12-2011

Elliptic Curve Crytography: A Computational Science Model

Nuh Aydin
Kenyon College, aydinn@kenyon.edu

Elliptic Curve Cryptography

A Computational Science Module

Supported by
National Science Foundation, NSF CCLI Grant (DUE 0618252)

Module Author: Nuh Aydin

Department of Mathematics
Kenyon College
aydinn@kenyon.edu

December 2011

Abstract

This is an advanced module in cryptography. Intended as a follow up to a previous module titled "Public Key Cryptography and the RSA Cryptosystem" by the author, it introduces a few cryptosystems based on the elliptic curves. Advantages of using elliptic curves over other public-key systems are also discussed. Elliptic curves are introduced with a minimal amount technical material accompanied by intuitive motivation. Some material from public key cryptosystems that are directly relevant to the elliptic curve cryptosystems are introduced or reviewed at the beginning of the module.

This module assumes some basic knowledge in cryptography and abstract algebra. Necessary background material include an understanding of basic ideas of symmetric-key and public-key cryptography, concepts from computational complexity, some elementary number theory and basic definitions of groups, rings and fields. In particular, it assumes that the reader is familiar with the content (or an equivalent knowledge) of the earlier module "Public Key Cryptography and the RSA Cryptosystem". It is assumed that students are familiar with such concepts as groups, subgroups, cyclic groups, order of an element, order of a group, generators, Lagrange's theorem, the multiplicative group $\mathbb{Z}_n^*$ of units modulo n, and definitions of rings and fields. These topics are typically covered in a first course in abstract algebra. In fact, this module could be used in an abstract algebra course to illustrate some of the applications of the topics in abstract algebra.

Keywords: public key cryptography, elliptic curves, elliptic curve cryptography.

1 Introduction and Motivation

Recall the basic difference between symmetric-key cryptosystems and public-key cryptosystems. In the former, there is a secret key that is used for both encryption and decryption. In the latter, the encryption key is public and separate from the decryption key which is private. One of the best known examples of the public-key cryptosystems is the RSA system that the reader is assumed to be familiar with. You recall that the security of the RSA system is based on the assumed difficulty of the integer factorization problem with "large" integers. Currently, "large" means integers that require around 1000 bits to store them in computer memory. Having to deal with such large numbers introduces difficulties in practical applications. It would be desirable to have a system that offers the same level of security as the

RSA but requires computations with integers of smaller magnitudes. Elliptic curve cryptography offers such an alternative.

In this module, you will learn about elliptic curves and how they are used in cryptography. Moreover, we will discuss their advantages over other public-key cryptosystems. First, we introduce the ElGamal cryptosystem, another well-known example of a public-key cryptosystem based on the discrete logarithm problem. The reason we introduce the ElGamal system is that it is the most natural system to apply the elliptic curves to. Like ElGamal system, Diffie-Hellman key exchange protocol is also based on the discrete logarithm problem.

2 Diffie-Hellman Key Agreement and the Discrete Logarithm Problem

One of the interesting applications of public-key cryptosystem is the generation of a secret key over an insecure (but authenticated) channel. A well-known example of such a protocol is proposed by Diffie and Helman [2], inventors of the idea of public-key cryptography, and is known as *Diffie-Hellman key agreement(exchange)*. Suppose Alice and Bob want to agree on a secret key over an insecure channel. Here is the protocol they need to follow according to this scheme.

1. Either Alice or Bob chooses a large prime, and a generator α of the cyclic group $\mathbb{Z}_p^*$. Both p and α can be made public.

2. Alice chooses a random (and secret) integer a, $1 < a < p$, and Bob chooses a random (and secret) integer b, $1 < b < p$.

3. Alice sends $x = \alpha^a$ to Bob, and Bob sends $y = \alpha^b$ to Alice.

4. Alice computes y^a and Bob computes x^b.

Note that all of the computations are performed in the group $\mathbb{Z}_p^*$, hence they are mod p. Also note that both Alice and Bob end up with the same value: $y^a = (\alpha^b)^a = \alpha^{ab} = (\alpha^a)^b = x^b$, which is the common secret key between Alice and Bob.

