + bc) + bd\beta^2$$

But, $\beta^3 = 1$ implies $0 = \beta^3 - 1 = (\beta - 1)(\beta^2 + \beta + 1)$. Since we know $\beta \neq 1$, we have $\beta^2 + \beta + 1 = 0$ which implies $\beta^2 = -(\beta + 1)$. Thus:

$$\begin{aligned}(a + b\beta)(c + d\beta) &= ac + \beta(ad + bc) - bd(\beta + 1) \\ &= (ac - bd) + \beta(ad + bc - bd)\end{aligned}$$ □

Thus we will use $\mathbb{F}_{p^2}$ as the underlying field in this distortion map and the modified Weil pairing.

Example 9.6. We show an example of a modified Weil pairing, where $p = 11$ and $E : Y^2 = X^3 + 1$. It is easily verifiable that $P = (5, 4) \in E[4]$ on $E(\mathbb{F}_{11})$. Then using our distortion map, $\phi(P) = (5\beta, 4)$, which also belongs to $E[4]$. Then

$$\hat{e}_4(P, P) = e_4\big(P, \phi(P)\big) = e_4\big((5, 4), (5\beta, 4)\big).$$

Plugging this into our Weil pairing function which has been adapted to work in $\mathbb{F}_{p^2}$,(found in Appendix D) and using $S = (7, 6)$, we end up with $\hat{e}_4(P, P) = 9 + 7\beta$. To verify this is a fourth root of unity in $\mathbb{F}_{11^2}$,

$$\begin{aligned}(9 + 7\beta)(9 + 7\beta) &= (81 - 49) + \beta(63 + 63 - 49) \bmod 11 \\ &\qquad \text{(by our work in the proof of Proposition 9.5)} \\ &= 32 + 77\beta \bmod 11 \\ &= 10\end{aligned}$$

Thus

$$(9 + 7\beta)^4 = \big((9 + 7\beta)^2\big)^2 = 10^2 = 1 \in \mathbb{F}_{11^2}.$$

Now we have the necessary tools for our tripartite Diffie-Helman key exchange. The first steps of the process are similar to the bipartite EC Difffie-Helman seen in Example 8.2: the common parameters of $q = p^2$ for some prime such that $p \equiv 2 \bmod 3$,

the curve ($Y^2 = X^3 + 1$ in this construction), and a starting point $P \in E[m]$ are published. In the first computation, Alice, Bob and Carlos compute $Q_a = [a]P$, $Q_b = [b]P$, $Q_c = [c]P$, respectively, where a, b, c are their respective secret integers. They then send out the first communications: the x-values of Q_a, Q_b, and Q_c. Now we will take the viewpoint of Alice: she computes the modified Weil pairing of Q_b and Q_c, and raise it to the power of her secret integer, a. This gives:

$$\hat{e}_m(Q_b, Q_c)^a = \hat{e}_m([b]P, [c]P)^a = \hat{e}_m(P, P)^{abc}.$$

If each of the three parties raise the modified Weil pairing of the other two points to their secret integer, they will each have the same key.

Here, as in the bipartite EC Diffie-Helman, Eve has to conquer the ECDLP in order to know the shared key. Since she knows Q_a, Q_b and Q_c from the round of communications, she is easily able to compute, say, $\hat{e}_m(Q_b, Q_c)$. But, she would need to know the secret integer a in order to have the common key. Thus if m is adequately large, there is no practical way for her to attain the common key.

Example 9.7. Now, we create an example tripartite Diffie-Helman key exchange. We begin with $p = 1223$, which satisfies $p \equiv 2 \bmod 3$, over the curve $Y^2 = X^3 + 1$. By inspection, we find that $P = (1103, 1213) \in E[408]$. For computing the Weil pairings, we use $S = (0, 1222) \in E[4]$, although in application, each member would likely use a different S. This will not effect the values attained since e_m is independent of choice of S. After communicating p, E and P, the first round of computations will be using the Double-and-Add algorithm to compute $[a]P, [b]P$, and $[c]P$. Supposing

$a = 121, b = 433, c = 97$, we have:

$$Q_a = [a]P = (694, 1049), \quad Q_b = [b]P = (764, 140), \quad Q_c = [c]P = (18, 84).$$

The x-values of these points are communicated, as in the bipartite EC Diffie-Helman key exchange, and our members begin the common key computation.

