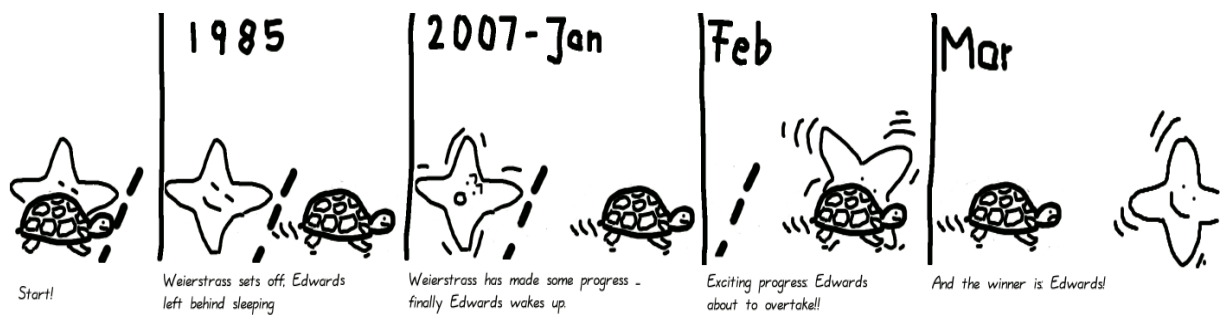


# MATHEMATICAL ASPECTS OF CRYPTOLOGY

## *Elliptic curve factorization using Edwards curves*



BACHELOR THESIS IN MATHEMATICS

TORBEN BRANDT HANSEN, 20091714

SEPTEMBER 20, 2012

ADVISOR: JØRGEN BRANDT



AARHUS  
UNIVERSITY

DEPARTMENT OF MATHEMATICS

Copyright © 2012 Torben Brandt Hansen

Typeset using L<sup>A</sup>T<sub>E</sub>X and the memoir document class. Graphics made using Geogebra.

Bibtex info:

```
@Misc{bachelorThesis:Torben,  
author = {Torben B. Hansen},  
title = {Mathematical Aspects of Cryptology - Elliptic curve factorization using  
Edwards curves},  
year = {2012},  
note = {Program available at http://home.imf.au.dk/himsen/Cryptography.html },  
}
```

# Introduction

Reflecting on the quote below by Gauss, consider the ancient problem: “Given an integer  $n$ , how do we factor  $n$ ?”. Actually this question is not difficult to answer; use trial division. So let us restate the problem: “Given an integer  $n$ , how do we factor  $n$  **fast**?”. Hopefully the reader knows that this turns out to be extremely difficult; the polynomial complexity barrier has not yet been accomplished and who knows, maybe we will never get there?

The interests in factoring integers may seem strange at a first glance. But numerous motivations exists:

- Security for some cryptographic schemes rest on the hardness of factoring. In this connection it is valuable information to know how big numbers, that human kind together with computers are able to factor.
- Factoring are used in several primality proving algorithms, see e.g. [10].
- Are there anything more fun than combining the awesome power of a computer with the beauty of mathematics?

This thesis attacks the factorization problem using the well known *elliptic curve method* originally developed by Lenstra in the mid 1980’s. Since then, multiple optimizations improving Lenstra’s algorithm has been developed; a 2. stage method and use of other elliptic curve models. In 2007 a new model for ECM was proposed by Daniel Bernstein and Tanja Lange building from work of Harold Edwards. It is this thesis goal to study and examine the work by Bernstein and Lange and make an implementation using there ideas.

In chapter 1 basic definitions and a brief reminder of the basic elliptic curve theory is stated. Since this thesis has been written in extension of a course of which curriculum contains some theory of elliptic curves, this is assumed known and as an effect, do not contain proofs.

Continuing to chapter 2 we analysis the original algorithm by Lenstra including a discussion of the standard continuation of the algorithm.

Chapter 3 develop and discuss the basic theory of Edwards curves. We present the connection between Edwards curves and elliptic curves in Weierstrass form, prove important results and analyse the performance of arithmetic on Edwards curves.

The last chapter serves as the documentation for the implementation of the elliptic curve method using Edwards curves, made by the author. It includes a section of experiments showing the performance of the implementation. A great number of further optimizations are also presented and discussed. The source code can be downloaded by visiting <http://home.imf.au.dk/himsen/Cryptography.html>.

I would like to take the opportunity to thank my advisor Jørgen Brandt for letting me write this thesis and answering my (frequently occurring) naive questions. Also, thank you Ann-Katrine for correction my many silly gramma mistakes with  $\frac{1}{\epsilon}$  precision - indeed ECM is paralyzing.

Torben Hansen, 29-08-2012.

*The problem of distinguishing prime numbers from composite numbers and  
of resolving the latter into their prime factors is known to be one of the  
most important and useful in arithmetic.*  
— Carl Friedrich Gauss(1777–1855)

# Contents

<b>Introduction</b>	<b>i</b>
<b>1 Elliptic curves</b>	<b>1</b>
1.1 Definition of an elliptic curve . . . . .	1
1.2 Weierstrass model . . . . .	2
1.3 Group structure on elliptic curves . . . . .	3
<b>2 Basic ECM</b>	<b>5</b>
2.1 Pseudo elliptic curves . . . . .	5
2.2 ECM . . . . .	7
2.3 Complexity . . . . .	9
2.4 2. Stage . . . . .	11
<b>3 Edwards curves</b>	<b>13</b>
3.1 Edwards curves . . . . .	13
3.2 Addition on Edwards curves . . . . .	17
3.3 Efficient operations on Edwards curves . . . . .	23
<b>4 ECM using Edwards curves</b>	<b>25</b>
4.1 Using Edwards curves . . . . .	26
4.2 Stage two . . . . .	26
4.3 Modular arithmetic . . . . .	27
4.4 Single-scalar multiplication . . . . .	28
4.5 Bounds $B_1$ and $B_2$ . . . . .	30
4.6 Curve selection . . . . .	31
4.7 Prime generation . . . . .	32
4.8 Additional comments . . . . .	33
4.9 Experiments . . . . .	33
<b>A Maple scripts</b>	<b>37</b>

<b>Bibliography</b>	<b>39</b>
---------------------	-----------

## CHAPTER 1

# Elliptic curves

In this chapter we define elliptic curves and present results which will be useful. One can consider this chapter as a toolbox for later.

### 1.1 Definition of an elliptic curve

Let  $\mathbb{F}$  be an arbitrary field and put  $f(x, y) = ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + iy + j$ . If at least one of  $a, b, c, d$  is non-zero then

$$f(x, y) = 0 \tag{1.1}$$

is a degree 3 curve in  $\mathbb{F}^2$  with affine solutions  $(x, y) \in \mathbb{F}^2$ . If we homogenize  $f$  we have  $F(x, y, z) = ax^3 + bx^2y + cxy^2 + dy^3 + ex^2z + fxyz + gy^2z + hxz^2 + iyz^2 + jz^3$  and the projective form of (1.1) is then

$$F(x, y, z) = 0. \tag{1.2}$$

This curve has the solutions  $(x, y, z) \in \mathbb{F}^3$ . Notice that if  $(x, y, z)$  is an solution with  $(x, y, z) \neq (0, 0, 0)$  then so is  $(\alpha x, \alpha y, \alpha z)$  for any  $\alpha \in \mathbb{F}^*$ . Hence it makes more sense to talk about solutions to (1.2) as being in the projective space  $\mathbb{P}^2(\mathbb{F})$ . Recall that the projective space  $\mathbb{P}^2(\mathbb{F})$  consists of equivalence classes  $[x, y, z]$  with respect to the equivalence relation:

$$(x, y, z) \sim (x', y', z') \Leftrightarrow \exists \alpha \in \mathbb{F}^* : \alpha x = x', \alpha y = y', \alpha z = z'.$$

The curve (1.2) is called non-singular if over  $\overline{\mathbb{F}}$  there is no point  $[x, y, z]$  on the curve (1.2) where all three partial derivatives of  $F$  vanish. We are now ready to state the definition of an elliptic curve.

**Definition 1.1.1.** A non-singular cubic curve of the form (1.2) with at least one rational point  $(x, y, z) \in \mathbb{F}^3 \setminus (0, 0, 0)$  is said to be an elliptic curve over  $\mathbb{F}$ .

Consider now the homogeneous equation

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3 \quad (1.3)$$

with  $a_i \in \mathbb{F}$  for some arbitrary field  $\mathbb{F}$ . It turns out that if the defining equation (1.2) is an elliptic curve then it is birationally equivalent to (1.3), see [18]. That is, we may express all elliptic curves on the form (1.3).

## 1.2 Weierstrass model

Assume we are given an elliptic curve of the form (1.3) over the field  $\mathbb{F}$  and let  $[x, y, z]$  be a point on it. Then  $(x, y, z) \neq (0, 0, 0)$  by assumption. The points with  $z = 0$  are called the points at infinity. Actually we should rather say *the* point at infinity; putting the point  $[x, y, 0]$  into (1.3) yield  $0 = x^3$  hence  $x = 0$ . Since points are only determined up to multiplication by a unit ( $y \neq 0$  since  $x = 0 = z$  and  $(0, 0, 0)$  is excluded),  $[0, 1, 0]$  must be the only point at infinity.

Next consider the curve

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \quad (1.4)$$

This is called the affine part of an elliptic curve. Solutions to the above curve are embedded in the solutions for (1.3) by  $(x, y) \mapsto [x, y, 1]$ . Likewise if  $z \neq 0$ , a solution  $[x, y, z]$  for the projective curve corresponds to the solution  $(x/z, y/z)$  for the affine curve. When  $z = 0$ ,  $[x, y, z]$  do not correspond to a solution on the affine curve - but there are only one of these. All points on the projective curve has the form  $[x, y, 1]$ , except the point at infinity, so we may interpret the points on the projective curves as points  $(x, y)$  satisfying (1.4) plus the point at infinity. When working with the affine part of an elliptic curve, the one point at infinity will be denoted by  $\mathcal{O}$ .

If the curve (1.4) defines an elliptic curve it is said to be in long Weierstrass form. If  $\text{char}(\mathbb{F}) \neq 2$  we may use the transformation  $(x, y) \mapsto (x, y - \frac{a_1x+a_3}{2})$  to transform the long Weierstrass form into Weierstrass form:

$$y^2 = x^3 + Cx^2 + Ax + B, \quad A, B, C \in \mathbb{F}. \quad (1.5)$$

Further if  $\text{char}(\mathbb{F}) \neq 2, 3$  then we can transform the above curve into short Weierstrass form

$$y^2 = x^3 + ax + b \quad (1.6)$$

using the transformation  $(x, y) \mapsto (x - \frac{C}{3}, y)$ .

When we are faced with curves on the form (1.5) and (1.6) it is easy to see whether they define an elliptic curve or not; in the case of (1.6) it defines an elliptic curve if  $4a^3 + 27b^2 \neq 0$  and (1.5) defines an elliptic curve if  $4A^3 + 27B^2 - 19ABC - A^2C^2 + 4BC^3 \neq 0$ .



From now on when we write elliptic curve one should think of the points on the curve (1.4) plus the point at infinity  $\mathcal{O}$  which is the projective point  $[0, 1, 0]$  on the projective form of the affine curve although we will be using the projective coordinates in practice because they induce faster addition and doubling. The special case curves (1.5) and (1.6) will be denoted  $E_{W,a,b,c}$  and  $E_{W,a,b}$  respectively.

### 1.3 Group structure on elliptic curves

**Definition 1.3.1.** Let  $E$  be an elliptic curve over the field  $\mathbb{F}$ . Then  $E(\mathbb{F})$  denotes the set of points on  $E$  plus the point at infinity. In the special cases (1.5) and (1.6) we have

$$E_{W,a,b,c}(\mathbb{F}) = \left\{ (x, y) \in \mathbb{F}^2 \mid y^2 = x^3 + cx^2 + ax + b \right\} \cup \{\mathcal{O}\}$$

$$E_{W,a,b}(\mathbb{F}) = \left\{ (x, y) \in \mathbb{F}^2 \mid y^2 = x^3 + ax + b \right\} \cup \{\mathcal{O}\}$$

By magic it is possible to define a composition on the set  $E(\mathbb{F})$  making it into a group! We restrict ourself to the case  $\text{char}(\mathbb{F}) \neq 2$ . It is possible to define a composition in general but it is more complicated and we do not need it.

Let  $P_1, P_2 \in E_{W,a,b,c}(\mathbb{F})$  (not necessarily distinct) and write  $P_1 = (x_1, y_1)$  and  $P_2 = (x_2, y_2)$ . Define the operator  $\oplus$  by:

- (1)  $-\mathcal{O} = \mathcal{O}$
- (2)  $-P_1 = (x_1, -y_1)$
- (3)  $\mathcal{O} \oplus P_1 = P_1$
- (4) If  $P_1 = -P_2$  then  $P_1 \oplus P_2 = \mathcal{O}$
- (5) If  $P_1 \neq -P_2$  define

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & x_2 \neq x_1 \\ \frac{3x_1^2 + 2cx_1 + a}{2y_1} & x_2 = x_1 \end{cases}$$

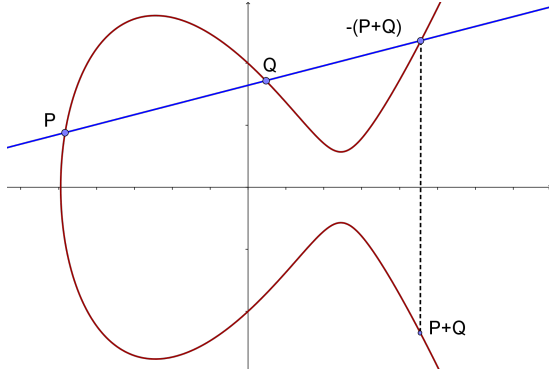
then  $P_1 \oplus P_2 = (x_3, y_3)$  with

$$x_3 = m^2 - c - x_1 - x_2 \tag{1.7}$$

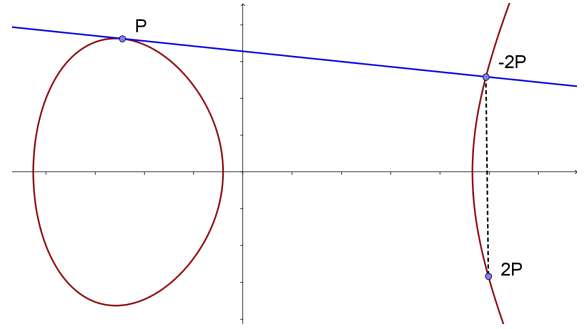
$$y_3 = m(x_1 - x_3) - y_1 \tag{1.8}$$

Popular  $\oplus$  is also referred to as the *chord-tangent construction* because of the following geometrical view: If  $P_1$  and  $P_2$  are distinct and not equal to  $\mathcal{O}$ , then take the straight line through both points and mark the third point of intersection with the curve (this point always exists unless the line is vertical in which case the sum would be the point

at infinity see case 4). Take the marked point and reflect it in the  $x$ -axis (exists since  $(x, y) \in E_{W,a,b,c}(\mathbb{F})$  if and only if  $(x, -y) \in E_{W,a,b,c}(\mathbb{F})$ ) and let this be the sum of  $P_1 \oplus P_2$ . If the two points are equal, take the tangent to that point (this is well defined because of the non-singular condition we put on elliptic curves) and mark the third point of intersection, reflect this point in the  $x$ -axis and let that point be the sum  $P_1 \oplus P_2$ . This geometrical description is a good way to visualize the composition in the case  $\mathbb{F} = \mathbb{R}$  where one may draw the addition, see figure 1.1 and 1.2. When e.g.  $\mathbb{F} = \mathbb{Z}/p\mathbb{Z}$  for some prime  $p$  it may be confusing to try to draw the addition. In any case the composition defines a group on the set  $E_{W,a,b,c}(\mathbb{F})$ . Below is the amazing theorem stating that elliptic curves really are



**Figure 1.1:** Dedicated addition on an elliptic curve.



**Figure 1.2:** Doubling on an elliptic curve.

groups; even finitely generated abelian groups.