Eve, of course, would like to be able to retrieve the secret key k from the information available to her: $p, \alpha, \alpha^a, \alpha^b$. This is known as the *Diffie-Hellman problem* (DFH). It turns out that this problem is closely related to another well-known problem, the discrete logarithm problem (DLP): Given a prime p, a generator α of $\mathbb{Z}_p^*$ and α^x, find x.

Exercise 2.0.1 *Show that DLP implies the DFH. That is, if Eve can solve the DLP then she can also solve the DFH.*

Therefore, if there is a polynomial time algorithm for the DLP, then there is a polynomial time algorithm for the DFH as well. However, no polynomial time algorithm is known for the DLP. It is also an open problem whether it is possible to solve the DFH problem, in general, without having to solve the DLP. To this date, the best known general solution for the DFH problem is through the DLP. For large enough groups, the DFH (and the DLP) is assumed to be a hard problem. The security of Diffie-Hellman key exchange protocol is based on this assumption. With this assumption the function $f : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ given by $f(x) = \alpha^x$ is considered to be a one-way function. Recall that the security of the RSA cryptosystem is based on the assumed difficulty of the factorization of large integers.

It is not too hard to modify the Diffie-Hellman protocol so that it works with three participants Alice, Bob and Carol.

Exercise 2.0.2 *Modify the Diffie-Hellman protocol so that Alice, Bob and Carol end up with the same secret key by communicating over an insecure channel.*

3 The ElGamal Cryptosystem and Signature Scheme

3.1 The Cryptosystem

We have seen a key exchange protocol whose security is based on the difficulty of the discrete logarithm problem. There is a full cryptosystem based on this problem. Proposed in 1985 [3], this scheme is known as ElGamal cryptosystem, named after the author who introduced it. It modifies the Diffie-Hellman protocol so that it can be used as an encryption and decryption system. Its security is also based on the difficulty of the DLP. It can be described as follows:

Alice wants to send secret messages to Bob. Bob chooses a large prime p, a random integer b, $1 \leq b \leq p-1$, and a primitive element α of $\mathbb{Z}_p^*$. He computes $\beta = \alpha^b$, and publishes the information (p, α, β). Alice wants to send her message m privately. She needs to do the following.

1. Download Bob's public information.

2. Treat her message m as an integer $0 \leq m < p$. If m is greater than $p-1$ she breaks m into smaller blocks. (Details of this are discussed in the earlier module on the RSA system [1].)

3. Alice chooses a random integer a and computes $R = \alpha^a$.

4. She also computes $M = \beta^a m$.

5. Alice sends the pair (R, M) to Bob.

Receiving the pair (R, M), Bob decrypts the message using his private key b. It is clear that if Eve can solve the DLP then she can break the ElGamal system.

Exercise 3.1.1 *Show how exactly Bob decrypts the ciphertext. Verify that his decryption works correctly.*

3.2 The Signature Scheme

The ElGamal cryptosystem can be modified to produce a signature scheme. Suppose Alice wants to send a signed message m to Bob. Here the message is not necessarily secret. What is important is for Bob to verify that it is signed by Alice. To make the system work, Alice first chooses a large prime p, a primitive element α of $\mathbb{Z}_p^*$, and a secret integer a. She then computes $\beta = \alpha^a$ and proceeds as follows.

1. Publish (p, α, β)

2. Choose a random and private integer k such that $\gcd(k, p-1) = 1$

3. Compute $R = \alpha^k$

4. Compute $S \equiv k^{-1}(m - aR) \mod p - 1$

5. Send the signed mesage (m, R, S) to Bob

Upon receiving the signed message, Bob proceeds as follows

1. Download Alice's public information.

2. Compute $V_1 = \beta^R R^S \mod p$

3. Compute $V_2 = \alpha^m \mod p$

4. Accept the signature if $V_1 \equiv V_1 \mod p$.

It is not hard to see that Bob's verification scheme works.