Each party will compute the modified Weil pairing of the other two points, and then raise it to their own secret integer. So, Alice's computation will be:

$$\begin{aligned}
\hat{e}_{408}(Q_b, Q_c)^a &= \hat{e}_{408}\big((764, 140), (18, 84)\big)^{121} \\
&= e_{408}\big((764, 140), (18\beta, 84)\big)^{121} \\
&= (438 + 50\beta)^{121} \\
&= 1094 + 192\beta
\end{aligned}$$

Similarly, Bob's computation will be:

$$\begin{aligned}
\hat{e}_{408}(Q_a, Q_c)^b &= \hat{e}_{408}\big((694, 1049), (18, 84)\big)^{433} \\
&= e_{408}\big((694, 1049), (18\beta, 84)\big)^{433} \\
&= (904 + 393\beta)^{433} \\
&= 1094 + 192\beta
\end{aligned}$$

Common Parameters Communicated $p = 1223,\ \ Y^2 = X^3 + 1,\ \ P = (1103, 1213) \in E[408]$		
Secret Integers		
Alice: $a = 121$	Bob: $b = 433$	Carlos: $c = 97$
First Computations		
$x\{Q_a\} = x\{[121]P\}$ $= x\{(694, 1049)\}$ $= 694$	$x\{Q_b\} = x\{[433]P\}$ $= x\{(764, 140)\}$ $= 764$	$x\{Q_c\} = x\{[97]P\}$ $= x\{(18, 84)\}$ $= 18$
Communication		
$x\{Q_a\} = 694,\ \ x\{Q_b\} = 764,\ \ x\{Q_c\} = 18$		
Common Key Computations		
$\hat{e}_{408}(Q_b, Q_c)^a$ $= (438 + 50\beta)^{121}$ $= 1094 + 192\beta$	$\hat{e}_{408}(Q_a, Q_c)^b$ $= (904 + 393\beta)^{433}$ $= 1094 + 192\beta$	$\hat{e}_{408}(Q_a, Q_b)^c$ $= (1103 + 1041\beta)^{97}$ $= 1094 + 192\beta$

Table 7: Tripartite Diffie-Hellman key exchange over $E(\mathbb{F}_{1223})$

Finally, for Carlos:

$$\begin{aligned}\hat{e}_{408}(Q_a, Q_b)^c &= \hat{e}_{408}\big((694, 1049), (764, 140)\big)^{97} \\ &= e_{408}\big((694, 1049), (764\beta, 140)\big)^{97} \\ &= (438_5 0\beta)^{97} \\ &= 1094 + 192\beta\end{aligned}$$

Included in the Appendix D are tools for working in $\mathbb{F}_{p^2}$, and using these it is verified that $1094 + 192\beta$ is a 408th root of unity in $\mathbb{F}_{p^2}$. This will be the common key shared by the three parties. The process is summarized in Table 7.

Appendices

A: Mathematica program for Weierstrass form

This section consists of **weirform**, a Mathematica program which, when input a general homogenous cubic polynomial, will output the Weierstrass form of the polynomial, as given in Theorem 1.6.

```
F = 1 x^3 + 0 x^2 y + 0 x y^2 + 0 y^3 + 2 x^2 z - 3 x z^2 + 0 x y z + 1 y^2 z + 0 y z^2 + 4 z^3
```