**Theorem 1.3.2.** *Let  $\mathbb{F}$  be a field with  $\text{char}(\mathbb{F}) \neq 3$ . Then  $(E_{W,a,b,c}(\mathbb{F}), \oplus)$  is a finitely generate abelian group with  $\mathcal{O}$  as neutral element and if  $P = (x, y) \in E_{W,a,b,c}(\mathbb{F})$  the inverse is  $-P = (x, -y)$ .*

Without confusion we will from now on write  $+$  instead of  $\oplus$ . The hardest thing to prove in the present theorem is the associativity. It can be proven using computer algebra systems such as Maple or if one is masochistic orientated, in hand.

*Remark 1.3.3.*  $(E_{W,a,b}(\mathbb{F}), +)$  also satisfy the above theorem with the defined composition. Simply substitute  $c = 0$  in the formulas. The theorem also apply for general elliptic curves but with a more intricate composition.

Let  $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ . The theorem below is due to Hasse and restrict the curve order of elliptic curves over fields of the form  $\mathbb{F}_p$ .

**Theorem 1.3.4.** *Let  $E_{W,a,b}(\mathbb{F}_p)$  be an elliptic curve. Then  $|E_{W,a,b}(\mathbb{F}_p)| \in [p+1-2\sqrt{p}, p+1+2\sqrt{p}]$ .*

This theorem is key in the analysis of ECM.

## CHAPTER 2

# Basic ECM

In the mid 1980's H. W. Lenstra proposed a new factorization algorithm now known as the *elliptic curve factorization method* abbreviated *ECM*. Even though this algorithm have a worst case complexity equal to some of its competitors its special since the complexity depends on the least prime in the prime factorization of the number trying to factor. Therefore ECM currently provide the fastest means of finding factors of up to approximately 20-40 decimal digits (see e.g. [8]). In practice ECM is often used as subroutines in e.g. the Number Field Sieve.

### 2.1 Pseudo elliptic curves

To formulate ECM it is not enough to know about elliptic curves over fields. We must, to some extend, generalize the concept of an elliptic curve. In the following we describe this generalization and introduce a partial addition on these (pseudo) elliptic curves.

The following example shows that the usual composition on elliptic curves does not give a group structure over a general field.

*Example 2.1.1.* Assume that we define an elliptic curve over  $\mathbb{Z}/n\mathbb{Z}$  with  $n$  composite as in definition 1.1.1 and using the same composition from theorem 1.3.2. Consider the curve  $E = E_{W,-1,1}(\mathbb{Z}/2^5\mathbb{Z})$ . We try to compute  $(1, 1) + (24, 4)$  on  $E$ . First observe that  $(1, 1) = (1, -24) = -(1, 24)$  hence  $(1, 1) + (1, 24) = \infty$ . The  $x$ -coordinate for  $(1, 1) + (41, 4)$  is

$$\left(\frac{3}{20}\right)^2 - 1 - 21$$

but  $\gcd(20, 2^5) > 1$ . We must have  $(1, 1) + (24, 4) = \infty$ . But in a group the element  $(1, 1)$  cannot have two inverse elements.

It should be emphasised that a rigorous construction exists but we will not need this. Therefore we will do with the pseudo construction below.

**Definition 2.1.2.** Let  $a, b \in \mathbb{Z}/n\mathbb{Z}$  with  $\gcd(n, 6) = 1$  and  $4a^3 + 27b^2 \in (\mathbb{Z}/n\mathbb{Z})^*$ . An elliptic curve over the ring  $\mathbb{Z}/n\mathbb{Z}$  is the set

$$E_{W,a,b}(\mathbb{Z}/n\mathbb{Z}) = \{(x, y) \in \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

where  $\mathcal{O}$  is the point at infinity.

While elliptic curves over  $\mathbb{Z}/n\mathbb{Z}$  do not form groups they have a natural projection to curves  $E_{W,a,b}(\mathbb{F}_p)$  with  $p|n$  and  $p > 3$ .

Let  $x \in \mathbb{Z}/n\mathbb{Z}$ . With the notation  $[x]_p$  we mean the unique integer satisfying  $x \equiv [x]_p \pmod{p}$  and  $0 \leq [x]_p < p$ . Also for  $(x, y) \in E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  we define  $(x, y)_p = ([x]_p, [y]_p)$ .

**Definition 2.1.3.** Let  $n \in \mathbb{N}$  and let  $(x, y)$  be an element on the pseudo elliptic curve  $E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  different from the point at infinity. Also let  $p$  be a prime dividing  $n$  and  $p > 3$ . Then the reduction module  $p$  is  $(x, y)_p$  as an element of  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$ . Also define  $\mathcal{O}_p = \mathcal{O}$  as the identity element in  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$

If  $p$  does not divide  $n$  we have no control over whether  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  really is an elliptic curve or not. This follows since by definition  $\gcd(4a^3 + 27b^2, n) = 1$  but then also  $\gcd(4a^3 + 27b^2, p) = 1$  but if  $p \nmid n$  we have no control over the behaviour of  $\gcd$ . If  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  is a well defined elliptic curve with  $p$  not dividing  $n$ , it is called a *good reduction*.

Next we define partial addition on  $E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$ .

**Definition 2.1.4. Partial addition** Given  $P_1, P_2 \in E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$ . Then we define  $P_1 + P_2$  to be the usual addition on elliptic curves if all the required inverse elements exists in the ring  $\mathbb{Z}/n\mathbb{Z}$  and likewise for  $[k]P_1$  with  $k \in \mathbb{N}$ . In this situation we say that the addition is *well defined*. If the addition require an inverse element which is not present it is not possible to make the addition and we say that the addition *fail*.

The definition of  $[k]P$  being well defined has some subtleties. For  $k = 8$  we may calculate  $[8]P$  in different ways e.g.  $((([2]P + [2]P) + [2]P) + [2]P)$  or  $([2]P + [2]P) + ([2]P + [2]P)$ . It may be that one gives a well defined addition but the other does not. If we in some way can calculate  $[k]P$  we say that it is well defined. Finding an inverse to  $x$  in the ring  $\mathbb{Z}/n\mathbb{Z}$  is possible if and only if  $x$  and  $n$  are co-prime. That is, failure in the partial addition occur if and only if  $\gcd(x, n) > 1$ , a possible non trivial factor! It is Lenstra's ingenious observation that through this failure of finding an inverse, we shall be able to factor the composite number  $n$ . The next lemma shows that the projection from definition 2.1.3 is well behaved.

**Lemma 2.1.5.** Let  $R, Q \in E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  and let  $p$  be a prime with  $p|n$  and  $p > 3$ . If the partial addition  $R + Q$  is well defined, then  $(R + Q)_p = R_p + Q_p$  and if  $[k]R$  is well defined then  $([k]R)_p = [k]R_p$  for  $k \in \mathbb{Z}$ ,  $k > 0$ .

*Proof.* We split the proof into the same cases as given in the composition on an elliptic curve, see section 1.3

- (3) Assume  $R = \mathcal{O}$  and  $Q \neq \mathcal{O}$ . Then  $(R + Q)_p = (\mathcal{O} + Q)_p = Q_p = \mathcal{O}_p + Q_p = R_p + Q_p$ . Same apply if  $Q = \infty$  and argument  $R \neq \infty$ . Assume for the  $R, Q \neq \infty$ .
- (4) Assume  $R = -Q$  hence  $R = (x, y)$  and  $Q = (x, -y)$ . Then  $(R + Q)_p = (\mathcal{O})_p = \mathcal{O}$  and  $R_p + Q_p = (x, y)_p + (x, -y)_p = ([x]_p, [y]_p) + ([x]_p, -[y]_p) = \mathcal{O}$  hence  $(R + Q)_p = R_p + Q_p$ .
- (5) Assume  $R \neq -Q$ . Write  $R = (r_1, r_2)$  and  $Q = (q_1, q_2)$ . If  $r_1 \neq q_1$  we have: By assumption  $\gcd(q_1 - r_1, n) = 1$  and  $m = (q_2 - r_2)(q_1 - r_1)^{-1}$  is well defined in  $\mathbb{Z}/n\mathbb{Z}$ . Therefore

$$R + Q = \left( m^2 - r_1 - q_1, m^2 \left( r_1 - (m^2 - r_1 - q_1) \right) - r_2 \right)$$

Then

$$\begin{aligned} (R + Q)_p &= \left( m^2 - r_1 - q_1, m^2 \left( r_1 - (m^2 - r_1 - q_1) \right) - r_2 \right)_p \\ &= \left( [m]_p^2 - [r_1]_p - [q_1]_p, [m]_p^2 \left( [r_1]_p - ([m]_p^2 - [r_1]_p - [q_1]_p) \right) - [r_2]_p \right) \end{aligned}$$

But  $R_p = ([r_1]_p, [r_2]_p)$  and  $Q_p = ([q_1]_p, [q_2]_p)$  hence

$$R_p + Q_p = \left( [m]_p^2 - [r_1]_p - [q_1]_p, [m]_p^2 \left( [r_1]_p - ([m]_p^2 - [r_1]_p - [q_1]_p) \right) - [r_2]_p \right)$$

from the normal addition and  $[m]_p = [(q_2 - r_2)(q_1 - r_1)^{-1}]_p = ([q_2]_p - [r_2]_p)([q_1]_p - [r_1]_p)^{-1}$ .

If  $r_1 = q_1$  the arguments are similar. The well defined assumption in the lemma is in this case used such that we know  $\gcd(2r_2, n) = 1$ .

If both  $R$  and  $Q$  equal  $\mathcal{O}$  it is trivial since by definition  $\mathcal{O}_p = \mathcal{O}$ .  $([k]R)_p = [k]R_p$  now follows by induction.  $\square$

With the knowledge that  $R + Q$  is well defined, we can with the above lemma in hand make statements about  $R_p + Q_p$  without even knowing the exact value of  $p$ . The Only thing we need to know beforehand is that  $p|n$ .

## 2.2 ECM

We will now state the elliptic curve method. It is actually really simple but pretty hard to analyse. The analysis will be done in the next section. The algorithm is displayed as algorithm 2.2.1

*Remark 2.2.1.* To make the algorithm terminate there must be some upper bound on the number of times we wish to allow a new curve to be picked but clearly if it terminate it will produce a non-trivial divisor in  $n$ .

---

**Algorithm 2.2.1** ECM (Lentra's original algorithm)

---

**Input:**  $n \in \mathbb{Z}/n\mathbb{Z}$ ,  $n > 0$  with  $\gcd(6, n) = 1$  and not a perfect power.

**Output:** Factor in  $n$ .

```

(1) Pick bound  $B_1$ 
(2) Find pseudo elliptic curve  $E = E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  and a point  $(x, y) \in E$ :
 $x, y, a \in_R [0, n-1]$ 
 $b := (y^2 - x^3 - ax) \bmod n$ 
 $g := \gcd(4a^3 + 27b^2, n)$ 
if  $g == n$  then
    Go to (1)
end if
if  $g > 1$  then
    return  $g$ 
end if
Pick  $E = E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  and  $P = (x, y)$ 
(3) Prime power multipliers:
Compute list of primes  $\{p_1, p_2, \dots, p_{\pi(B_1)}\}$ 
for  $i = 1 \rightarrow \pi(B_1)$  do
    Find largest integer  $a_i$  such that  $p_i^{a_i} \leq B_1$ 
    for  $j = 1 \rightarrow a_i$  do
         $P = [p_i]P$  using the partial addition. If the addition fails then if  $\gcd(d, n) \neq n$  one
        Return  $\gcd(d, n)$  where  $d$  is the addition-slope denominator which do not have an
        inverse in  $\mathbb{Z}/n\mathbb{Z}$ .
    end for
end for
(4) Failure
Go to (2) or increment  $B_1$ 

```

---

We now add some additional notes to each block in the ECM algorithm.

- (1) This bound is really an experimental thing which must be tunable. Optimally it depends on the least prime factor in  $n$  which a priori is unknown. Therefore one must choose a bound and adjust it with respect to the practical behaviour of the algorithm.
- (2) Here we pick the pseudo elliptic curve which we will be working over. The notation  $\in_R$  means that we pick the elements out randomly (with a uniform distribution). There is a slight change (depending on  $n$ ) that we pick  $x, y$  and  $a$  such that we do not define an pseudo elliptic curve. This is checked with  $\gcd(4a^3 + 27b^2, n)$ .
- (3) What we do here is to compute  $[k]P$  for a  $k$  that is chosen to consist of a lot of small primes and powers of these. Explicitly we pick  $k = \prod_{i=1}^{\pi(B_1)} p_i^{a_i}$  where  $p_i$  and  $a_i$  are as described in the algorithm.
- (4) If the addition does not fail we pick a new curve. There is an extension at this point which increase the chances of success. This is called *The second stage* and will be described in section 2.4.

A lot of work has been put into optimizing the original algorithm proposed by Lenstra. Optimisations such as: The choice of curve, the choice of elliptic curve model and coordinate system, the choice of  $k$  and how to compute it (addition chains) and a second stage. In chapter 3 and 4 we will be optimizing the basic ECM using Edwards curves including some of the ideas just mentioned.

## 2.3 Complexity

ECM is a probabilistic algorithm and only a heuristic complexity estimate exists but which in turn may be made rigorous except for one unproven conjecture concerning the smoothness distribution in the Hasse interval. In this section we give an estimate of the running time of ECM but with some simplifications to make the analysis easier.

We must first settle the obvious question; why do ECM work? We give a sufficient condition.

**Lemma 2.3.1.** *Let  $n$  be composite with  $\gcd(6, n) = 1$  and not a perfect power. Pick  $a, x, y \in \mathbb{Z}/n\mathbb{Z}$  random and put  $b = y^2 - x^3 - ax \pmod n$ ,  $Q = (x, y)$ . Suppose  $\gcd(4a^3 + 27b^2, n) = 1$  then  $E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  is well defined. Also suppose that  $p$  is a prime dividing  $n$ . If  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  is  $B_1$ -power smooth ( $B_1$  is the bound in algorithm 2.2.1) we have*

$$[k]Q_p = \mathcal{O} \text{ on } E_{W,[a]_p,[b]_p}(\mathbb{F}_p) \quad (2.1)$$

where  $k = \prod_{i < \pi(B_1)} p_i^{a_i}$  with  $a_i$  maximal such that  $p_i^{a_i} \leq B_1$ .

*Proof.* Put  $E = E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$ . Since  $|E|$  is  $B_1$ -power smooth we have  $|E_p| \mid k$  and hence  $\exists \delta$  such that  $|E| \cdot \delta = k$ . This give

$$[k]Q_p = [|E|\delta]Q_p = [\delta]([|E|]Q_p) = [\delta]\mathcal{O} = \mathcal{O}.$$

□

Observe  $[k]Q = \mathcal{O}$  over  $E$  in particular happens  $|E|$  divides  $k$  i.e. that  $|E|$  is  $B_1$ -power smooth.

**Proposition 2.3.2.** *Let the situation be as in lemma 2.3.1. Assume  $[k]Q_q \neq \mathcal{O}$  on  $E_{W,[a]_q,[b]_q}(\mathbb{F}_q)$  for a prime dividing  $n$ . Then we have a factor in  $n$ .*

*Proof.* Assume for the sake of a contradiction that  $[k]Q$  is well defined over  $E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$ . If  $[k]Q = \mathcal{O}$  then by lemma 2.1.5  $[k]Q_q = ([k]Q)_q = \mathcal{O}_q = \mathcal{O}$  contradicting our assumption. If  $[k]Q \neq \mathcal{O}$  then  $[k]Q = (x, y)$  for some  $x, y \in \mathbb{Z}/n\mathbb{Z}$ . These satisfy  $y^2 = x^3 + ax + b$ . Reducing module  $p$  we obtain two points which satisfy the same equation, hence  $[k]Q_p = (x, y)_p \neq \mathcal{O}$ . A contradiction by lemma 2.3.1. □

The above essentially says that when considering all primes dividing  $n$ , if there is at least one pair  $(p, q)$  of divisors of  $n$  such that the curve order when reducing  $p$  is  $B_1$ -power smooth and the curve order reducing  $q$  is not  $B_1$ -power smooth we will discover a factor in  $n$ . Since the curve order of all good reductions is restricted by theorem 1.3.4 the possibility that all prime factors of  $n$  will have  $B_1$ -power smooth reduction is small if  $B_1$  is chosen appropriate. If more than one prime divisor of  $n$  has  $B_1$ -power smooth reduction we will probably not find a prime divisor but some composite divisor of  $n$ .