Exercise 3.2.1 *Show that Bob's verification scheme is correct.*

The security of this signature scheme also depends on the difficulty of DLP. If Eve can find out the value of a (by solving the DLP problem for example) then she can follow the protocol and attach Alice's signature on any document she chooses.

If Alice wants to sign a second document, then it is very important that she uses a different value of the private number k. If not, it is quite likely for Eve to obtain the value of the private key a, and consequently completely break the system. If Alice signs two different messages m_1 and m_2 using the same random value k, then she produces the same $R = \alpha^k$ value for both messages but different S values: $S_1 \equiv k^{-1}(m_1 - aR)$ and $S_2 \equiv k^{-1}(m_2 - aR)$. It follows that $S_1 k - m_1 \equiv -aR \equiv S_2 - m_2 \mod p-1$, and $(S_1 - S_2)k \equiv m_1 - m_2 \mod p-1$. The last congruence has a solution if $d = \gcd(S_1 - S_2, p-1)$ divides $m_1 - m_2$ in which case it has d solutions for k (the only unknown in the equation) and there is a well known and efficient procedure to find these solutions. Eve can test every possible value of k until she finds the one with $\alpha^k = R$. Next she can solve the congruence $aR \equiv m_1 - kS_1 \mod p-1$ for a. Again there are $\gcd(R, p-1)$ solutions when $\gcd(R, p-1)$ divides $m_1 - kS_1$. She tests each possible solution until she finds the one that satisfies $\alpha^a = \beta$. At that point she completely breaks the system.

Example 3.2.2 .

Let us show this point with a concrete example. Suppose Alice uses the prime $p = 337741$ (which is too small to use in practice but will serve adequately as an illustration), and the generator $\alpha = 173$. She has a secret integer a. Her public information is $(p, 173, \beta)$ where $\beta = 251222$. Suppose her first message to sign is $m_1 = 120324$. She chooses a secret value for k (which is relatively prime with $p-1$). She finds $R = \alpha^k \mod p = 163,949$. Next she computes $S_1 \equiv k^{-1}(m_1 - aR) \equiv 28774 \mod p-1$. Her signed message is then the triple $(120324, 163949, 28774)$.

Now suppose Alice decides to sign another message $m_2 = 201027$ using the same secret value k. Following the protocol, she comes up with the signed message $(201027, 163949, 191293)$. Observing Alice's messages, Eve immediately realizes that she used the same value of k from the fact that the middle values in each message are equal. Eve proceeds as follows to first obtain the value of k, then the value of a.

First, she constructs the congruence $(S_1 - S_2)k \equiv m_1 - m_2 \mod p-1$ in which k is the only unknown. In this case the equation turns out to be $-162519k \equiv -80703 \mod p-1$, or $175221k \equiv 257037 \mod 337740$. Since $\gcd(S_1 - S_2, p-1) = 3$ divides 337740, this equation has 3 solutions. These solutions can be found by first dividing the congruence by 3 and getting $58407k \equiv 85679 \mod 112580$. Now, $\gcd(58407, 112580) = 1$ so this congruence has a unique solution $\mod 112580$ which

is 1237. Then, the 3 solutions of the previous congruence are 1237, 113817, and 226397. Testing each solution Eve finds that the correct value of k is 1237.

Having obtained the value of k, Eve can discover the value of a as well from the equation $aR \equiv m_1 - S_1 k \mod p-1$ in which a is the only unknown for her. Obtained by rewriting the equation for S_1, this gives her the congruence $163949a \equiv 327326 \mod 337740$. Since $\gcd(163949, 337740) = 1$, this congruence has the unique solution 132394, which is the value of the secret key a. Now, Eve can forge Alice's signature on any document she chooses.

3.3 An Implementation of ElGamal Cryptosystem in Maple

We assume that the reader is familiar with the previous module **??** on the RSA cryptosystem and its implementation in Maple. Therefore, we will only explain parts that do not appear in the previous module (including the supplementary Maple files).