$$x^3 + 2\,x^2\,z + y^2\,z - 3\,x\,z^2 + 4\,z^3$$

```
weirform[F_] :=
 Block[{f, inflection, X1, Y1, Z1, X2, Y2, Z2, Xi, Yi, Zi, A1, F1, R, F1y, tang, A2,
   F2, A3, F3, A4, F4, A5, F5, A6, A7, matrix},
  f = F;
  inflection[f] := Block[{DHess, DH1, DH0, sol, Xs, X, Y, Z, G, FZ1, FZ0},
    DHess = Det[D[f, {{x, y, z}, 2}]];
    DH1 = DHess /. z → 1;
    FZ1 = f /. z → 1;
    sol = FindInstance[{DH1 == 0 && FZ1 == 0}, {x, y}, Reals] // N;
    X = x /. sol[[1, 1]];
    Y = y /. sol[[1, 2]];
    Z = 1;
    Return[{X, Y, Z}]];
  {Xi, Yi, Zi} = Chop[inflection[f]];
  A1 = {{0, Xi, 1}, {0, Yi, 1}, {1, 1, 0}};
  {X1, Y1, Z1} = Chop[A1.{x, y, z}];
  F1 = Chop[Expand[F /. {x -> X1, y -> Y1, z -> Z1}]];
  F1y = F1 /. y → 1;
  tang = {D[F1y, {x, 1}] /. {x → 0, z → 0}, D[F1y, {z, 1}] /. {x → 0, z → 0}};
  R = RotationMatrix[{tang, {0, 1}}];
  A2 = {{R[[1, 1]], 0, R[[2, 1]]}, {0, 1, 0}, {R[[1, 2]], 0, R[[2, 2]]}};
  {X2, Y2, Z2} = Chop[A2.{x, y, z}];
  F2 = Chop[Expand[F1 /. {x → X2, y → Y2, z → Z2}]];
  F3 = F2 /. z → z / Coefficient[F2, y^2 z];
  A3 = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1 / Coefficient[F2, y^2 z]}};
  F4 = Chop[Expand[F3 /. y → y - x / 2 Coefficient[F3, x y z] - z / 2 Coefficient[F3, y z^2]]];
  A4 = {{1, 0, 0}, {-1 / 2 Coefficient[F3, x y z], 1, -1 / 2 Coefficient[F3, y z^2]}, {0, 0, 1}};
  F5 = Chop[Expand[1 / Coefficient[F4, x^3] F4 /. z → z Coefficient[F4, x^3]]];
  A5 = {{1, 0, 0}, {0, 1, 0}, {0, 0, Coefficient[F4, x^3]}};
  A6 = {{1 / Coefficient[F4, x^3]^(1 / 3), 0, 0}, {0, 1 / Coefficient[F4, x^3]^(1 / 3), 0},
    {0, 0, -1 / Coefficient[F4, x^3]^(1 / 3)}};
  A7 = {{1, 0, Coefficient[F5, x^2 z] / 3}, {0, 1, 0}, {0, 0, 1}};
  matrix = A1.A2.A3.A4.A5.A6.A7;
  {XX, YY, ZZ} = Chop[matrix.{x, y, z}];
  f = Chop[Expand[F /. {x → XX, y → YY, z → ZZ}]];
  Return[f]]
```

```
weirform[F]
```

$$1.\,x^3 - 1.\,y^2\,z - 0.000189953\,x\,z^2 - 1.91334 \times 10^{-6}\,z^3$$

B: Mathematica programs for EC's over $\mathbb{C}$

The following Mathematica programs are meant to be used when the underlying field is $\mathbb{C}$. All of the programs assume that the EC has been put into Weierstrass form of Theorem 1.6, and require the input of **a** and **b**, the coefficients of the elliptic curve.

The program **esum** computes the elliptic curve addition algorithm, Theorem 1.8, on $E(\mathbb{C})$.

```
a = 1; b = 2;

esum[p_, q_] := Block[{x1, y1, x2, y2, m, x3, y3},
   If[p == "O" && q == "O", Return["O"]];
   If[p == "O", Return[q]];
   If[q == "O", Return[p]];
   {x1, y1} = p;
   {x2, y2} = q;
   If[x1 == x2 && y1 == -y2, Return["O"]];
   If[x1 == x2, λ = Chop[(3 x1^2 + a) / (2 y1)], λ = Chop[(y1 - y2) / (x1 - x2)]];
   x3 = Chop[λ^2 - x1 - x2 ] // N;
   y3 = Chop[λ (x3 - x1) + y1] // N;
   Return[{x3, -y3}];
  ];
```

The double-and-add algorithm of Section 2 is implemented in the program **doubleadd**, which requires the use of **esum**.

```
doubleadd[n_, p_] := Block[{N = n, U = p, R = "O"},
  While[N > 0, If[Mod[N, 2] == 1, R = esum[R, U]]; U = esum[U, U]; N = Floor[N / 2]]; Return[R]
 ]
```