The complexity is ruled by the number of curves we must use and how long time each curve takes to process. We begin with an estimate of the number of curves we may possibly use.

What corollary 2.3.2 and the discussion above shows is that the lowest (2.3.2 gives a sufficient condition) change of success with ECM depends on the smoothness distribution of the elliptic curves  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  for primes  $p$  dividing  $n$ . Let  $p$  be the smallest prime divisor of  $n$ . We make the assumption that the likelihood of the events in proposition 2.3.2 is dominated by the event that  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  is  $B_1$ -power smooth. We also make the assumption that being  $B_1$ -power smooth is the same as being  $B_1$ -smooth, because only primes below  $\sqrt{B_1}$  occur with a power different from 1.

Let  $\text{prob}(B_1)$  denote the probability of success in algorithm 2.2.1 using the bound  $B_1$  hence we need approximate  $\frac{1}{\text{prob}(B_1)}$  curves to find a factor in  $n$ . By the above simplifications  $\text{prob}(B_1)$  equals the probability that  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  is  $B_1$ -smooth. We now assume that the order of  $E_{W,[a]_p,[b]_p}(\mathbb{F}_p)$  is uniformly distributed in the Hasse interval (By a theorem of Deuring [10] p. 334 all integers in the Hasse interval actually corresponds to at least one elliptic curve).

Now lets look at the cost for one curve in ECM. The primary work is done in the two for-loops we therefore neglect the other costs. For each  $p_i$  we need to make  $p_i^{a_i}$  elliptic curve operations costing about  $\ln(p_i^{a_i})$  since we exponentiate. Notice this is  $\leq \ln(B_1)$  since  $p_i^{a_i} \leq B_1$ . The number of primes up to  $B_1$  is approximately  $\pi(B_1) \approx \frac{B_1}{\ln(B_1)}$ . Therefore the cost for the two loops is around  $\sum_{i=1}^{\pi(B_1)} \ln(p_i) \leq \sum_{i=1}^{\pi(B_1)} \ln(B_1) = \pi(B_1) \ln(B_1) \approx \frac{B_1}{\ln(B_1)} \ln(B_1) = B_1$ . Hence the overall work is approximate  $\frac{B_1}{\text{prob}(B_1)}$ .

To minimize the estimated running time, the number  $B_1$  should be chosen such that  $\frac{B_1}{\text{prob}(B_1)}$  is minimal. To proceed we need a conjecture. Define  $L(x) = e^{\sqrt{\ln x \ln \ln x}}$  then the hope is

**Conjecture 2.3.3.** *Let  $\alpha$  be a real number. Then the probability that a random positive integer  $s \in [x + 1 - 2\sqrt{x}, x + 1 + 2\sqrt{x}]$  has all its prime factors  $\leq L(x)^\alpha$  is  $L(x)^{-\frac{1}{2\alpha} + o(1)}$  for  $x \rightarrow \infty$*

For  $x = p$  and the discussion above, conjecture 2.3.3 implies  $\text{prob}(L(p)^\alpha) = L(p)^{-\frac{1}{2\alpha} + o(1)}$  for  $p \rightarrow \infty$ . Put  $B_1 = L(p)^\alpha$ . Then  $\frac{B_1}{\text{prob}(B_1)} = \frac{L(p)^\alpha}{L(p)^{-\frac{1}{2\alpha} + o(1)}} = L(p)^{\frac{1}{2\alpha} + \alpha + o(1)}$  for  $p \rightarrow \infty$ .



We must minimize  $a + \frac{1}{2\alpha}$  which is easy to see occur at  $\alpha = \frac{\sqrt{2}}{2}$ . Hence we should pick  $B_1 = L(p)^{\frac{\sqrt{2}}{2} + o(1)}$  and thereby obtain  $\frac{B_1}{\text{prob}(B_1)} = L(p)^{\sqrt{2} + o(1)}$ . We have given a rough review of the conjecture running time of ECM (conjecture 2.10 [17])

**Conjecture 2.3.4.** *Let  $n$  be a positive integer not divisible by 2 and 3. Let  $M(n)$  denote an upper bound for the time, measured in bit operations, that is needed to perform a single addition (EC addition of points) and let  $p$  be the smallest prime dividing  $p$ . Then the complexity of ECM algorithm 2.2.1 is*

$$O\left(e^{\sqrt{(2+o(1)) \ln p \ln \ln p}} M(n)\right)$$

*Remark 2.3.5.* By the former considerations, we need  $O\left(e^{\sqrt{\frac{1}{2} \ln p \ln \ln p}}\right)$  curves and use  $O\left(e^{\sqrt{2 \ln p \ln \ln p}}\right)$  elliptic curve additions.

Note that the running time depends on the least prime dividing  $n$ . Other known factoring algorithms such as (general)NFS and QS both have running times that depends solely on  $n$ ;  $L_n[1/3, (63/9)^{1/3}]$  and  $L_n[1/2, 1]$  respectively. Theoretically this must give an upper hand to ECM when factoring numbers which have some small prime factors.

One problem with ECM is that a priori we have no idea what  $p$  is and it is therefore hard to pick the optimal bound this is also why we must leave  $B_1$  as a configurable bound in the algorithm.

## 2.4 2. Stage

One way to really optimize the change of finding a factor using ECM is to implement a second step called the 2. stage. To see the logic in this we start by assuming  $p$  is the least divisor in  $n$  and the curve order  $|E_{W,[a]_p, [b]_p}(\mathbb{F}_p)| = |E|$  turn out not to be  $B_1$ -power smooth. Then we expect the algorithm to fail. But what if  $|E|$  is  $(B_1, B_2)$ -smooth for some reasonable (here reasonable should be thought of as that the positive difference  $B_2 - B_1$  should not be too large)  $B_2$ ? This means that we can write  $|E| = q \prod_{\text{some } p_i^{a_i} \leq B_1} p_i^{a_i}$  for some prime  $q$  with  $B_1 < q \leq B_2$  and  $q \neq p_i$  for all  $i$ . The extra prime  $q$  is the reason that  $k$  did not become a multiple of the curve order  $|E|$ .

Because ECM failed we are in the possession of a point  $Q$  satisfying  $Q = \left[\prod_{p_i^{a_i} \leq B_1} p_i^{a_i}\right] P$  where  $P$  is the initial point in ECM. Let  $\{q_0, q_1, \dots, q_s\}$  be the primes from  $B_1$  to  $B_2$  and define  $\Delta_i = q_{i+1} - q_i$  for  $i = 0, 1, \dots, s-1$ . Then the 2. stage idea in ECM is to check the points

$$[q_0]Q, \quad [q_0 + \Delta_0]Q, \quad [q_0 + \Delta_0 + \Delta_1]Q, \quad \dots, \quad [q_0 + \Delta_0 + \Delta_1 + \dots + \Delta_{s-1}]Q \quad (2.2)$$

Note that since  $q$  from before is a prime between  $B_1$  and  $B_2$  we will actually catch it here; say  $q = q_i$  then  $[q_i]Q = [q_0 + \Delta_0 + \dots + \Delta_{i-1}]Q$ .

The crucial observation to make is that we use almost no work per prime. Say we want to calculate

$$[q_0 + \Delta_0 + \Delta_1 + \cdots + \Delta_i]Q = [q_0 + \Delta_0 + \cdots + \Delta_{i-1}]Q + [\Delta_i]Q.$$

Beforehand we have computed  $[q_0 + \Delta_0 + \cdots + \Delta_{i-1}]Q$  and saved it in an auxiliary register  $R$ . We then have to calculate  $R + [\Delta_i]Q = R + [q_{i+1} - q_i]Q$ . Since  $q_i$  and  $q_{i+1}$  are two consecutive primes, their difference is not that large.

A more efficient way to do this is to pre compute a table  $T$  with  $R_1 = [2]Q$ ,  $R_2 = [2 \cdot 2]Q$ , ...,  $R_d = [2 \cdot d]Q$  where  $d$  is the largest integer such that  $2d \leq \xi$ , where  $\xi$  is some limit, see section 4.2. Say that we again would like to calculate

$$[q_0 + \Delta_0 + \Delta_1 + \cdots + \Delta_i]Q = [q_0 + \Delta_0 + \cdots + \Delta_{i-1}]Q + [\Delta_i]Q.$$

Again  $R$  contains  $[q_0 + \Delta_0 + \cdots + \Delta_{i-1}]Q$  and we need to compute  $R + [\Delta_i]Q$ . But  $\Delta_i = q_{i+1} - q_i$  and since both  $q_{i+1}$  and  $q_i$  are odd positive integers their difference is even. Hence we may find some  $\delta$  such that  $\Delta_i = 2 \cdot \delta$ . which imply  $[\Delta_i]Q = [2 \cdot \delta]Q = R_\delta$  where  $R_\delta$  is a precomputed element from our table  $T$ . This means we only need to compute  $R + R_\delta$ , only *one* EC-operation. That is, with the precomputed table we need only one EC-operation per prime and since the table can be computed efficiently, this method has a far better performance than the first. One downside is that the method requires more memory, but not much. Implementation of the latter version is discussed in section 4.2.

## CHAPTER 3

# Edwards curves

In [13] Edwards gave a new normal form for elliptic curves. He showed that every elliptic curve over a field  $\mathbb{F}$  of characteristic not 2, can be written as  $x^2 + y^2 = c^2 + c^2x^2y^2$  over a finite extension field of  $\mathbb{F}$ . In [5] Bernstein and Lange generalized Edwards notion to  $x^2 + y^2 = c^2(1 + dx^2y^2)$  to allow more elliptic curves to be written in Edwards form without an extension of the underlying field.

In this chapter  $\mathbb{F}$  denotes a field with  $\text{char}(\mathbb{F}) \neq 2$  if nothing else is stated. It is possible to define Edwards curves over binary fields (see [11]) which in particular is interesting for implementing elliptic cryptography using Edwards curves on chips and smart cards.

### 3.1 Edwards curves

We now define Edwards cruves. Consider the curve

$$\bar{x}^2 + \bar{y}^2 = \bar{c}^2(1 + \bar{d}\bar{x}^2\bar{y}^2), \quad \bar{c} \neq 0.$$

This curve is isomorphic to

$$x^2 + y^2 = 1 + dx^2y^2, \quad d = \bar{d}\bar{c}^4,$$

which follows from the map  $(\bar{x}, \bar{y}) \mapsto (\bar{c}x, \bar{c}y)$ .

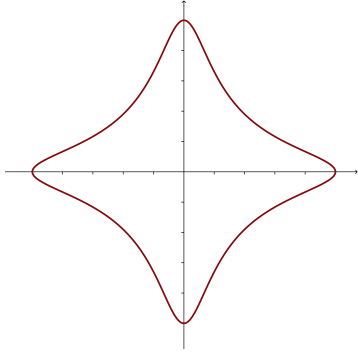
**Definition 3.1.1.** Let  $d \in \mathbb{F} \setminus \{0, 1\}$ . An Edwards curve over  $\mathbb{F}$  is a curve on the from

$$x^2 + y^2 = 1 + dx^2y^2 \tag{3.1}$$

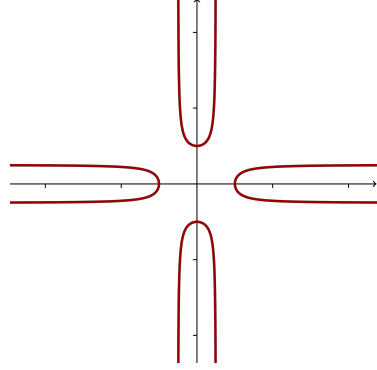
with  $d \in \mathbb{F} \setminus \{0, 1\}$ . We denote the Edwards curve by  $E_{E,d}$ .

Both Euler and Gauss has worked on the special case  $d = -1$ . In [14] Gauss presented addition formulas for that particular case;  $(s, c) + (s', c') = \left( \frac{sc' + s'c}{1 - ss'cc'}, \frac{cc' - ss'}{1 + ss'cc'} \right)$ . With his choice of  $s$  and  $c$  he probably hinted to the connection with the addition laws for sine and cosine (look at the numerators and try to remember how the additions formulas for sine and cosine looks like). For more on this analogy see remark 3.2.2.

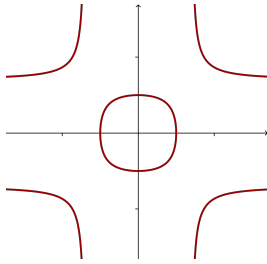
*Remark 3.1.2.* The reason for not allowing  $d = 0$  or  $d = 1$  is the following. If  $d = 0$  then  $x^2 + y^2 = 1$  which is a genus 0 curve, not an elliptic curve (Generally an elliptic curve is a projective non singular genus 1 curve. This definition is equivalent with the one given in definition 3.1.1 by the Riemann-Roch theorem). If  $d = 1$  then  $0 = 1 + x^2y^2 - x^2 - y^2 = (y^2 - 1)(x^2 - 1)$  which again does not describe an elliptic curve.



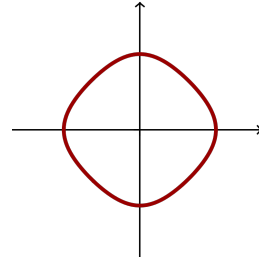
**Figure 3.1:** Edwards curve  $E_{E,-30}$  over  $\mathbb{R}$ .



**Figure 3.2:** Edwards curve  $E_{E,4}$  over  $\mathbb{R}$ .



**Figure 3.3:** Edwards curve  $E_{E,\frac{1}{2}}$  over  $\mathbb{R}$ .



**Figure 3.4:** Edwards curve  $E_{E,-1}$  over  $\mathbb{R}$ .

In this thesis we are interested in using Edwards curves in connection with ECM. To do this properly we must know several things: We need a lot of elliptic curves that can be written in Edwards form over the original field, addition must in some sense correspond to addition on the original elliptic curve and arithmetic on Edwards cannot be too bad otherwise it would be productive to switch to Edwards form (when one wants to be faster).

We need the notion of birationally equivalent curves.

**Definition 3.1.3.** Let  $V$  and  $V'$  be two curves.  $V$  and  $V'$  are *birationally equivalent* if there exists two rational maps  $\phi : V \rightarrow V'$  and  $\phi' : V' \rightarrow V$  such that  $\phi \circ \phi' = \text{id}$  and  $\phi' \circ \phi = \text{id}$  for all but finitely many points.

First we need a connection between Edwards curves and Weierstrass curves. In the original article [5] Bernstein and Lange used a possible twist and an assumption of existence of an unique point of order 2 on the elliptic curve, but these are redundant. The proof is completely constructive.

**Theorem 3.1.4.** *Let  $E$  be an elliptic curve over  $\mathbb{F}$ . Assume the group  $E(\mathbb{F})$  has an element of order 4. Then  $E$  is birationally equivalent to an Edwards curve.*

*Proof.* Assume  $E$  has a point of order 4. Let this point be  $P = (l, r)$  and  $r \neq 0$  otherwise  $P$  would have order 2. Since  $\text{char}(\mathbb{F}) \neq 2$  we may assume  $E$  is on the form  $v^2 = u^3 + a_2u^2 + a_4u + a_6$ . Since  $P$  has order 4,  $[2P] = (l', r')$  must have order 2 implying  $(l', r') = (l', -r')$  and so  $r' = 0$ . We now move this point to origo. Hence WLOG we assume  $l' = 0$  i.e.  $[2]P = (0, 0)$  and  $a_6 = 0$ ; in the general case make the transformation  $\bar{l}' + l' = u$ . If  $l = 0$  we would have  $r^2 = 0^3 + a_20^2 + a_40 = 0$  contradicting  $r \neq 0$  hence  $l \neq 0$ .