Bob chooses a prime number as follows (which is much smaller than what should be used in practice)

```
> N := (rand(10^19 .. 10^20-1))():
> p := nextprime(N);
                        47745199971245512067
> isprime(p);
                                  true
```

Next, he needs to find a generator of the cyclic group $\mathbb{Z}_p^*$. A practical way of doing this is to pick a random value and check that it is a generator (if not try another value). This can be accomplished as follows:

```
> alpha := (rand(10^5 .. 2*10^5))();

                                183417
>Primitive(x-alpha) mod p;

                                  true
```

Finally, Bob chooses a random and secret value b. Let us say he chooses $b = 17305$ and computes

```
> b := 17305;
> beta := `mod`(Power(alpha, b), p);
                        31627624511892192254
```

He then publishes the following information:
$(p, \alpha,\ beta) = (75434670514966341829, 183417, 31627624511892192254)$.

Now it is Alice's turn. She first downloads Bob's public data. Suppose her secret message is "Red door". She converts this message to a number $\mod p$. She can do that as follows:

```
>plaintext := "Red door";
                              "Red door"
> plaintextNumber := convert(plaintext, bytes);
             [82, 101, 100, 32, 100, 111, 111, 114]
> m := convert(plaintextNumber, base, 256, p);
                        [82459319185695307O6]
```

Note that if the message (after converting to a number) was larger than p, she would need to break it into smaller parts. (This is shown in the previous module). Next, she chooses a random integer a. Say, she picks $a = 454086$. Then she computes:

```
> a := 454086:
> R := `mod`(Power(alpha, a), p);
                         45601211411269254811
> M := `mod`(Power(beta, a)*m, p);
                        [89575804709965808 20]
```

And she sends the pair (R, M) to Bob. After receiving this pair all Bob needs to do is to compute

```
> decipherModp := `mod`(M*Power(R, p-b-1), p);
                        [82459319185695307O6]
```

which is exactly Alice's message in numerical form. What Bob really computes to decrypt is the quantity MR^{-b}. Since Maple's Power function does not accept a negative integer as exponent we used the fact that $R^{-b} = R^{p-1-b} \mod p$ because $R^{p-1} = 1 \mod p$. To get the message in English, Bob converts the numerical value back to characters

```
> decipherMod256 := convert(decipherModp, base, p, 256);
             [82, 101, 100, 32, 100, 111, 111, 114]
> decipherText := convert(decipherMod256, bytes);
                            "Red door"
```

4 Elliptic Curves in Cryptography

4.1 Introduction

Elliptic curves turn out to be useful for solving various types of important problems in mathematics. For example, Andrew Wiles' proof of a famous problem in number theory called Fermat's last theorem, that was an open conjecture for more than three centuries, employs elliptic curves (in addition to many advanced tools in various branches of mathematics) [12]. In the mid 1980's Miller and Koblitz proposed the idea of using elliptic curves in cryptography [4, 8]. Around the same time, Lenstra showed how to use elliptic curves to factor integers, an important problem for the RSA cryptosystem. In this module we will learn some basics facts about elliptic curves and how they can be used in cryptography. Moreover, we will discuss their advantages over other public-key cryptosystems.

Our treatment of elliptic curves will be somewhat simplified. We will skip or simplify some of the technical aspects of elliptic curves, yet give enough of the idea of their structure and their use in crytography. For a more rigorous and technically complete treatment of elliptic curves see, for example, the book [9].

Let F be a field of characteristic $\neq 2, 3$, and let $a, b \in F$. Suppose that the polynomial $x^3 + ax + b$ has no multiple roots. This is equivalent to assuming that $4a^3 + 27b^2 \neq 0$. Consider the equation

$$y^2 = x^3 + ax + b \tag{1}$$

An *elliptic curve* is the set of all ordered pairs $(x, y) \in F \times F$ that satisfy the equation (4.1), together with a special point denoted by $\mathcal{O}$ called the *point at infinity*. The ordered pairs (x, y) that satisfy the equation are called the points on the curve. When the field over which we consider the elliptic curve is the field of real numbers $\mathbb{R}$ we can sketch the graph of an elliptic curve. For example, consider the elliptic curve $y^2 = x^3 - 4x$ over $\mathbb{R}$. Its graph is given in Figure 1 (on the next page).