The program **H** computes the function $h_{T,T}$ (in Theorem 5.1 Part (a)) with desired divisors for Miller's algorithm. Note that implementation will require the program **esum**.

```
H[P_, Q_] := Block[{s},
  Which[
    P[[1]] == Q[[1]] && P[[2]] == -Q[[2]], Return[x - P[[1]]],
    P[[1]] == Q[[1]] && P[[2]] == Q[[2]], λ = Chop[(3 P[[1]]^2 + a) / (2 P[[2]])];
    Return[Chop[(y - P[[2]] - λ (x - P[[1]])) / (x + P[[1]] + Q[[1]] - λ^2)]],
    P[[1]] ≠ Q[[1]], λ = Chop[(P[[2]] - Q[[2]]) / (P[[1]] - Q[[1]])];
    Return[Chop[(y - P[[2]] - λ (x - P[[1]])) / (x + P[[1]] + Q[[1]] - λ ^2)]]];
 ]
```

The next program, **millercalc**, implements Miller's algorithm while plugging in the necessary x and y-values to compute the Weil pairing. Note that implementation requires the program **H**, and thus **esum**.

```
millercalc[m_, P_, R_] := Block[{T = P, f = 1, mlist = Reverse[IntegerDigits[m, 2]]},
  Do[f = Chop[f^2 H[T, T]];
   f = Chop[f /. {x → R[[1]], y → R[[2]]}];
   T = esum[T, T];
   If[mlist[[i]] == 1, f = f H[T, P];
    f = Chop[f /. {x → R[[1]], y → R[[2]]}];
    T = esum[T, P] ];
   , {i, Length[mlist] - 1, 1, -1}];
  Return[f]
 ]
```

Finally, the program **weilpairing** uses the **millercalc** program to find and evaluate f_P and f_Q at the desired points, giving the Weil pairing of P and Q.

```
weilpairing[m_, P_, Q_, S_] := Block[{A, B, C, D, nS = {S[[1]], -S[[2]]} },
  A = millercalc[m, P, esum[Q, S]];
  B = millercalc[m, P, S];
  C = millercalc[m, Q, esum[P, nS]];
  D = millercalc[m, Q, nS];
  Return[Chop[(A D) / (B C)]]
 ]
```

C: Mathematica programs for EC's over $\mathbb{F}_p$

This section consists of **fpesum**, **fpdoubleadd**, **fpH**, **fpmillercalc**, and **fpweilpairing**, all of which serve the same purposes of their counterparts in Appendix B, but work over a finite field F_p for prime p. Note that these also require the curve to be in the Weierstrass form of Theorem 1.6, and thus require the input of **a** and **b**, the coefficients of the elliptic curve. These programs also require the input of **mod**, the prime p which the modular arithmetic will be done under.

```
mod = 1223;
a = 30; b = 34;
```

```
fpesum[p_, q_] := Block[{x1, y1, x2, y2, λ, x3, y3},
   If[p == "O" && q == "O", Return["O"]];
   If[p == "O", Return[q]];
   If[q == "O", Return[p]];
   {x1, y1} = p;
   {x2, y2} = q;
   If[x1 == x2 && y2 == Mod[-y1, mod], Return["O"]];
   If[x1 == x2, λ = PolynomialMod[(3 x1^2 + a) / (2 y1), mod],
    λ = PolynomialMod[(y1 - y2) / (x1 - x2), mod]];

   x3 = Mod[PowerMod[λ, 2, mod] - x1 - x2 , mod];
   y3 = Mod[λ (x3 - x1) + y1, mod];

   Return[{x3, Mod[-y3, mod]}];
  ];
```

```
fpdoubleadd[n_, p_] := Block[{N = n, U = p, R = "O"},
   While[N > 0, If[Mod[N, 2] == 1, R = fpesum[R, U]]; U = fpesum[U, U]; N = Floor[N / 2]]; Return[R]
  ];
```

```
fpH[p_, q_] := Block[{λ},
   Which[
     p[[1]] == q[[1]] && p[[2]] == Mod[-q[[2]], mod], Return[PolynomialMod[x - p[[1]], mod]],
     p[[1]] == q[[1]] && p[[2]] == q[[2]],
     λ = PolynomialMod[(3 p[[1]]^2 + a ) / (2 p[[2]]), mod];
     Return[PolynomialMod[(y - p[[2]] - λ (x - p[[1]])) / (x + p[[1]] + q[[1]] - λ^2), mod]],
     p[[1]] ≠ q[[1]], λ = PolynomialMod[(p[[2]] - q[[2]]) / (p[[1]] - q[[1]]), mod];
     Return[PolynomialMod[(y - p[[2]] - λ (x - p[[1]])) / (x + p[[1]] + q[[1]] - λ^2), mod]] ];
  ];
```

```
  fpmillercalc[M_, p_, R_] := Block[{T = p, f = 1, mlist = Reverse[IntegerDigits[M, 2]]},
     Do[f = PolynomialMod[f^2 fpH[T, T], mod];
      f = PolynomialMod[f /. {x → R[[1]], y → R[[2]]}, mod];
      T = fpesum[T, T];
      If[mlist[[i]] == 1,
       f = PolynomialMod[f fpH[T, p], mod];
       T = fpesum[T, p];
       f = PolynomialMod[f /. {x → R[[1]], y → R[[2]]}, mod]];
      ,
      {i, Length[mlist] - 1, 1, -1}];
     Return[f]
    ];
```

```
fpweilpairing[m_, P_, Q_, S_] := Block[{A, B, C, D, nS = {S[[1]], -S[[2]]}},
  A = fpmillercalc[m, P, esum[Q, S]];
  B = fpmillercalc[m, P, S];
  C = fpmillercalc[m, Q, esum[P, nS]];
  D = fpmillercalc[m, Q, nS];
  Return[PolynomialMod[(A D) / (B C), mod]]
 ]
```