We now express  $a_2$  and  $a_4$  in terms of  $l$  and  $r$ . By the doubling law on  $E$  (identity 1.8) and  $[2]P = (0, 0)$  we get

$$0 = \left( \frac{3l^2 + 2a_2l + a_4}{2r} \right) (l - 0) - r \Leftrightarrow 2r^2 = 3l^3 + 2a_2l^2 + a_4l.$$

Since  $P$  is also on the curve  $E$  we have the identity  $r^2 = l^3 + a_2l^2 + a_4l$ . Subtracting this identity two times from the above yield

$$0 = l^3 - a_4l \Leftrightarrow l^2 = a_4$$

because  $l \neq 0$ . We also obtain

$$a_2 = \frac{a_2l^2}{l^2} = \frac{r^2 - l^3 - a_4l}{l^2} = \frac{r^2 - 2l^3}{l^2} = \frac{r^2}{l^2} - 2l$$

Our curve  $E$  is therefore

$$v^2 = u^3 + \left( \frac{r^2}{l^2} - 2l \right) u^2 + l^2u. \quad (3.2)$$

Define  $d = 1 - 4\frac{l^3}{r^2}$ . We argue  $d \neq 0, 1$ . If  $d = 1$  then  $l^3 = 0$  contradicting  $l \neq 0$ . If  $d = 0$  then  $4l^3 = r^2$ .  $E$  then has the form

$$\begin{aligned} v^2 &= u^3 + \left( \frac{r^2}{l^2} - 2l \right) u^2 + l^2u \\ &= u^3 + \left( \frac{4l^3}{l^2} - 2l \right) u^2 + l^2u \\ &= u^3 + 2lu^2 + l^2u \\ &= u(u^2 + 2lu + l^2) \\ &= u(u + l)^2 \end{aligned}$$

implying  $E$  to be a singular curve, contradicting  $E$  being an elliptic curve.

We now define a map from  $E$  to the Edwards curve  $E_{E,d}$  by  $\varphi : (u, v) \mapsto \left( \frac{ru}{lv}, \frac{u-l}{u+l} \right)$ . We show this actually maps a point on  $E$  to  $E_{E,d}$ . Put  $x = \frac{ru}{lv}$  and  $y = \frac{u-l}{u+l}$  then we need

to show  $x^2 + y^2 = 1 + dx^2y^2$ . We calculate

$$\begin{aligned} x^2 + y^2 - 1 - dx^2y^2 &= \frac{r^2u^2}{l^2v^2} + \frac{(u-l)^2}{(u+l)^2} - 1 - \left(1 - 4\frac{l^3}{r^2}\right) \frac{r^2u^2}{l^2v^2} \frac{(u-l)^2}{(u+l)^2} \\ &= \frac{r^2u^2(u+l)^2 + l^2v^2(u-l)^2 - l^2v^2(u+l)^2 - r^2u^2(u-l)^2 + 4l^3u^2(u-l)^2}{l^2v^2(u+l)^2}. \end{aligned}$$

Observe

$$\begin{aligned} r^2u^2(u+l)^2 - r^2u^2(u-l)^2 &= 4lr^2u^3 \\ l^2v^2(u-l)^2 - l^2v^2(u+l)^2 &= -4uv^2l^3. \end{aligned}$$

This yield

$$\begin{aligned} x^2 + y^2 - 1 - dx^2y^2 &= \frac{4l^3u^2(u-l)^2 + 4lr^2u^3 - 4uv^2l^3}{l^2v^2(u+l)^2} \\ &= \frac{4l^2u^2(u-l)^2 + 4r^2u^3 - 4uv^2l^2}{lv^2(u+l)^2} \\ &= \frac{4l^2u^4 + 4l^4u^2 - 8l^3u^3 + 4r^2u^3 - 4uv^2l^2}{lv^2(u+l)^2} \\ &= \frac{4u(l^2u^3 + l^4u - 2l^3u^2 + r^2u^2 - v^2l^2)}{lv^2(u+l)^2}. \end{aligned}$$

Since  $(u, v)$  is on the curve  $E$  the point satisfy identity 3.2. After multiplying with  $l^2$  and rearranging we obtain  $-l^2v^2 + l^2u^3 - 2l^3u^2 + l^4u = -r^2u^2$ . Using this in the above calculation we get

$$x^2 + y^2 - 1 - dx^2y^2 = \frac{4u(r^2u^2 - r^2u^2)}{lv^2(u+l)^2} = 0.$$

Proving  $x^2 + y^2 = 1 + dx^2y^2$ . Exceptional points for  $\varphi$  occur when  $lv = 0$  or  $u = -l$  which clearly is only possible for finitely many points. Define a map  $\psi$  from  $E_{E,d}$  to  $E$  by  $\psi : (x, y) \mapsto \left(l\frac{1+y}{1-y}, r\frac{1+y}{x(1-y)}\right)$ . Script 1 in appendix A verify that  $\psi$  really map from  $E_{E,d}$  to  $E$ . The cases  $y = 1$  and  $x = 0$  are the exceptional cases and clearly occur for only finitely many points. Now

$$\begin{aligned} \varphi \circ \psi((u, v)) &= \left(\frac{2lu(u+l)}{2l(u+l)}, \frac{2lrvu(u+l)}{2lu(u+l)}\right) = (u, v) \\ \psi \circ \varphi((x, y)) &= \left(\frac{rl(1+y)(1-y)x}{rl(1+y)(1-y)}, \frac{2yl(1-y)}{2l(1-y)}\right) = (x, y) \end{aligned}$$

i.e.  $\varphi \circ \psi = \text{id}$  and  $\psi \circ \varphi = \text{id}$ . Hence  $E$  and  $E_{E,d}$  are birationally equivalent.  $\square$

Let  $E$  be an elliptic curve over a field  $\mathbb{F}_p$ .  $E$  is finite and by theorem 1.3.2 also abelian. By the fundamental theorem for finite abelian groups we may write  $E$  as a product of  $\mathbb{Z}/p^k\mathbb{Z}$  for some primes  $p$ . Since any elliptic curve over a field  $\mathbb{F}_p$  is either cyclic or a product of two cyclic groups (see [10] p. 322 theorem 7.1.3) we do not have the possibility that e.g.  $E = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$  hence when an elliptic curve  $E$  is divisible by 8 it has a point of order 4 and theorem 3.1.4 show that  $E$  is birationally equivalent to an Edwards curve. This might indicate that we have plenty of Edwards curves.

## 3.2 Addition on Edwards curves

We start by defining the addition law on Edwards curves.

**Definition 3.2.1.** Let  $E_{E,d}$  be an Edwards curve and  $(x_1, y_1), (x_2, y_2)$  two points on it. The Edwards addition law is given by

$$(x_1, y_1), (x_2, y_2) \mapsto \left( \frac{x_1 y_2 + y_1 x_2}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right) \quad (3.3)$$

Addition on Edwards curves will be denoted  $+$ . Noting the danger of ambiguous notation, the author trusts the reader in telling from the context whether we are adding on an Edwards curve or a Weierstrass curve.

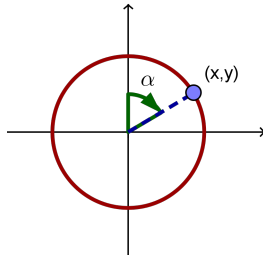
*Example 3.2.2.* One motivation for the formula (3.3) may be giving in the shape of the *clock group*. Consider the set  $\mathcal{U}$  of all tuples  $(x, y) \in \mathbb{F}^2$  that satisfy  $x^2 + y^2 = 1$ . On this set define the addition (see figure 3.6)

$$(x_1, y_1), (x_2, y_2) \mapsto (x_1 y_2 + x_2 y_1, x_1 x_2 - y_1 y_2)$$

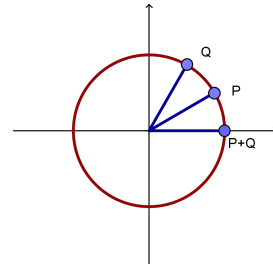
One may prove that this addition define a commutative composition making  $\mathcal{U}$  into a group with neutral element  $(0, 1)$  and with each point  $(x, y)$  having inverse  $(-x, y)$ . For  $\mathbb{F} = \mathbb{R}$  we may for any point  $(x, y)$  in  $\mathcal{U}$  draw a straight line from that point to the origin forming an angle  $\alpha$  between the positive  $y$ -axis and the line in the clockwise direction, see figure 3.5. Therefore  $(x, y) = (\sin \alpha, \cos \alpha)$  which can be done for all points in  $\mathcal{U}$ . Recall the addition laws for sine and cosine

$$\begin{aligned} \sin(\alpha_1 + \alpha_2) &= \sin \alpha_1 \cos \alpha_2 + \cos \alpha_1 \sin \alpha_2 \\ \cos(\alpha_1 + \alpha_2) &= \cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2. \end{aligned}$$

Comparing this addition with the addition in  $\mathcal{U}$  and addition on Edwards curves reveals some similarities.



**Figure 3.5:** Angle of a point in the clock group over  $\mathbb{R}$ .



**Figure 3.6:** Addition in the clock group over  $\mathbb{R}$ .  $P = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$ ,  $Q = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$  and  $P + Q = (1, 0)$ .

We will show that the addition law (3.3) satisfy what we need. In theorem 3.2.4 we prove that the Edwards addition maps (when defined) to the Edwards curve, theorem

3.2.5 shows that ordinary addition on the birationally equivalent elliptic curve in 3.1.4 corresponds to addition on the Edwards curve. Finally in theorem 3.2.8 we prove a very strong property Edwards addition possess.

*Remark 3.2.3.* The neutral element for Edwards addition is  $(0, 1)$  and the inverse for a point  $(x, y)$  is  $(-x, y)$ . An Edwards curve always has two points of order 4 namely  $(1, 0)$  and  $(-1, 0)$ . This is immediate from the calculation

$$\begin{aligned}(1, 0) + (1, 0) + (1, 0) + (1, 0) &= (0, -1) + (0, -1) = (0, 1) \\ (-1, 0) + (-1, 0) + (-1, 0) + (-1, 0) &= (0, -1) + (0, -1) = (0, 1)\end{aligned}$$

Notice that  $\{(0, 1), (0, -1), (1, 0), (-1, 0)\}$  defines a group with the Edwards addition. In general the addition formula is unified; it works for both squaring and addition.

**Theorem 3.2.4.** *Let  $E_{E,d}$  be an Edwards curve and let  $(x_1, y_1), (x_2, y_2)$  be points on  $E_{E,d}$ . Assume  $1 \pm dx_1x_2y_1y_2 \neq 0$ . Define  $x_3 = \frac{x_1y_2+y_1x_2}{1+dx_1x_2y_1y_2}$  and  $y_3 = \frac{y_1y_2-x_1x_2}{1-dx_1x_2y_1y_2}$ . Then  $(x_3, y_3)$  is a point on  $E_{E,d}$ .*

*Proof.* There is really no magic involved in this proof - only humdrum. We must prove  $x_3^2 + y_3^2 = 1 + dx_3^2y_3^2$ .

First we need an identity. Let

$$\delta = (x_1y_2 + y_1x_2)^2(1 - dx_1x_2y_1y_2)^2 + (y_1y_2 - x_1x_2)^2(1 + dx_1x_2y_1y_2)^2$$

Then one can check with e.g. Sage or Maple, the following

$$\delta = (x_1^2 + y_1^2 - (x_2^2 + y_2^2)dx_1^2y_1^2)(x_2^2 + y_2^2 - (x_1^2 + y_1^2)dx_2^2y_2^2) + d(x_1y_2 + y_1x_2)^2(y_1y_2 - x_1x_2)^2$$

Script 2 in appendix A verifies this. Since  $(x_1, y_1)$  and  $(x_2, y_2)$  are both points on  $E_{E,d}$  they satisfy

$$x_1^2 + y_1^2 = 1 + dx_1^2y_1^2 \tag{3.4}$$

$$x_2^2 + y_2^2 = 1 + dx_2^2y_2^2. \tag{3.5}$$

Multiplying (3.4) with  $dx_2^2y_2^2$  and subtracting the new identity from (3.5) yield

$$x_2^2 + y_2^2 - (x_1^2 + y_1^2)dx_2^2y_2^2 = 1 - (dx_1x_2y_1y_2)^2. \tag{3.6}$$

Similarly by multiplying (3.5) with  $dx_1^2y_1^2$  and subtraction this from (3.4) we get

$$x_1^2 + y_1^2 - (x_2^2 + y_2^2)dx_1^2y_1^2 = 1 - (dx_1x_2y_1y_2)^2. \tag{3.7}$$

When substituting (3.6) and (3.7) into the identity for  $\delta$  we obtain

$$\delta = (1 - (dx_1x_2y_1y_2)^2)^2 + d(x_1y_2 + y_1x_2)^2(y_1y_2 - x_1x_2)^2.$$



The finishing touch is obvious now:

$$\begin{aligned}
 x_3^2 + y_3^2 &= \left( \frac{x_1 y_2 + y_1 x_2}{1 + dx_1 x_2 y_1 y_2} \right)^2 + \left( \frac{y_1 y_2 - x_1 x_2}{1 - dx_1 x_2 y_1 y_2} \right)^2 \\
 &= \frac{(x_1 y_2 + y_1 x_2)^2 (1 - dx_1 x_2 y_1 y_2)^2 + (y_1 y_2 - x_1 x_2)^2 (1 + dx_1 x_2 y_1 y_2)^2}{(1 + dx_1 x_2 y_1 y_2)^2 (1 - dx_1 x_2 y_1 y_2)^2} \\
 &= \frac{\delta}{(1 + dx_1 x_2 y_1 y_2)^2 (1 - dx_1 x_2 y_1 y_2)^2} \\
 &= \frac{(1 - (dx_1 x_2 y_1 y_2)^2)^2 + d(x_1 y_2 + y_1 x_2)^2 (y_1 y_2 - x_1 x_2)^2}{(1 + dx_1 x_2 y_1 y_2)^2 (1 - dx_1 x_2 y_1 y_2)^2} \\
 &= \frac{(1 - (dx_1 x_2 y_1 y_2)^2)^2}{(1 - (dx_1 x_2 y_1 y_2)^2)^2} + d \frac{(x_1 y_2 + y_1 x_2)^2 (y_1 y_2 - x_1 x_2)^2}{(1 + dx_1 x_2 y_1 y_2)^2 (1 - dx_1 x_2 y_1 y_2)^2} \\
 &= 1 + d \left( \frac{x_1 y_2 + y_1 x_2}{1 + dx_1 x_2 y_1 y_2} \right)^2 \left( \frac{y_1 y_2 - x_1 x_2}{1 - dx_1 x_2 y_1 y_2} \right)^2 \\
 &= 1 + dx_3^2 y_3^2.
 \end{aligned}$$

□

Script 3 in appendix A also verifies theorem 3.2.4. Imagine an application where you need to calculate a series of computations on an elliptic curve say  $nP_1 + mP_2$  where  $P_1$  and  $P_2$  are points on the curve. If  $n$  and  $m$  are large, reducing the cost of addition on the elliptic curve is preferable. In section 3.3 we will see that arithmetic on Edwards curves are way faster than arithmetic on a Weierstrass curve. Actually we will see that arithmetic on Edwards curves is superior to almost all known schemes of addition on elliptic curves.

The following shows that it is possible to change to an Edwards curve if there exists a birational equivalence between an Edwards curve and the elliptic curve.

**Theorem 3.2.5.** *Assume the situation from theorem 3.2.4. Let  $E_{E,d}$  be an Edwards curve and  $E$  be the elliptic curve  $\frac{1}{1-d}v^2 = u^3 + 2\left(\frac{1+d}{1-d}\right)u^2 + u$ . For  $i = 1, 2, 3$  define*

$$P_i = \begin{cases} \mathcal{O} & (x_i, y_i) = (0, 1) \\ (0, 0) & (x_i, y_i) = (0, -1) \\ \left( \frac{1+y_i}{1-y_i}, 2\frac{1+y_i}{(1-y_i)x_i} \right) & x_i \neq 0 \end{cases} \quad (3.8)$$

where  $(x_i, y_i)$  are points on  $E_{E,d}$  and  $(x_1, y_1) + (x_2, y_2) = (x_3, y_2)$ . Then  $P_i \in E(\mathbb{F})$  and  $P_1 + P_2 = P_3$ .

*Proof.* Notice that if  $y_i = 1$  then  $x^2 + 1 = 1 + dx^2$  if and only if  $x^2(1 - d) = 0$ . Hence  $x = 0$  otherwise  $d = 1$  contradicting  $E_d$  being an Edwards curve. That is, in (3.8) we will not assign  $P_i$  with the last case when  $y_i = 1$ .