Note that this graph is symmetric about the x-axis because whenever the point (x, y) is on the curve, so is $(x, -y)$. Intuitively, the point at infinity is considered to be at either end of the y-axis (or any vertical line). A more rigorous way of describing

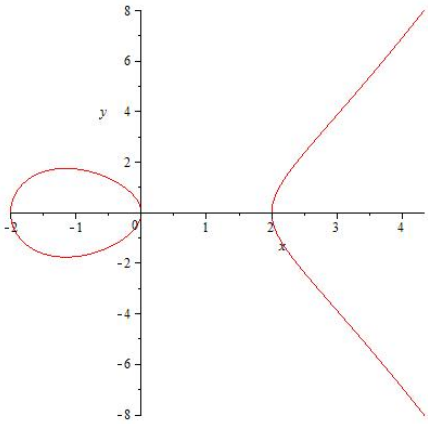


Figure 1: The graph of the curve $y^2 = x^3 - 4x$

this point is via projective geometry. We will not go into that direction, an intuitive idea will suffice.

We can consider the same equation $y^2 = x^3 - 4x$ over another field, for example over the finite field $\mathbb{Z}_5$ of integers modulo 5. In this case, we cannot sketch a graph of the solution set of the equation. It is simply a set of discrete points with coordinates in $\mathbb{Z}_5$ but we still call this set an elliptic curve over $\mathbb{Z}_5$. What are the points on this curve? To answer this question, first notice that the set of squares in $\mathbb{Z}_5$ are $\{0, 1, 4\}$. Since there are only 5 elements in the field $\mathbb{Z}_5$ we can simply test each element. Clearly, for $x = 0$, $y = 0$. So, $(0, 0)$ is one of the points. For $x = 1$ or $x = 3$, we get $y^2 = 2$ but this equation has no solution in $\mathbb{Z}_5$. For, $x = 2$ or $x = 4$ we get $y^2 = 0$, so the points $(2, 0)$ and $(4, 0)$ are also on the curve. Therefore, there are four points on this elliptic curve: $(0, 0), (2, 0), (3, 0)$ and the point $\mathcal{O}$. Note that we still consider the point at infinity even when there are a finite number of points on the curve. We will see that the point at infinity plays a crucial role for elliptic curves.

Exercise 4.1.1 *Find all the points on the curve $y^2 = x^3 - 4x$ over $\mathbb{Z}_7$.*

Remark 4.1.2 *Have you noticed that when they have a graph that we can sketch, elliptic curves do not have graphs that are ellipses? Then why are they named elliptic curves? They got their names from elliptic integrals. An integral of the form $\int \frac{dx}{\sqrt{x^3+ax^2+bx+c}}$ is called an elliptic integral because it arises from the computation of the arc length of an ellipse.*

Remark 4.1.3 *We can also define elliptic curves for fields of characteristic 2 and 3. In fact, the most general definition of an elliptic curve is an equation of the form*

$y^2 + axy + by = x^3 + cx^2 + dx + e$ where a, b, c, d, e are scalars (field elements). In the case of characteristic other than 2 or 3, the above equation can be transformed into the form given by equation 4.1.

4.2 The Group Structure

A surprising and useful property of elliptic curves is that the set of *rational points* on an elliptic curve forms a group under a certain operation. When we consider an elliptic curve E over $\mathbb{R}$, a rational point on E is a tuple (x, y) both of whose coordinates are in $\mathbb{Q}$, the field of rational numbers. When we consider an elliptic curve over a finite field $\mathbb{F}_q$, rational points are those that have both coordinates in $\mathbb{F}_q$, as opposed to some extension of $\mathbb{F}_q$.