D: Mathematica programs for EC's over $\mathbb{F}_{p^2}$

This section contains all of the equivalent programs as Appendix C, but made to work in the field $\mathbb{F}_{p^2}$. There are first a few programs we need in order to work easily with numbers of the form $a + b\beta \in \mathbb{F}_{p^2}$. The program **fp2multiply** takes as input two numbers of the form $a + b\beta$ and outputs their multiplied value in $\mathbb{F}_{p^2}$, while the program **fp2power** uses a double-and-add algorithm to repeatedly multiply a number in $\mathbb{F}_{p^2}$ by itself n times.

```
fp2multiply[x_, y_] :=
 Block[{a = Coefficient[x, β, 0], b = Coefficient[x, β], c = Coefficient[y, β, 0],
   d = Coefficient[y, β]}, Return[PolynomialMod[(a c - b d) + β (a d + b c - b d), mod]] ]

fp2power[n_, x_] := Block[{N = n, U = x, R = 1},
  While[N > 0, If[Mod[N, 2] == 1, R = fp2multiply[R, U]]; U = fp2multiply[U, U]; N = Floor[N / 2]];
  Return[R]
 ]
```

The program **fp2inverse** creates the multiplicative inverse of a number in $\mathbb{F}_{p^2}$. Thus, if one has $x, y \in \mathbb{F}_{p^2}$ and wants to compute the value of $x/y \in \mathbb{F}_{p^2}$, they would compute: **fp2multiply[x,fp2inverse[y]].**

```
fp2inverse[x_] := Block[{a = Coefficient[x, β, 0], b = Coefficient[x, β], inv},
  inv = PolynomialMod[(a - b) / (a^2 - a b + b^2) - b / (a^2 - a b + b^2) β, mod];
  Return[inv];
 ]
```