We show  $P_i \in E(\mathbb{F})$  by splitting into the three cases in (3.8). The first two are obvious. Therefore assume the last case. Put  $x_i = x$  and  $y_i = y$ . We simple calculate

$$\begin{aligned}
 & \frac{(1-y)^3(1-d)}{y+1} \left( \left( \frac{1+y}{1-y} \right)^3 + 2 \left( \frac{1+d}{1-d} \right) \left( \frac{1+y}{1-y} \right)^2 + \frac{1+y}{1-y} \right) \\
 &= (1+y)^2(1-d) + 2(1+d)(1+y)(1-y) + (1-y)^2(1-d) \\
 &= (1+2y+y^2)(1-d) + 2(1+d)(1-y^2) + (1-2y+y^2)(1-d) \\
 &= 1+2y+y^2-d-2dy-dy^2+2-2y^2+2d-2dy^2+1-2y+y^2-d+2dy-dy^2 \\
 &= 4(1-dy^2) \\
 &= 4 \left( 1 - \frac{1-x^2-y^2}{x^2} \right) \\
 &= 4 \frac{(1-y)(1+y)}{x^2}.
 \end{aligned}$$

Multiply through with  $\frac{y+1}{(1-y)^3(1-d)}$  to obtain

$$\left( \frac{1+y}{1-y} \right)^3 + 2 \left( \frac{1+d}{1-d} \right) \left( \frac{1+y}{1-y} \right)^2 + \frac{1+y}{1-y} = 4 \frac{(1+y)^2}{(1-y)^2 x^2} = \left( 2 \frac{1+y}{(1-y)x} \right)^2$$

Proving  $P_i \in E(\mathbb{F})$ .

We are left with the task of proving  $P_1 + P_2 = P_3$ . This will split into several cases.

If  $(x_1, y_1) = (0, 1)$  then  $P_1 = \mathcal{O}$  and  $(x_3, y_3) = (x_1, y_1) + (x_2, y_2) = (x_2, y_2)$ . Thus  $P_1 + P_2 = \mathcal{O} + P_2 = P_2 = P_3$ . Same arguments work if  $(x_2, y_2) = (0, 1)$ . For the rest we therefore assume  $(x_1, y_1)$  and  $(x_2, y_2)$  is not  $(0, 1)$ .

If  $(x_3, y_3) = (0, 1)$  then  $(x_1, y_1) = (-x_2, y_2)$  and  $P_3 = \mathcal{O}$ . We must show  $P_1 = -P_2$ . Suppose  $(x_1, y_1) = (0, -1)$  then  $(x_2, y_2) = (0, -1)$  and  $P_1 = (0, 0) = P_2$  so  $P_1 = -P_2$ . Symmetric if  $(x_2, y_2) = (0, -1)$ . If the latter is not the case,  $x_1, x_2 \neq 0$ . Then

$$P_1 = \left( \frac{1+y_1}{1-y_1}, 2 \frac{1+y_1}{(1-y_1)x_1} \right) = \left( \frac{1+y_2}{1-y_2}, 2 \frac{1+y_2}{(1-y_2)(-x_2)} \right) = -P_2$$

From now on assume  $(x_3, y_3) \neq (0, 1)$

If  $(x_1, y_1) = (0, -1)$  then  $P_1 = (0, 0)$  and  $(x_3, y_3) = (0, -1) + (x_2, y_2) = \left( \frac{-x_2}{1}, \frac{-y_2}{1} \right) = (-x_2, -y_2)$ . This imply  $(x_2, y_2) \neq (0, -1)$  otherwise  $(x_3, y_3) = (0, 1)$  which has been handled. Thus  $x_2 \neq 0$  and we have

$$u_2 = \frac{1+y_2}{1-y_2}, \quad v_2 = 2 \frac{1+y_2}{(1-y_2)x_2} = 2 \frac{u_2}{x_2}$$

such that  $P_2 = (u_2, v_2)$ .  $u_2$  and  $v_2$  satisfy  $\frac{1}{1-d}v_2^2 = u_2^3 + 2\frac{1+d}{1-d}u_2^2 + u_2$  (by theorem 3.2.4). Multiplying with  $\frac{1}{u_2^2}$  ( $y_2 \neq -1$  hence  $u_2 \neq 0$ ) and rearranging we get  $\frac{1}{1-d} \left( \frac{v_2}{u_2} \right)^2 - u_2 - 2\frac{1+d}{1-d} = \frac{1}{u_2}$ . Now standard addition on  $E$  give  $P_1 + P_2 = (0, 0) + (u_2, v_2) = (l_3, r_3)$  where

$$\begin{aligned}
 l_3 &= \frac{1}{1-d} \left( \frac{v_2 - 0}{u_2 - 0} \right)^2 - 2 \frac{1+d}{1-d} - u_2 - 0 = \frac{1}{u_2} \\
 r_3 &= \frac{v_2 - 0}{u_2 - 0} (0 - l_3) - 0 = -\frac{v_2}{u_2^2}
 \end{aligned}$$

Also

$$\begin{aligned} P_3 &= \left( \frac{1+y_3}{1-y_3}, 2 \frac{1+y_3}{(1-y_3)x_3} \right) = \left( \frac{1-y_2}{1+y_2}, -2 \frac{1-y_2}{(1+y_2)x_2} \right) = \left( \frac{1}{u_2}, -2 \frac{1}{u_2 x_2} \right) \\ &= \left( l_3, -2 \frac{u_2}{u_2^2 x_2} \right) = \left( l_3, -\frac{v_2}{u_2^2} \right) = (l_3, r_3) \end{aligned}$$

Thus  $P_1 + P_2 = P_3$ . If  $(x_2, y_2) = (0, -1)$  similar arguments apply.

For the rest of this proof we assume  $x_1, x_2 \neq 0$ . We can now put  $P_i = (u_i, v_i)$  with  $u_i = \frac{1+y_i}{1-y_i}$  and  $v_i = 2 \frac{u_i}{x_i}$  for  $i = 1, 2$ .

If  $(x_3, y_3) = (0, -1)$  then  $(x_1, y_1) = (x_1, y_1) + (x_2, y_2) - (x_2, y_2) = (x_3, y_3) - (x_2, y_2) = (0, -1) + (-x_2, y_2) = (x_2, -y_2)$  and  $P_3 = (0, 0)$ . With almost the same calculations as before  $u_1 = \frac{1}{u_2}$  and  $v_1 = \frac{v_2}{u_2^2}$ . As before we have

$$-P_3 + P_2 = (0, 0) + P_2 = \left( \frac{1}{u_2}, -\frac{v_2}{u_2^2} \right) = (u_1, -v_1) = P_1$$

proving  $P_1 + P_2 = P_3$ . Script 4 in appendix A verifies that  $P_1 + P_2 = P_3$  in the above case. Now we can also assume  $x_3 \neq 0$  and put  $P_3 = (u_3, v_3)$  with  $u_3 = \frac{1+y_3}{1-y_3}$  and  $v_3 = 2 \frac{u_3}{x_3}$ .

If  $P_1 = -P_2$  then  $u_1 = u_2$  and  $v_1 = -v_2$ . Thus  $x_1 = 2 \frac{u_1}{v_1} = 2 \frac{u_2}{v_2} = x_2$  and  $y_1 = \frac{u_1-1}{u_1+1} = \frac{u_2-1}{u_2+1} = -y_2$  implying  $(x_3, y_3) = (0, 1)$ . This case has already been handled. Assume from now on that  $P_1 \neq -P_2$ .

If  $u_1 = u_2$  and  $v_1 \neq -v_2$  (we assume  $P_1 \neq -P_2$ ). Then by the standard addition law we get  $l_3 = \frac{1}{1-d} m^2 - 2 \frac{1+d}{1-d} - 2u_1$ ,  $r_3 = m(u_1 - l_3) - v_1$ ,  $m = \frac{3u_1^2 + 4((1+d)/(1-d))u_1 + 1}{(2/(1-d))v_1}$  where  $(x_1, y_1) + (x_2, y_2) = (l_3, r_3)$ . As before it is (with a lot of paper) straight forward to verify  $(l_3, r_3) = (u_3, v_3)$ .

Last(!) case: If  $u_1 \neq u_2$ . Again using the standard addition law we obtain  $m = \frac{v_2 - v_1}{u_2 - u_1}$ ,  $l_3 = \frac{1}{1-d} m^2 - 2 \frac{1+d}{1-d} - u_1 - u_2$  and  $r_3 = m(u_1 - l_3) - v_1$  with  $(u_1, v_1) + (u_2, v_2) = (l_3, r_3)$ . One can again check that  $(l_3, r_3) = (u_3, v_3)$ . This and the latter case is checked in [12].

We are done!  $\square$

*Remark 3.2.6.* In the proof above we used several times that the only points on an Edwards curve with zero  $x$ -coordinate are  $(0, 1)$  and  $(0, -1)$ . This is immediate if we substitute  $x = 0$  in the defining equation of an Edwards curve:  $1 = y^2$  so  $0 = (1 - y)(1 + y)$ . We also used a generalized form of addition; an elliptic curve on the form  $By^2 = x^3 + cx^2 + x$  is called a Montgomery curve. Addition formulas for this kind of curve differ a little with respect to the usual addition formulas for addition on elliptic curves. Namely (1.7) change to  $x_3 = Bm^2 - c - x_1 - x_2$ , (1.8) stays as it is and

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - y_2} & x_2 \neq x_1 \\ \frac{3x_1^2 + 2cx_1 + a}{2By_1} & x_2 = x_1 \end{cases}$$

We can now show that the condition in theorem 3.1.4 is not only a sufficient condition but a necessary condition.

**Theorem 3.2.7.** *Let  $E$  be an elliptic curve over  $\mathbb{F}$ . If  $E$  is birationally equivalent to an Edwards curve, the group  $E(\mathbb{F})$  has an element of order 4.*

*Proof.* Assume  $E$  is birationally equivalent over  $\mathbb{F}$  to an Edwards curve  $E_{E,d}$ . With the rational map  $(x, y) \mapsto \left(\frac{1+y}{1-y}, 2\frac{1+y}{(1-y)x}\right)$  we map points on  $E_{E,d}$  to the elliptic curve  $E' : \frac{1}{1-d}v^2 = u^3 + 2\frac{1+d}{1-d}u^2 + u$ . The inverse map is  $(u, v) \mapsto \left(2\frac{u}{v}, \frac{u-1}{u+1}\right)$ . This gives a birational equivalence between  $E_{E,d}$  and  $E'$  thus also a birational equivalence between  $E$  and  $E'$ . In theorem 3.2.5 we saw that addition on  $E_{E,d}$  corresponds to addition on the elliptic curve  $E'$ . Since the point  $(1, 0)$  on  $E_{E,d}$  has order 4 the corresponding point on  $E'$  also has order 4 and  $E$  must have a point of order 4.  $\square$

It turns out that in some cases the addition formula on Edwards curves is even complete i.e works for all input; in this case any addition on the Edwards curve is without risk of failure.

**Theorem 3.2.8.** *Let  $E_{E,d}$  be an Edwards curve. Assume  $d$  is not a square. Let  $(x_1, y_1)$  and  $(x_2, y_2)$  be points on  $E_{E,d}$ . Then  $1 \pm dx_1x_2y_1y_2 \neq 0$ .*

*Proof.* This will be a proof by contradiction. Let  $\delta = dx_1x_2y_1y_2$  and suppose for the sake of contradiction that  $\delta = \pm 1$ . It follows  $x_1, x_2, y_1, y_2 \neq 0$ . Then

$$\begin{aligned} (x_1 + \delta y_1)^2 &= x_1^2 + \delta^2 y_1^2 + 2x_1y_1\delta = x_1^2 + (-1)^2 y_1^2 + 2x_1y_1\delta \\ &= 1 + dx_1^2 y_1^2 + 2x_1y_1\delta = \delta^2 + dx_1^2 y_1^2 + 2x_1y_1\delta \\ &= dx_1^2 y_1^2 + d^2 x_1^2 x_2^2 y_1^2 y_2^2 + 2x_1y_1\delta = dx_1^2 y_1^2 (1 + dx_2^2 y_2^2) + 2x_1y_1\delta \\ &= dx_1^2 y_1^2 (x_2^2 + y_2^2) + 2x_1y_1\delta = dx_1^2 y_1^2 (x_2^2 + y_2^2) + 2x_1y_1 d^2 x_1^2 x_2^2 y_1^2 y_2^2 \\ &= dx_1^2 y_1^2 (x_2^2 + 2x_2y_2 + y_2^2) = dx_1^2 y_1^2 (x_2 + y_2)^2. \end{aligned}$$

If  $x_2 + y_2 \neq 0$  (recall  $x_1, y_1 \neq 0$ ) then  $d = \left(\frac{x_1 + \delta y_1}{x_1 y_1 (x_2 + y_2)}\right)^2$  contradicting  $d$  being a non-square. One can do similar calculations as above and get  $(x_1 - \delta y_1)^2 = dx_1^2 y_1^2 (x_2 - y_2)^2$ . If  $x_2 - y_2 \neq 0$  then  $d = \left(\frac{x_1 - \delta y_1}{x_1 y_1 (x_2 - y_2)}\right)^2$  contradiction. Hence  $x_2 + y_2 = 0$  and  $x_2 - y_2 = 0$ . We quickly spot  $0 = (x_2 + y_2) + (x_2 - y_2) = 2x_2$  and  $0 = 2y_2$ . Since  $\text{char}(\mathbb{F}) \neq 2$  we get  $x_2, y_2 = 0$  our final contradiction.  $\square$

We will not be using this property in the implementation since we go for speed not for this simplicity or as we shall see now, security. Consider an implementation of some cryptographic scheme using double-scaler-multiplication on elliptic curves i.e  $[n]P + [m]Q$ . The usual addition formulas for the Weierstrass model has several exceptional cases and an irritating distinction between addition and doubling; you can not double a point as  $P + P$  and use the addition formula. The plethora of cases has caused a variety of problems, in particular, when in [16] Paul Kocher described a timing attack of several widely used crypto systems and laid the ground for side-channel attacks. In recent times timing attacks

are exploiting the property that addition and doubling are different operations enabling that the involved secrets could be revealed from only a single execution of the used algorithm. Several countermeasures such as adding dummy operations or rewriting formulas are known and used, but the complete addition formula for Edwards curves solves this in one sweep move.

### 3.3 Efficient operations on Edwards curves

In this section we introduce efficient formulas for computing on Edwards curves and compare these formulas to other popular schemes. Efficiency of the operations are ordered by the number of operations in the underlying field. In particular we count; number of multiplications **M** (each costing **M**), number of squarings **S** (each costing **S**), multiplication by  $d$  costing **D** (each costing **D**) and number of additions/subtractions **A** (each costing **A**). We do not keep track of inversions since we avoid these by using projective coordinates.

The reader may wonder why we keep a separate tally of squaring when a squaring is essentially a multiplication. It is true that squaring and multiplication both has the same complexity, but multiplication algorithms normally simplify when inputting a square which will speed up squaring by a constant factor.

The reason for avoiding inversions is the known fact that inversions is inefficient compared to doing multiplications or additions. Of course the ratio inversion/multiplication differ depending on which platform and hardware being used, but generally one should expect a factor that is quite high. For instance in [12] Bernstein and Lange use  $\mathbf{I}/\mathbf{M} = 100$ .