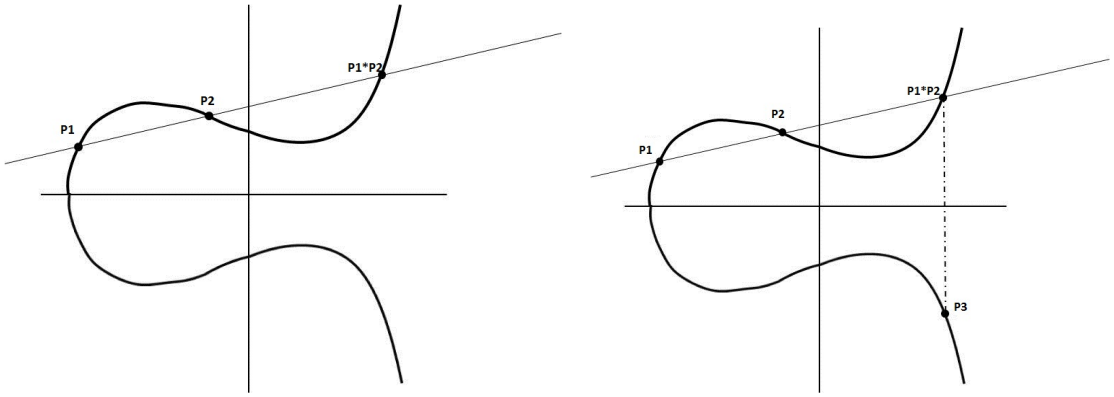
We will first explain this group structure then show how that can be useful in cryptography. Again, our treatment will not be strictly technical. Recall that to define a group we need a set and a binary operation. In this case the set is already defined: the set of rational points on an elliptic curve, together with the point at infinity. The binary operation will be an addition but it is not the straightforward, component-wise addition of points. There is a geometric intuition behind the way addition is defined on the rational points on an elliptic curve. We can describe and illustrate it for the curves defined over $\mathbb{R}$, and then generalize to curves defined over arbitrary fields. We derive algebraic formulas using geometric intuition from the curves over $\mathbb{R}$ and the good news is those algebraic formulas continue to hold in other fields even if we can not sketch graphs over arbitrary fields. Let us get started!

4.3 The Addition Law

Given two rational points P_1 and P_2 on an elliptic curve E, we can produce a third point P_3 which we call the sum of P_1 and P_2. This will be the basic binary operation needed to define a group. First, we draw a line ℓ that goes through P_1 and P_2. In case $P_1 = P_2$, we take the tangent line to the curve at that point. Let $P_1 * P_2$ be the point of intersection of the line ℓ as illustrated in Figure 2.

The next step is to define P_3 as the reflection of the point $P_1 * P_2$ about the x-axis, as illustrated in Figure 2.

Now, what are the coordinates of P_3 in terms of P_1 and P_2? Let two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ with rational coordinates that are not on a vertical line be given. We first need to find $P_1 * P_2 = (x_3, y_3)$. Then $P_3 = P_1 + P_2$ will be $P3 = (x_3, -y_3)$. The equation of the line ℓ through P_1 and P_2 is $\ell : y = Mx + B$ where $M = \frac{y_2 - y_1}{x_2 - x_1}$ and $B = y_1 - Mx_1 = y_2 - Mx_2$. We already know two of the three



(a) First step in finding the sum of P_1 and P_2: The point of intersection $P_1 * P_2$

(b) Finding the point $P_3 = P_1 + P_2$ through reflection

Figure 2: Finding $P_1 + P_2$ in two steps

points where this line intersects the curve, namely P_1, P_2. To find the third point of intersection, set the equation $y^2 = (Mx + B)^2 = x^3 + ax + b$. Rearranging this equation we obtain $p(x) = x^3 - M^2x^2 + (a - 2MB)x + b - B^2 = 0$. The polynomial $p(x)$ has three roots: x_1, x_2 and x_3. Since P_1 and P_2 satisfy this equation, we already know two of the three roots of $p(x)$. We can write $p(x) = (x - x_1)(x - x_2)(x - x_3)$ and equate the coefficients of like terms and obtain $x_3 = M^2 - x_1 - x_2$. Then $y_3 = Mx_3 + B$. Note that when the points P_1 and P_2 have rational components, x_3 and y_3 are also rational numbers. Therefore, P_3 is another rational point on the curve.