Finally, we have the equivalent programs to work on elliptic curves in $\mathbb{F}_{p^2}$. Note that we still only need to input **mod**$= p$, not p^2.

```
a = 0; b = 1; mod = 1223;

fp2esum[p_, q_] := Block[{x1, y1, x2, y2, λ, x3, y3},
   If[p == "O" && q == "O", Return["O"]];
   If[p == "O", Return[q]];
   If[q == "O", Return[p]];
   {x1, y1} = p;
   {x2, y2} = q;
   If[ToString[PolynomialMod[x1, mod]] == ToString[PolynomialMod[x2, mod]] &&
     ToString[PolynomialMod[y1, mod]] == ToString[PolynomialMod[-y2, mod]], Return["O"]];
   If[ToString[PolynomialMod[x1, mod]] == ToString[PolynomialMod[x2, mod]],
    λ = PolynomialMod[(3 x1^2 + a) / (2 y1), {β^2 + β + 1, mod}],
    λ = PolynomialMod[(y1 - y2) / (x1 - x2), {β^2 + β + 1, mod}]];

   x3 = PolynomialMod[λ^2 - x1 - x2 , {β^2 + β + 1, mod}];
   y3 = PolynomialMod[λ (x3 - x1) + y1, {β^2 + β + 1, mod}];

   Return[{x3, PolynomialMod[-y3, mod]}];
  ];

fp2doubleadd[n_, p_] := Block[{N = n, U = p, R = "O"},
  While[N > 0, If[Mod[N, 2] == 1, R = fp2esum[R, U]]; U = fp2esum[U, U]; N = Floor[N / 2]]; Return[R]
 ]

fp2H[p_, q_] := Block[{λ, x1, x2, y1, y2},
   {x1, y1} = p;
   {x2, y2} = q;
   Which[
    ToString[PolynomialMod[x1, mod]] == ToString[PolynomialMod[x2, mod]] &&
     ToString[PolynomialMod[y1, mod]] == ToString[PolynomialMod[-y2, mod]],
    Return[PolynomialMod[x - x1, {β^2 + β + 1, mod}]],
    ToString[PolynomialMod[x1, mod]] == ToString[PolynomialMod[x2, mod]] &&
     ToString[PolynomialMod[y1, mod]] == ToString[PolynomialMod[y2, mod]],
    λ = PolynomialMod[(3 x1^2 + a ) / (2 y1), {β^2 + β + 1, mod}];
    Return[PolynomialMod[(y - y1 - λ (x - x1)) / (x + x1 + x2 - λ^2), {β^2 + β + 1, mod}]],
    ToString[PolynomialMod[x1, mod]] ≠ ToString[PolynomialMod[x2, mod]],
    λ = PolynomialMod[(y2 - y1) / (x2 - x1), {β^2 + β + 1, mod}];
    Return[PolynomialMod[(y - y1 - λ (x - x1)) / (x + x1 + x2 - λ^2), {β^2 + β + 1, mod}]] ];
  ];


fp2millercalc[M_, p_, R_] := Block[{T = p, f = 1, mlist = Reverse[IntegerDigits[M, 2]]},
   Do[f = PolynomialMod[(f^2) fp2H[T, T], {β^2 + β + 1, mod}];
    f = PolynomialMod[f /. {x → R[[1]], y → R[[2]]}, {β^2 + β + 1, mod}];
    T = fp2esum[T, T];
    If[mlist[[i]] == 1,
     f = PolynomialMod[f fp2H[T, p], {β^2 + β + 1, mod}];
     T = fp2esum[T, p];
     f = PolynomialMod[f /. {x → R[[1]], y → R[[2]]}, {β^2 + β + 1, mod}]];
    ,
    {i, Length[mlist] - 1, 1, -1}];
   Return[f]
  ];

fp2weilpairing[m_, P_, Q_, S_] :=
 Block[{A, B, C, D, WP, nS = {S[[1]], PolynomialMod[-S[[2]], {β^2 + β + 1, mod}]}},
  A = fp2millercalc[m, P, esum[Q, S]];
  B = fp2millercalc[m, P, S];
  C = fp2millercalc[m, Q, esum[P, nS]];
  D = fp2millercalc[m, Q, nS];
  WP = PolynomialMod[(A D) / (B C), {β^2 + β + 1, mod}];
  Return[PolynomialMod[Numerator[WP] fp2inverse[Denominator[WP]], {β^2 + β + 1, mod}]]
 ]
```

Bibliography

[1] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman *An Introduction to Mathematical Cryptography*. Springer, New York, 2008.

[2] Joseph H. Silverman *The Arithmetic of Elliptic Curves*. Springer-Verlag, New York, 1986.

[3] Joseph H. Silverman, John Tate *Rational Points on Elliptic Curves*. Springer, New York, 1992.

[4] Victor S. Miller *Short Programs for Functions on Curves*. Unpublished, IBM, Thomas J. Watson Research Center, 1986.

[5] Antoine Joux *A One Round Protocol for Tripartite Diffie-Hellman*. Journal of Cryptography, 17: 263276, 2004.

[6] Antoine Joux, Kim Nguyen *Separating Decision Diffie-Hellman from Computational Diffie-Hellman in Cryptographic Groups*. Journal of Cryptography, 16: 239247, 2003.

Vita

Alex Edward Aftuck

Place of birth:

Date of birth:

Education:

Bachelor of Science: Mathematical Sciences, May 2008

University of North Florida: Jacksonville, FL

Masters of Science: Mathematical Sciences, August 2011

University of North Florida: Jacksonville, FL

Teaching Experience: Graduate Teaching Assistant

Department of Mathematics and Statistics

University of North Florida: Jacksonville, FL