To avoid inversions when computing on Edwards curves we homogenize the Edwards curve to  $(x^2 + y^2)z^2 = z^4 + dx^2y^2$ . A point  $[x, y, z]$  corresponds to the affine point  $(\frac{x}{z}, \frac{y}{z})$  for  $z \neq 0$ . Putting the two points  $(\frac{x_1}{z_1}, \frac{y_1}{z_1})$  and  $(\frac{x_2}{z_2}, \frac{y_2}{z_2})$  into the addition formula for the Edwards curve yields

$$\begin{aligned} \frac{\frac{x_1y_2 + x_2y_1}{z_1z_2}}{1 + \frac{dx_1x_2y_1y_2}{z_1^2z_2^2}} &= \frac{(x_1y_2 + x_2y_1)z_1z_2}{z_1^2z_2^2 + dx_1x_2y_1y_2} = \frac{(x_1y_2 + x_2y_1)z_1z_2(z_1^2z_2^2 - dx_1x_2y_1y_2)}{(z_1^2z_2^2)^2 - (dx_1x_2y_1y_2)^2} \\ \frac{\frac{y_1y_2 - x_1x_2}{z_1z_2}}{1 - \frac{dx_1x_2y_1y_2}{z_1^2z_2^2}} &= \frac{(y_1y_2 - x_1x_2)z_1z_2(z_1^2z_2^2 + dx_1x_2y_1y_2)}{(z_1^2z_2^2)^2 - (dx_1x_2y_1y_2)^2}. \end{aligned}$$

Put  $\delta = dx_1x_2y_1y_2$ . Addition on the projective form of the Edwards curve is (with  $z_1, z_2 \neq 0$ )

$$\begin{aligned} ([x_1, y_1, z_1], [x_2, y_2, z_2]) &\xrightarrow{ADD} \\ &\left( (x_1y_2 + x_2y_1)z_1z_2(z_1^2z_2^2 - \delta), (y_1y_2 - x_1x_2)z_1z_2(z_1^2z_2^2 + \delta), (z_1^2z_2^2)^2 - \delta^2 \right). \end{aligned}$$

The neutral element is  $[0, 1, 1]$  and if  $[x, y, z]$  is a point on the homogenized curve then  $[-x, y, z]$  is the inverse. Rewriting  $x_1y_2 + x_2y_1 = (x_1 + x_2)(y_1 + y_2) - x_1x_2 - y_1y_2$  and

exploiting different common sub expressions one may count  $10\mathbf{M}+1\mathbf{S}+1\mathbf{D}+7\mathbf{A}$  for one addition.

In some cases we may save  $1\mathbf{M}$  namely in a mixed addition; an additions with  $z_2 = 1$ . In this case we do not need to do the calculation  $z_1 z_2$ . The count for mixed addition then reads  $9\mathbf{M}+1\mathbf{S}+1\mathbf{D}+7\mathbf{A}$  for one mixed addition.

Doublings are even faster. Using the ordinary addition on Edwards curves with two equal points yield

$$\left( \frac{xy + yx}{1 + dx^2y^2}, \frac{y^2 - y^2}{1 - dx^2y^2} \right) = \left( \frac{2xy}{x^2 + y^2}, \frac{y^2 - x^2}{2 - x^2 - y^2} \right) = \left( \frac{2xy}{x^2 + y^2}, \frac{x^2 - y^2}{x^2 + y^2 - 2} \right)$$

using that  $(x, y)$  satisfy  $x^2 + y^2 = 1 + dx^2y^2$ . In projective coordinates the doubling formula is ( $z \neq 0$ )

$$([x, y, z]) \xrightarrow{DUP} (2xy(2z^2 - (x^2 + y^2)), (x^2 - y^2)(x^2 + y^2), (x^2 + y^2)(x^2 + y^2 - 2z^2)).$$

Again rewriting  $2xy = (x + y)^2 - x^2 - y^2$  and exploiting common sub expressions we count  $3\mathbf{M}+4\mathbf{S}+6\mathbf{A}$  for one duplication. Notice this formula is independent of  $d$ .

Register allocations for the above formulas are given in section 4.4 where one easily read of the counts stated above.

In [5] Bernstein and Lange among other things, did a survey on the efficiency of several additions schemes in the literature. They gathered addition and squaring counts for different curves and coordinates systems and compared them. The results presented in their article clearly shows that Edwards curves provide one of the fastest addition and doubling known in the literature. Using Edwards curves Bernstein and Lange has discovered even faster formulas. In [6] they presented *inverted edwards coordinates*; a point  $[x, y, z]$  with  $xyz \neq 0$  on the projective curve  $(x^2 + y^2)z^2 = x^2y^2 + dz^4$  corresponds to  $\left(\frac{z}{x}, \frac{z}{y}\right)$ . They report costs of addition and doubling as:  $9\mathbf{M}+1\mathbf{S}+1\mathbf{D}+1\mathbf{A}$  and  $3\mathbf{M}+4\mathbf{S}+1\mathbf{D}+1\mathbf{A}$  respectively. Compared to the previous formulas we trade  $1\mathbf{M}$  in the addition formula with a  $1\mathbf{D}$  in the doubling formula.

# ECM using Edwards curves

This chapter serves as documentation for the implementation of ECM using Edwards curves the author made in connection with this thesis. We describe which techniques and speed ups we use. We also include a section of experiments/statistics that should provide an overview of the efficiency of the implementation. In the following the program will be referred to as *EECM\_Torben*.

Generally any implementation of ECM that needs to be fast are essentially faced with the two following algorithmic challenges: Fast modular arithmetic and efficient curve operations. Efficient modular arithmetic will be provided by the *BigInteger library* in Java and is therefore not a class implemented by the author. This is discussed in section 4.3. Efficient curve operations is provided by (as you may have guessed) arithmetic on Edwards curves. How Edwards curves relate to the elliptic curve factorization method is discussed in 4.1.

*EECM\_Torben* has the same general structure as algorithm 2.2.1 but with some major differences.

- *EECM\_Torben* use Edwards curves instead of Weierstrass curves, and therefore instead of picking a curve  $E_{W,a,b}(\mathbb{Z}/n\mathbb{Z})$  we pick an Edwards curve  $E_{E,d}$  and calculate over  $\mathbb{Z}/n\mathbb{Z}$ .
- When we have computed  $[s]P$  we check the gcd between the  $x$ -coordinate and  $n$ . This is because the neutral element on the Edwards curve is  $(0, 1)$  (in projective  $[0, 1, 1]$ ). At a first glance the addition formula in 3.3 seems to allow a factor of  $n$  not to accumulate in the  $x$ -coordinate i.e. if one time during the calculation of  $[s]P$  we find a factor of  $n$  it is not clear that this factor remain present in the  $x$ -coordinate. But by inspecting the source code of *EECM\_Torben* one sees that factors indeed do accumulate, because we use the  $x$ -coordinate from the newly computed point as  $x_2$  in the next iteration (see also algorithm 4.4.3). In the ordinary ECM algorithm one would check the  $z$ -coordinate when using projective Weierstrass coordinates.

- We use a second stage method to improve the probability of finding a factor. This implementation is discussed in section 4.2.
- The calculation of  $[s]P$  is done using a NAF representation of the primes. This is discussed in section 4.4.
- We do not check for  $\gcd(6, n)$  and  $n$  not being a perfect power. In practice EECM\_Torben find primes in  $n$  whether or not it is a perfect power and since checking for  $n$  being a perfect power is not extremely cheap we neglect this. Instead of checking the gcd between 6 and  $n$  we instead do a trial division for all primes from 2 to 1000. these primes has been hard coded into EECM\_Torben and it is therefore really fast to do this check and is much more efficient than only trying to find a factor of 2 or 3 in  $n$ . Actually the bound 1000 may be varied depending of  $n$  but it all comes down to how much time one want to spend doing trial division versus the time it takes to execute the rest of the algorithm.

## 4.1 Using Edwards curves

There are obvious improvements from using Edwards curves. In 3.3 we saw that arithmetic on Edwards curves is more efficient than doing arithmetic on Weierstrass curves and actually the formulas is some of the most efficient known. Also the reduced curve order is certain to be divisible by 4 because of theorem 3.2.7. Heuristically this give a better chance of being smooth.

We also expect the approximate same complexity as the original ECM presented and analysed in chapter 2. This is because, reducing an Edwards curve, say  $E_{E,d}$ , over a positive integer  $n$  with a prime  $p$  dividing  $n$  would give an ordinary Edwards curve. From remark 3.2.6 we see that if EECM\_Torben hit a point with a  $x$ -coordinate that is divisible by  $p$  then it is probably the neutral element for the reduced Edwards curve by  $p$ . Theorem 3.2.5 shows that the calculations that were done on the reduced Edwards curve corresponds to an birationally equivalent elliptic curve and that we from the reduced Edwards curve has hit the neutral element on the corresponding elliptic curve. Using the same considerations that we did in section 2.3 we see that the complexity should be unchanged.

## 4.2 Stage two

Stage 2 in EECM\_Torben is implemented as described in section 2.4 (using the last version). The implementation is called the standard continuation, but is implemented with some speed up tricks.

In places where we need to add the same point to another a lot of times we normalize the point. This require  $1\mathbf{I}+2\mathbf{M}$  but if e.g.  $\mathbf{I}/\mathbf{M} = 150$  we still save multiplications as long



a we have to add the given point over approximately 150 times because a mixed addition is 1M cheaper than a dedicated addition.

We also store multiples of 2 of the point  $Q = \left[ \prod_{p_i^{a_i} \leq B_1} p_i^{a_i} \right] P$  up to some limit. This limit actually do not need to be very large! We do a little analysis. We need the precomputed points to calculate  $[p_{i+1} - p_i]Q$  i.e. we need to pre compute  $[2s]Q$  up to the biggest prime gap for primes below  $B_2$ . If  $g_n = p_{n+1} - p_n$  denotes the prime gap between the  $n$ th and  $(n+1)$ th prime then a maximal prime gap is a gap  $g_n$  such that  $g_n > g_m$  for all  $m < n$ . The distribution of the length of prime gaps is not that well understood, but there exists numerical results of maximal prime gaps for huge numbers (far greater than we need). On <http://www.trnicely.net/gaps/gaplist.html> Thomas Nicely host list of prime gaps including the maximal ones (in the lists these are indicated by an asterisk mark). Since the magnitude of primes in EECM\_Torben is bounded to approximately 1.4 billion due to memory requirements, we do not need many multiples of 2 in our second stage. If one looks in Nicely's table one see that the maximal prime gap below 1.4 billion is 320 and since we only need multiples of 2 we need at most 160 precomputed points. This saves foremost a lot compared to computing e.g.  $\frac{B_2}{2}$  multiples (which would be the naive choice) but memory requirements is also greatly reduced.

There exists several improvements to the 2. stage. We give a high level description of some of these. One approach is: Take two integers  $\tau$  and  $\sigma$  and compute  $[\tau]Q + [\sigma]Q$  where  $Q$  is the point after the first stage. Set  $[\tau]Q = (x_\tau, y_\tau)$  and  $[\sigma]Q = (x_\sigma, y_\sigma)$ . If  $[\tau]Q + [\sigma]Q = \mathcal{O} \pmod{p}$  then  $x_\tau \equiv x_\sigma \pmod{p}$  and  $\gcd(x_\tau - x_\sigma, n) > 1$ . The way  $\tau$  and  $\sigma$  is picked is how different continuations of this form differ. The birthday paradox form is to pick  $\sigma \in T$  and  $\tau \in S$  in two set  $T, S \subset \mathbb{N}$ .  $T$  and  $S$  are either picked out random or as geometric progressions. One then hope that all combinations  $\tau + \sigma$  hits all primes in  $[B_1, B_2]$ . In addition one might hit larger primes. This approach may be optimized using fast polynomial arithmetic. A popular choice is to use a Fast Fourier Transformation. An extension of this kind is called a FFT extension. The polynomial arithmetic is applied, in different ways to obtain  $\prod_{\tau \in S} \prod_{\sigma \in T} (x_\sigma - x_\tau) \pmod{n}$  in a fast way. For more 2. stage continuations consult [24].

## 4.3 Modular arithmetic

As mentioned, the modular arithmetic is provided by the BigInteger library from Java's standard library. There is two main reason for choosing this option. The first is while implementing a new big number library is doable then since the BigInteger library has been developed trough several years, creating a library that is faster must be considered a low probability event and something which would take a lot of time. The other reason is that it is not the scope of this thesis to develop a new big number library. We do not

claim that a development of an efficient dedicated ring library is not worth the effort. It will without doubt speed up computations because the BigInteger library we use is a big number library where modular arithmetic has been added subsequently.

A reference to the methods used from the BigInteger library inside EECM\_Torben is <http://docs.oracle.com/javase/6/docs/api/java/math/BigInteger.html>

## 4.4 Single-scalar multiplication

The most time consuming operation in EECM\_Torben is the calculation of  $\left[\prod_{p^{\alpha_i} \leq B_1} p^{\alpha_i}\right] P$  also called a single-scalar multiplication, see algorithm 2.2.1. For this to be fast we need efficient formulas for doubling and addition/subtraction on the Edwards curve and a good strategy for computing the point. We do not compute the product  $\prod_{p^{\alpha_i} \leq B_1} p^{\alpha_i}$  and then calculate the point. Instead we do one prime at a time. First we discuss the arithmetic.

We saw in chapter 3 that the addition law is complete when  $d$  is a non square. This is, as mentioned, a really good property if one want to use Edwards curves in cryptography or simplify code. But in this thesis we do not go for simplification, we go for speed. Since doubling on Edwards curves is faster than ordinary addition, we will not use the property of completeness. Instead we actually neglect the possibility that a non defined addition/doubling could occur; when  $n$  is very large the probability that  $dx_1x_2y_1y_2 = \pm 1$  is very little. Below is shown register allocations of how we compute in EECM\_Torben using the formulas discussed in 3.3. The formulas are implemented in the class *Edward*.

**Register allocations.** In the following  $r_1, r_2$  and  $r_3$  contains  $x_1, y_1$  and  $z_1$  respectively.  $r_4, r_5$  and  $r_6$  likewise contains  $x_2, y_2$  and  $z_2$ .  $r_7$  and  $r_8$  are temporary registers.

Addition  $[x_1, y_1, z_1] + [x_2, y_2, z_2]$  (from left to right):

$$\begin{aligned} r_3 &\leftarrow r_3 \cdot r_6, & r_7 &\leftarrow r_1 + r_2, & r_8 &\leftarrow r_4 + r_5, & r_1 &\leftarrow r_1 \cdot r_4, & r_2 &\leftarrow r_2 \cdot r_5, & r_7 &\leftarrow r_7 \cdot r_8 \\ r_7 &\leftarrow r_7 - r_1, & r_7 &\leftarrow r_7 - r_2, & r_7 &\leftarrow r_7 \cdot r_3, & r_8 &\leftarrow r_1 \cdot r_2, & r_8 &\leftarrow d \cdot r_8, & r_2 &\leftarrow r_2 - r_1, \\ r_2 &\leftarrow r_2 \cdot r_3, & r_3 &\leftarrow r_3^2, & r_1 &\leftarrow r_3 - r_8, & r_3 &\leftarrow r_3 + r_8, & r_2 &\leftarrow r_2 \cdot r_3, & r_3 &\leftarrow r_3 \cdot r_1, \\ r_1 &\leftarrow r_1 \cdot r_7. \end{aligned}$$

Doubling  $2[x_1, y_1, z_1]$  (from left to right):

$$\begin{aligned} r_4 &\leftarrow r_1 + r_2, & r_1 &\leftarrow r_1^2, & r_2 &\leftarrow r_2^2, & r_3 &\leftarrow r_3^2, & r_4 &\leftarrow r_4^2, & r_3 &\leftarrow r_3 + r_3, & r_5 &\leftarrow r_1 + r_2, \\ r_2 &\leftarrow r_1 - r_2, & r_4 &\leftarrow r_4 - r_5, & r_3 &\leftarrow r_5 - r_3, & r_1 &\leftarrow r_3 \cdot r_4, & r_3 &\leftarrow r_3 \cdot r_5, & r_2 &\leftarrow r_2 \cdot r_5. \end{aligned}$$

Mixed addition does not need the first computation in the addition allocations. Subtraction is done by first calculating the additive inverse and then call addition. This means that subtraction cost a very tiny bit more than addition.