Example 4.3.1

Consider the elliptic curve $y^2 = x^3 - 4x$ over $\mathbb{Z}_7$. It is easy to verify that the points $P_1 = (1, 2)$ and $P_2 = (3, 1)$ are on this curve. Let us find their sum $P_3 = P_1 + P_2$ according to the calculations above. First find $M = \frac{y_2 - y_1}{x_2 - x_1} = \frac{-1}{2} = -4 = 3 \mod 7$, then $B = y_1 - Mx_1 = 2 - 3 \cdot 1 = -1 = 6 \mod 7$. Next, $x_3 = M^2 - x_1 - x_2 = 3^2 - 1 - 3 = 5$ and $y_3 = Mx_3 + B = 3 \cdot 5 + 6 = 0 \mod 7$. Note that in this case $-y_3 = y_3$. Therefore, we obtain the point $P_3 = (5, 0)$. It is easy to verify that this point is on the curve as well as on the line.

What if we want to add a point on the curve to itself, i.e. $P_1 = P_2$? We can modify the above procedure by replacing the line that goes through P_1 and P_2 with the line that is tangent to the curve at P_1. In this case, we use calculus (and implicit differentiation) to find the slope of the tangent line to the curve $y^2 = x^3 + ax + b$ at $P_1 = (x_1, y_1)$. It turns out to be $M = \frac{3x_1^2 + a}{2y_1}$. By following similar set of calculations

we find the coordinates of $P_3 = (x_3, -y_3)$ to be $x_3 = M^2 - 2x_1$ and $y_3 = Mx_3 + B$. This is illustrated in the figure below. We also use the convention that the tangent line to the curve at P_1 intersects the curve three times with P_1 counted twice.

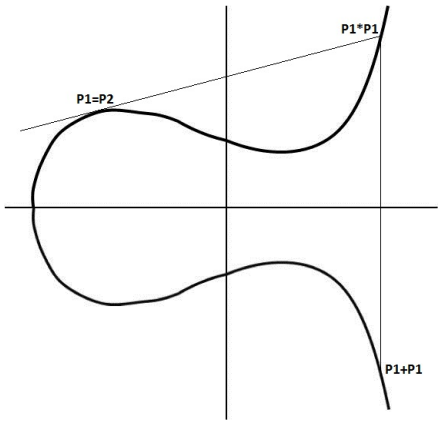


Figure 3: Finding the point $P_3 = P_1 + P_1 = 2P_1$

Exercise 4.3.2 *Show that the slope of the tangent line to the elliptic curve at* $P_1 = (x_1, y_1)$ *is* $M = \frac{3x_1^2+a}{2y_1}$*, and show the derivation of the formulas for* x_3 *and* y_3*.*

So, given a point P on the elliptic curve, we know how to compute $2P$. Then we can compute $P + 2P = 3P$, and $n \cdot P = \underbrace{P + P + \cdots + P}_{n-times}$ for any positive integer n.

Example 4.3.3

Still consider the elliptic curve $y^2 = x^3 - 4x$ over $\mathbb{Z}_7$. Let us add the point $P_1 = (3, 6)$ to itself. We compute $M = \frac{3\cdot 3^2+3}{2\cdot 6} = \frac{30}{12} = \frac{2}{5} = \frac{2}{-2} = -1 = 6 \mod 7$, $B = y_1 - Mx_1 = 6 - (-1)3 = 9 = 2 \mod 7$, $x_3 = (-1)^2 - 2 \cdot 3 = 1 - 6 = -5 = 2 \mod 7$, and $y_3 = (-1) \cdot 2 + 2 = 0$. Hence, $P_3 = (2, 0)$. It is easy to verify that the point $(2, 0)$ is indeed on the curve.