**NAF.** Here we describe the strategy used to compute  $\left[\prod_{p^{\alpha_i} \leq B_1} p^{\alpha_i}\right] P$  in particular we describe the addition chain being used. Say  $A$  is an array containing the primes from 2

---

**Algorithm 4.4.1** Basic calculation of  $\left[\prod_{p^{\alpha_i} \leq B_1} p^{\alpha_i}\right] P$ 


---

```

for  $i = 0 \rightarrow |A| - 1$  do
   $p = A[i]$ 
   $j = 1$ 
  while  $p^j \leq B_1$  do
     $P = [p]P$ 
     $j = j + 1$ 
  end while
end for

```

---

to  $B_1$ . We want to optimize the following computation where  $P$  is some point. First idea is to use a binary ladder. If a number  $\beta$  is written in its binary decomposition  $\beta = \sum_i a_i 2^i$  with  $a_i \in \{1, 0\}$ , on average  $\frac{1}{2}$  of the numbers  $a_i$  is zero. We are not satisfied with that. Instead we use a NAF (Non-adjacent form) representation introduced by Reitwieser in [22]. We use the property that any number  $\beta$  may be written as  $\beta = \sum_i b_i 2^i$  with  $b_i \in \{1, 0, -1\}$  and the additional property  $b_i b_{i+1} = 0$ . This may add one additional bit to the representation compared to the binary decomposition but instead on average  $\frac{2}{3}$  of the  $b_i$ 's are 0 (proved by Morain and Olivos in [20]). Generally the NAF representation is a unique signed-digit representation but in our representation we do not need a sign and hence the most significant bit is always 1. The two algorithms 4.4.2 and 4.4.3 compute what we need in EECM\_Torben

---

**Algorithm 4.4.2** Compute NAF representation
 

---

**Input:** Positive integer  $n$ .

**Output:** NAF representation  $(n_0, n_1, \dots, n_i)$  of  $n$ .

```

 $j = 0$ 
while  $n > 0$  do
  if  $n$  odd then
     $n_j = 2 - [n]_4$ 
     $n = n - n_j$ 
  else
     $n_j = 0$ 
  end if
   $n = \frac{n}{2}$ 
   $j = j + 1$ 
end while

```

---

Notice that in algorithm 4.4.2 if  $n$  is odd in some iteration of the while loop, we make  $n$  divisible by 4. Hence when we divide out by two in the end of the while loop, The number remains even and therefore the next bit will be set to 0 forcing the condition  $b_j b_{j+1} = 0$ . If  $n$  is an  $i$  bit number then its NAF representaion has either  $i$  or  $i + 1$  bits.

Compared to algorithm 4.4.1, algorithm 4.4.3 has several advantages. The number of additions is greatly reduced and we trade a lot of dedicated additions with doublings

---

**Algorithm 4.4.3** NAF calculation of  $\left[\prod_{p^{\alpha_i} \leq B_1} p^{\alpha_i}\right] P$ 


---

**Input:** Point  $P$  and a table of primes  $Q$  from 1 to  $B_1$ .**Output:** The single scalar multiplication  $\left[\prod_{p^{\alpha_i} \leq B_1} p^{\alpha_i}\right] P$ 

```

for  $i = 0 \rightarrow |Q| - 1$  do
   $j = 1$ 
   $p = Q[i]$ 
  while  $p^j \leq B_1$  do
    Compute NAF of  $p$ ;  $(n_0, n_1, \dots, n_l)$ , using algorithm 4.4.2
     $\overline{P} = P$ 
    for  $s = l \rightarrow 1$  do
       $P.dup()$ 
      if  $n_s = 1$  then
         $P = P.add(\overline{P})$ 
      end if
      if  $n_s = -1$  then
         $P = P.sub(\overline{P})$ 
      end if
    end for
     $\overline{P} = P$ 
     $j = j + 1$ 
  end while
end for

```

---

which is cheaper.

There exists various other single-scalar multiplication schemes that can be deployed. Instead of using a window of 2 one could use a window of 4 or even greater. In [3] Bernstein et al. also discusses a double-base single-scalar multiplication with a basis  $\{2, 3\}$ . Another method is to use a sliding window; compute point  $[a]P$  where  $a$  is a member of some basis e.g.  $\{1, 2, 3, 4, 9\}$ . This is also discussed in the paper by Bernstein et al.

## 4.5 Bounds $B_1$ and $B_2$

Bounds  $B_1$  and  $B_2$  are really the key if one wants to find a factor. In section 2.3 we saw that finding a factor, say  $p$ , of  $n$  really depends on whether the curve order of the reduction with  $p$  is  $B_1$ -power smooth or not.  $B_1$  could be picked optimal if  $p$  is known, but of course this knowledge will totally ruin the need for using ECM. Instead one must experiment with different values. In practice, what is done is that one value  $B_1$  is run sufficiently many times without success for one to become convinced that a higher value is called for. One normally starts with a low  $B_1$  value (In EECM\_Torben the default value for  $B_1$  is 10000 and for  $B_2$  its 100000) and maybe increase by a factor of 100, or double the value. The bound  $B_2$  is also a debatable value. In [10] (p. 343) it is suggested that when using a highly efficient second stage (such as a FFT extension) one should

pick  $B_2 \approx 100B_1$ . In EECM\_Torben we use  $B_2 = 10B_1$  since we do not use the highly advanced speed ups discussed in [10]. This also depends on your baby (your computer red.). The conventional wisdom is that stage 1 and stage 2 should take an approximate equal amount of time to process one curve.

In [23] Silverman and Wagstaff propose optimal values for  $B_1$  and  $B_2$  depending on the factor size one is searching for. That is, if we search for a factor size of 20 digits say, they suggest values for  $B_1$  and  $B_2$  that give a high probability of finding this factor. Paul Zimmermann has likewise computed optimal parameters for the bound  $B_1$  and suggest the expected amount of curves which is needed with respect to the factor size one search for. This can be found here <http://www.loria.fr/~zimmerma/records/ecm/params.html>. It should be taken into account that these estimates are relatively old i.e. was made before the invention of more advanced second stage methods (e.g. the FFT 2. stage).

Considering the former discussion the author has made the bounds in EECM\_Torben configurable by the flags  $-B1$  and  $-B2$  e.g. if a user wants to change the bounds to  $B_1 = 10^6$  and  $B_2 = 10^7$  one input  $-B1100000 -B21000000$ . The program does not allow a  $B_2$  value that is lower than  $B_1$ . Also a factor size flag has been added implementing Zimmermann's scheme; accessed by the flag  $-FS$  e.g. if one search for a factor of size, say 23 digit, then input option  $-FS23$ . Note that both inputting a factor size and bounds will make the program skip the factor size option.

## 4.6 Curve selection

EECM\_Torben is using the following method for picking a suitable curve. Pick  $x, y \in_R \mathbb{Z}/n\mathbb{Z}$  such that  $\gcd(xy, n) = 1$ . Then we may set  $(x^2 + y^2 - 1)(x^2y^2)^{-1} = d$  and check if  $d \neq 1$  (In practice we do not need to check for  $d = 0$ ). If one of these checks is false, then pick two new elements in  $\mathbb{Z}/n\mathbb{Z}$ .  $(x, y)$  is now a point on the Edwards curve  $E_{E,d}$  over  $\mathbb{Z}/n\mathbb{Z}$ . Below we discuss other (and possible more efficient) ways to do it. The theory behind is not discussed in this thesis and proof are omitted, but references are given. To specify the number of curves EECM\_Torben should maximally process input the option  $-NC$ .

In [4] Bernstein et al. present two different parametrisations for Edwards curves (and one for the more general twisted Edwards curves); the Atkin-Morain construction and the Montgomery construction which are both well known constructions for ordinary elliptic curves, is translated to Edwards curves. The Atkin-Morain and Montgomery construction force torsion groups  $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$  and  $\mathbb{Z}/12\mathbb{Z}$  respectively. Below we present the two different construction.

**Theorem 4.6.1.** (*Atkin-Morain on Edwards curves*) *Pick a point  $(s, t)$  on  $E_{W,-8,-32}(\mathbb{Q})$  and define  $\alpha = \left(\frac{t+25}{s-9} + 1\right)^{-1}$ ,  $\beta = 2\alpha\frac{4\alpha+1}{8\alpha^2-1}$  and  $d = \frac{2(2\beta-1)^2-1}{(2\beta-1)^4}$ . Then  $E_{E,d}$  has torsion*

group isomorphic to  $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$  and a point  $(x, y)$  with  $x = \frac{(2\beta-1)(4\beta-1)}{6\beta-5}$  and  $y = \frac{(2\beta-1)(t^2+50t-2s^3+27s^2-104)}{(t+3s-2)(t+s+16)}$

Finding points  $(s, t)$  may be done by choosing  $(s, t) = (12, 40)$  and computing multiples.

**Theorem 4.6.2.** (*Montgomery on Edwards curves*) Pick a point  $(s, t)$  on  $E_{W,-12,0}(\mathbb{Q})$  with  $(s, t) \notin \{(0, 0), (-2, \pm 4), (6, \pm 12)\}$ . Define  $d = \frac{-(s-2)^3(s+6)^3(s^2-12s-12)}{1024s^2t^2}$ . Then  $E_{E,d}$  has torsion group isomorphic to  $\mathbb{Z}/12\mathbb{Z}$  and point  $(x, y)$  with  $x = \frac{8t(s^2+12)}{(s-2)(s+6)(s^2+12-12)}$  and  $y = \frac{-4s(s^2-12s-12)}{(s-2)(s+6)(s^2-12)}$

In [4] the authors went even further than ensuring large torsion groups. They constructed curves with small parameters, large torsion and positive rank which, heuristically, should give a speed up. These “good curves” can be found on <http://eecm.cr.yp.to/goodcurves.html>. This optimization has not been implemented in EECM\_Torben.

## 4.7 Prime generation

We need a lot of primes for both stage 1 and 2. The number of primes is bounded by the stage 2 bound  $B_2$ . For finding these primes we use Eratosthenes sieve. This works by first creating a list from 2 to  $B_2$  (if  $B_2$  is odd otherwise we only need up to  $B_2 - 1$ ). Take the smallest number in the list; 2, and cross out all its multiples up to  $B_2$ . Continue by taking the next number in the list that has not been crossed (this will be 3) out and cross out all the multiples of this number up to  $B_2$ . Continue by taking the next number in the list that has not been crossed out (will be 5) and cross out all the multiples of this number up to  $B_2$ .... Continue until we reach  $\sqrt{B_2}$ . All numbers not crossed out are the primes up to  $B_2$ .

The implementation of Eratosthenes sieve is provided by the author. We use several speed ups e.g. to minimize the memory needed we only store odd numbers and we only check odd multiples of a prime. This sieving method is extremely fast; amortized using only  $O(\log \log B_2)$  work per prime. The drawback is the memory requirement. We save a factor of 2 by only storing odd numbers but we are still in the need of  $\frac{B_2}{2}$  space. It is not optimal but sufficient for our purpose.

Some optimizations for the memory use and speed is: Use a segmented sieve where sieving is done in segments. This reduce the memory need for the sieving array to the size of the segment. One may consider Eratosthenes sieves as a segmented sieve with just one segment. One way to obtain a significant speed up is to implement a wheel in which you skip multiples of 2, 3, 5,.. up to some limit. This may also reduce the memory requirements.

## 4.8 Additional comments

On a machine with multiple cores (which modern machines normally has (or should have)) a significant speed up may be archived by distributing the work of an algorithm among the cores. ECM is particular easy to parallelize; One may run one (or more) curves on each core. In EECM\_Torben this is done by implementing the *Runnable* interface. Although in Java you are not 100% certain that all cores are used. The user may specify how many threads that should be used with the option `-NT` (The number of threads running at the same time, not the total number of threads).

## 4.9 Experiments

In this chapter we present performance tests for EECM\_Torben. The method used is the following: For a random  $\beta$ -bit prime we picked a random  $(200-\beta)$ -bit prime and multiplied these together to obtain a 200-bit number with two prime factors. For a particular  $\beta$ -bit prime a sample set of 500 numbers was created. It was then run with EECM\_Torben using the option `-FS $\alpha$`  where  $\alpha$  is the factor size with  $\alpha \approx \frac{\beta}{3}$ . We have measured: Time, number of curves, number of modular multiplications (multiplications and squaring) and the total number of modular operations.

The following charts are average plots e.g. a run with the 15-bit sample set all times are added and divided with 500 to get the average time over the 500 samples. All output and sample set files may be downloaded by visiting <http://home.imf.au.dk/himsen/Cryptography.html>.

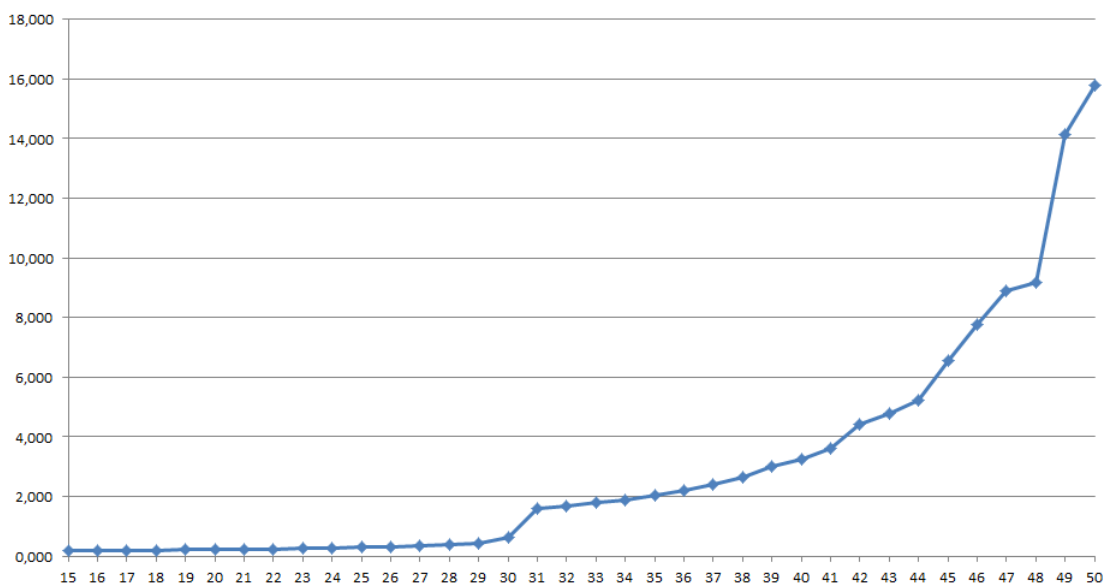
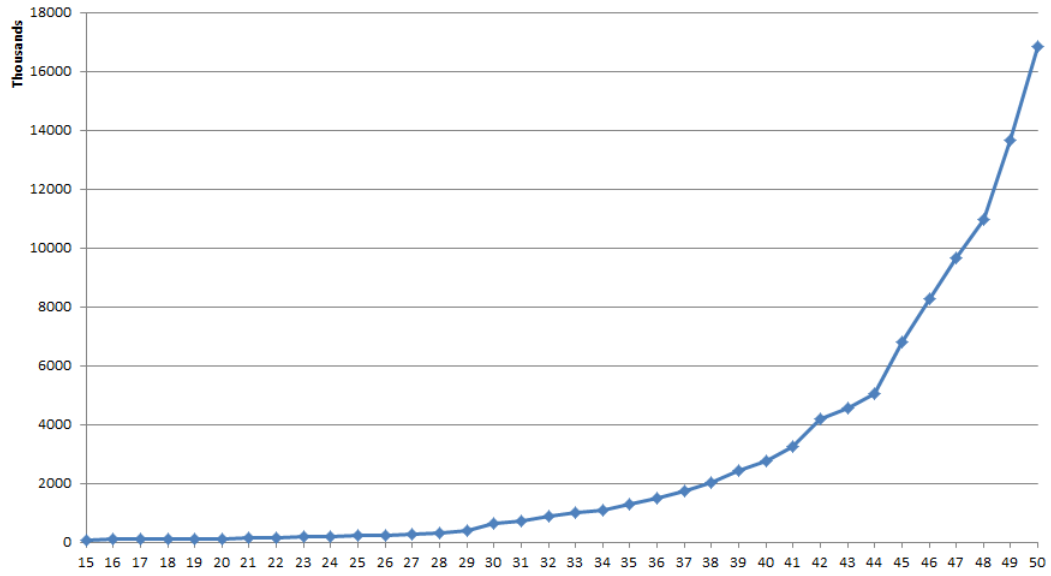
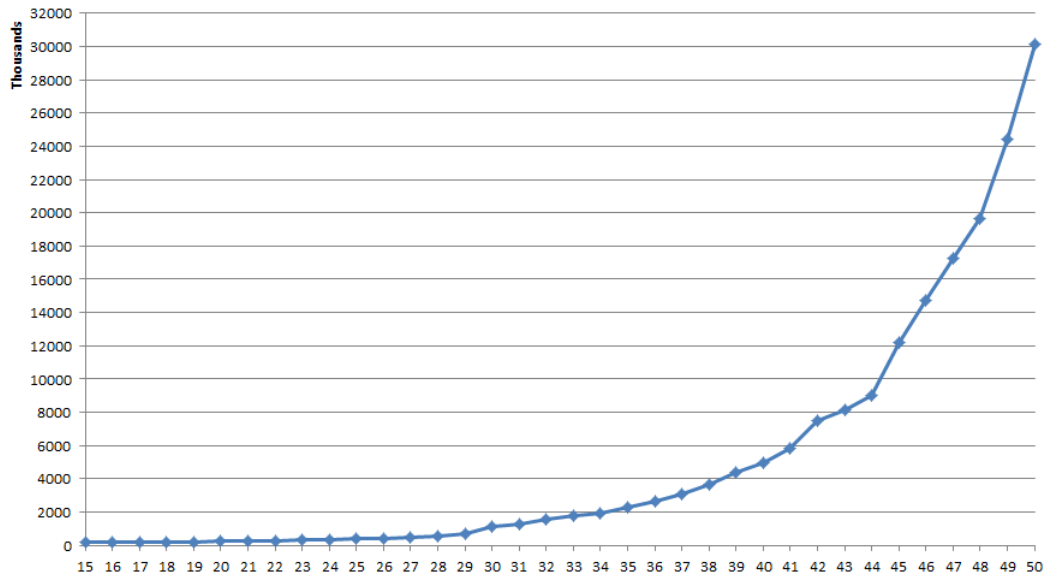


Figure 4.1: Average time.

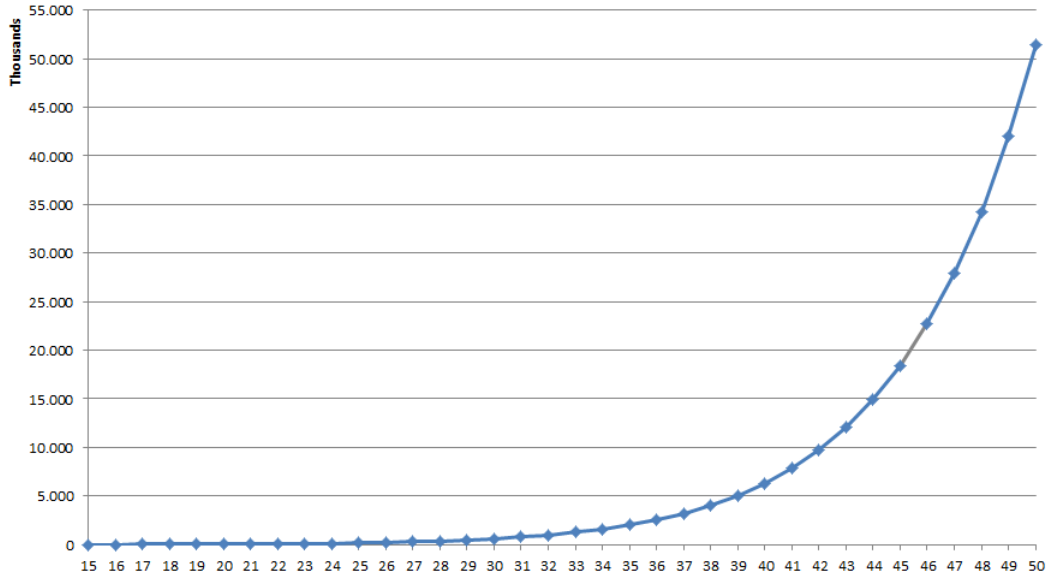


**Figure 4.2:** Average modular multiplications (multiplications+squarings).

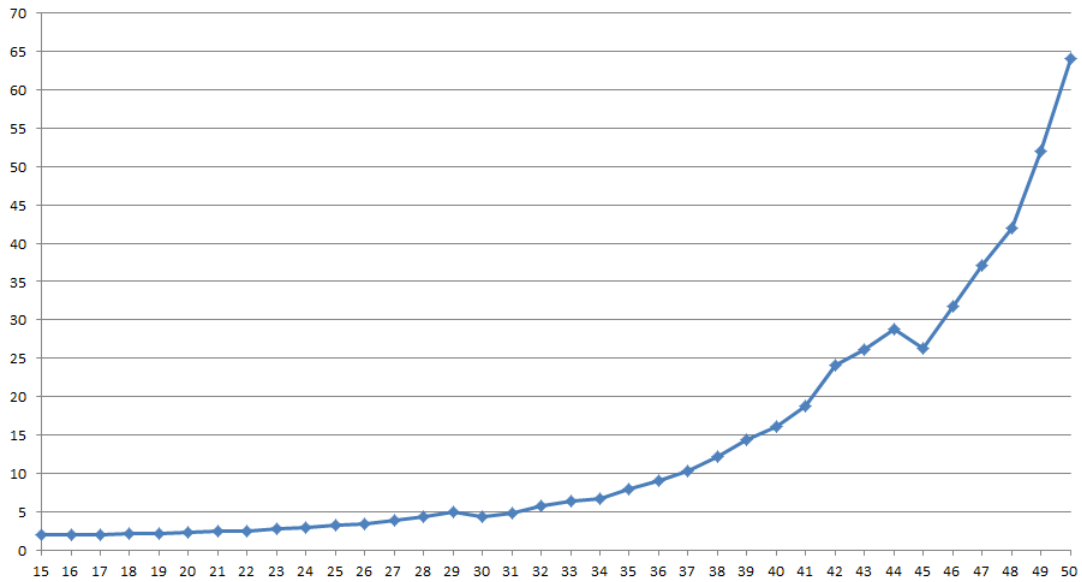


**Figure 4.3:** Average total modular operations (multiplications+squarings+additions+inversions)

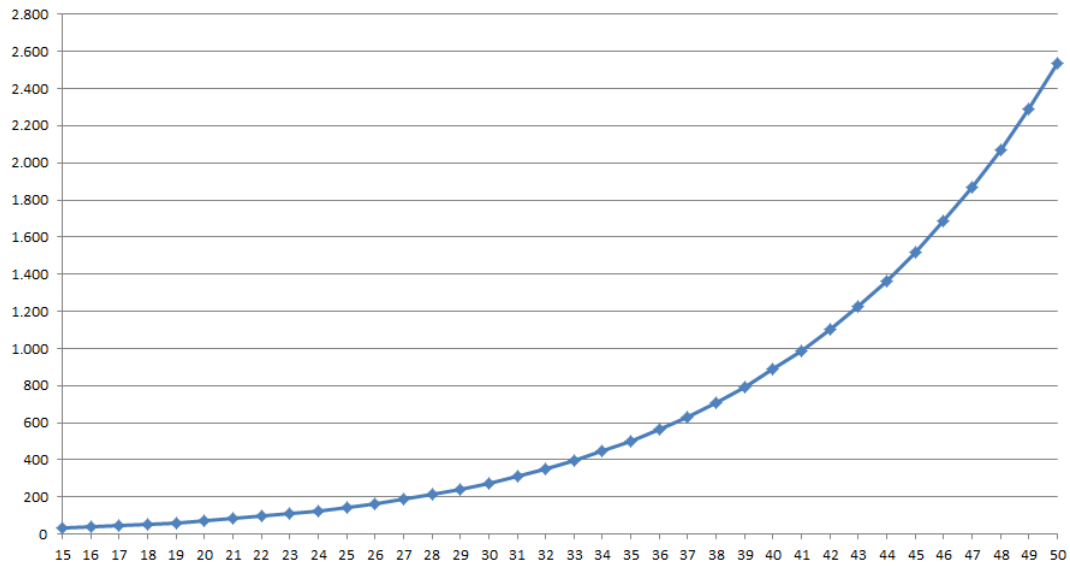




**Figure 4.4:** The theoretical total number of modular operations. For a  $\beta$ -bit factor  $8 \cdot e^{\sqrt{2 \ln(2^\beta) \ln(\ln(2^\beta))}}$  is plotted. See remark 2.3.5.



**Figure 4.5:** Average number of curves.



**Figure 4.6:** The theoretical number of curves needed. For a  $\beta$ -bit factor  $e^{\sqrt{\frac{1}{2} \ln(2^\beta) \ln(\ln(2^\beta))}}$  is plotted. See remark 2.3.5.

*The magic words are squeamish ossifrage<sup>1</sup>.*

— RSA message encoded in 1977 by Ron Rivest.

---

<sup>1</sup>Rivest estimated that breaking this message by factoring the 125-digit number would require 40 quadrillion years. It was broken using idle times on machines connected to the internet.

## APPENDIX A

# Maple scripts

The following are scripts for the computer algebra system Maple (Maple 12), verifying different kind of identities in the thesis. The maple files may be downloaded by visiting <http://home.imf.au.dk/himsen/Cryptography.html>.

### Script 1

```
> T := (x, y) -> (x^2 + y^2 - 1 - (1 - 4*r^3/r^2) * x^2 * y^2) : #d=1-4*r^3/r^2.
> W := (x, y) -> (y^2 - x^3 - (r^2/l^2 - 2*l) * x^2 - l^2 * x) :
> simplify([W((l*(1+y)/(1-y), r*(1+y)/(x*(1-y))), [T(x, y)]); # (x,y) -> (l*(1+y)/(1-y), r*(1+y)/(x*(1-y))) E_E, d to E
```

### Script 2

```
> T := (x, y) -> (x^2 + y^2 - 1 - d * x^2 * y^2) :
> delta := (x1, y1, x2, y2) -> ((x1*y2 + y1*x2)^2 * (1 - d * x1*x2*y1*y2)^2 + (y1*y2 - x1*x2)^2 * (1 + d * x1*x2*y1*y2)^2) :
> delta := (x1, y1, x2, y2) -> ((x1^2 + y1^2 - (x2^2 + y2^2) * d * x1^2 * y1^2) * (x2^2 + y2^2 - (x1^2 + y1^2) * d * x2^2 * y2^2) + d * (x1*y2 + y1*x2)^2 * (y1*y2 - x1*x2)^2) :
> simplify([delta(x1, y1, x2, y2) - delta(x1, y1, x2, y2)], [T(x1, y1), T(x2, y2)]); #Check identity \delta.
```

### Script 3

```
> T := (x, y) -> (x^2 + y^2 - 1 - d * x^2 * y^2) :
> addE := (x1, y1, x2, y2) -> ((x1*y2 + x2*y1) / (1 + d * x1*x2*y1*y2), (x1*x2 - y1*y2) / (1 - d * x1*x2*y1*y2)) :
> simplify([T(addE(x1, y1, x2, y2))], [T(x1, y1), T(x2, y2)]); # (x1, y1) +_Edwards (x2, y2) = (x3, y3). Verifies x3^2 + y3^2 = 1 + d * x3^2 * y3^2
```

**Script 4**

- >  $T := (x, y) \rightarrow (x^2 + y^2 - 1 - d \cdot x^2 \cdot y^2):$
- >  $ui := (x1, y1) \rightarrow \frac{(1 + y1)}{(1 - y1)}:$
- >  $vi := (x1, y1) \rightarrow \frac{2 \cdot (1 + y1)}{(1 - y1) \cdot x1}:$
- >  $addx := (x1, y1, x2, y2) \rightarrow \left( \frac{\left( \frac{1}{1-d} \right) \cdot (y2 - y1)^2}{(x2 - x1)^2} - \frac{2 \cdot (1 + d)}{1 - d} - x1 - x2 \right):$
- >  $addy := (x1, y1, x2, y2) \rightarrow \left( \frac{(y2 - y1)}{(x2 - x1)} \cdot (x1 - addx(x1, y1, x2, y2)) - y1 \right):$
- >  $simplify\left(\left[ addx(0, 0, ui(x2, y2), vi(x2, y2)) - \frac{1}{ui(x2, y2)} \right], [T(x2, y2)]\right); \#x\text{-coordinate in } -P_3 + P_2 = \left( \frac{1}{u_2}, -\frac{v_2}{u_2^2} \right).$
- >  $simplify\left(\left[ addy(0, 0, ui(x2, y2), vi(x2, y2)) + \frac{vi(x2, y2)}{ui(x2, y2)^2} \right], [T(x2, y2)]\right); \#y\text{-coordinate in } -P_3 + P_2 = \left( \frac{1}{u_2}, -\frac{v_2}{u_2^2} \right).$
- >  $simplify\left(\left[ ui(x1, y1) - \frac{1}{ui(x1, -y1)} \right], [T(x1, y1)]\right); \#u_1 = \frac{1}{u_2}.$
- >  $simplify\left(\left[ vi(x1, y1) - \frac{vi(x1, -y1)}{ui(x1, -y1)^2} \right], [T(x1, y1)]\right); \#v_1 = \frac{v_2}{u_2^2}.$

# Bibliography

- [1] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. Cryptology ePrint Archive, Report 2008/013, 2008. <http://eprint.iacr.org/>.
- [2] Daniel J. Bernstein, Peter Birkner, and Tanja Lange. Starfish on strike. *IACR Cryptology ePrint Archive*, 2010:367, 2010.
- [3] Daniel J. Bernstein, Peter Birkner, Tanja Lange, and Christiane Peters. Optimizing double-base elliptic-curve single-scalar multiplication. Cryptology ePrint Archive, Report 2007/414, 2007. <http://eprint.iacr.org/>.
- [4] Daniel J. Bernstein, Peter Birkner, Tanja Lange, and Christiane Peters. Ecm using edwards curves. *IACR Cryptology ePrint Archive*, 2008:16, 2008.
- [5] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. 4833:29–50, 2007.
- [6] Daniel J. Bernstein and Tanja Lange. Inverted edwards coordinates. In *AAECC*, pages 20–27, 2007.
- [7] Daniel J. Bernstein, Tien ren Chen, Chen mou Cheng, Tanja Lange, and Bo yin Yang. Ecm on graphics cards.
- [8] Richard P. Brent. Some integer factorization algorithms using elliptic curves. *Australian Computer Science Communications*, 8:149–163, 1986.
- [9] Wouter Castryck, Steven Galbraith, and Reza Rezaeian Farashahi. Efficient arithmetic on elliptic curves using a mixed edwards-montgomery representation. Cryptology ePrint Archive, Report 2008/218, 2008. <http://eprint.iacr.org/>.
- [10] Richard Crandall and Carl Pomerance. *Prime numbers*. Springer, New York, second edition, 2005. A computational perspective.
- [11] Reza Rezaeian Farashahi Daniel J. Bernstein, Tanja Lange. Binary edwards curves. In *Cryptographic hardware and embedded systems—CHES 2008*, Lecture Notes in Computer Science.

- [12] Tanja Lange Daniel J. Bernstein. Explicit-formulas database. <http://hyperelliptic.org/EFD>, 2007.
- [13] Harold M. Edwards. A normal form for elliptic curves. *Bull. Amer. Math. Soc. (N.S.)*, 44(3):393–422 (electronic), 2007.
- [14] Carl Friedrich Gauss. Werke.
- [15] Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Twisted edwards curves revisited. In *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ASIACRYPT '08*, pages 326–343, Berlin, Heidelberg, 2008. Springer-Verlag.
- [16] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. pages 104–113. Springer-Verlag, 1996.
- [17] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126(3):649–673, 1987.
- [18] J.S. Milne. *Elliptic Curves*. BookSurge Publishers, 2006.
- [19] P. L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. 48:243–264, 1987.
- [20] F. Morain and J. Olivos. Speeding up the computations on an elliptic curve using addition-subtraction chains. 24:531–544, 1990.
- [21] Christiane Peters. *Curves, Codes, and Cryptography*. PhD thesis, Technische Universiteit Eindhoven, 2011.
- [22] George W. Reitwiesner. Binary arithmetic. *Advances in Computers*, pages 231–308, 1960.
- [23] R. D. Silverman and S. S. Wagstaff, Jr. A practical analysis of the the elliptic curve factoring algorithm. 61:445–462, 1993.
- [24] Paul Zimmermann and Bruce Dodson. 20 years of ecm. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*, volume 4076 of *Lecture Notes in Computer Science*, pages 525–542. Springer, 2006.