Now consider the case that x-coordinates of the points P_1 and P_2 are the same (and y-coordinates are different), i.e., they lie on a vertical line. What is the intersection of a vertical line with the elliptic curve? This is one of the places where we invoke the point at infinity $\mathcal{O}$. We will define the sum $P_1 + P_2 = \mathcal{O}$ to be the point at infinity in this case. Note that we are also assuming the reflection of the point $\mathcal{O}$ is itself, intuitively corresponding to the convention that it can be obtained by extending a vertical line indefinitely in either direction.

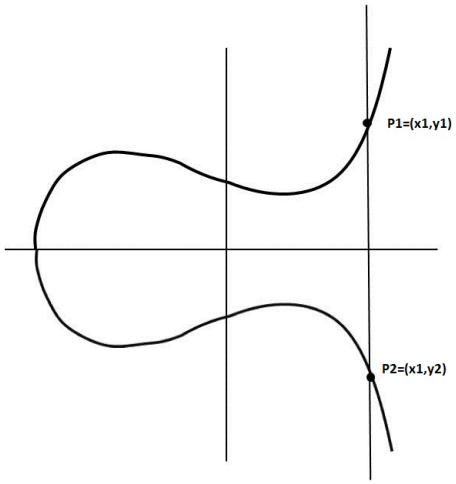


Figure 4: Vertical line and the point at infinity

4.4 Group Axioms

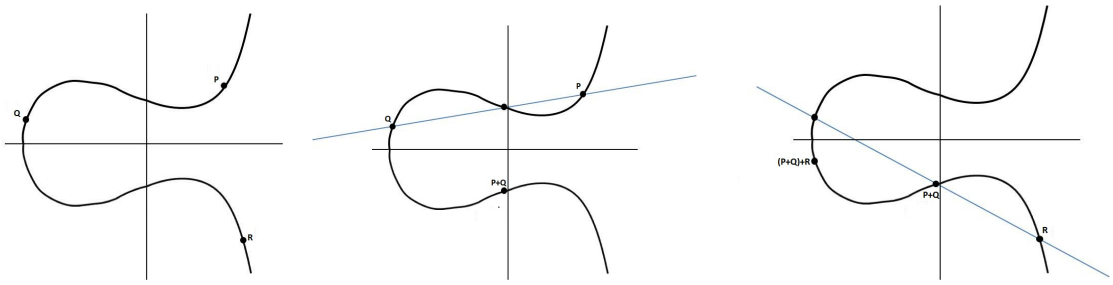
After defining the addition operator on the set of rational points on an elliptic curve, we would like to show that this operation defines a group. Recall that a set G together with a binary operation $*$ defined on it is called a *group* provided it satisfies the following conditions

1. $*$ is associative, i.e. for all $a, b, c \in G$, $(a * b) * c = a * (b * c)$
2. There is an identity for $*$, i.e. there exists $e \in G$ such that $e * g = g * e = g$ for all $g \in G$. (In additive notation: $0 + g = g$)
3. For every element $g \in G$, there is an element g^{-1}, called the inverse of g, such that $g * g^{-1} = g^{-1} * g = e$. (In additive notation: $g + (-g) = 0$)

If, moreover, the binary operation $*$ satisfies the condition $g * h = h * g$ for all $g, h \in G$, we say that $(G, *)$ is an abelian (or commutative) group. Let us discuss each of these conditions for the addition that we defined on the set E of rational points on an elliptic curve. First, we indeed have a well-defined binary operation. That is, we can always add two points P_1 and P_2 on the elliptic curve and the result is another point P_3 on the curve. If we examine the way we obtain P_3 we can see that it is chosen to be a point on the curve. We have seen a few cases when adding to points together, namely

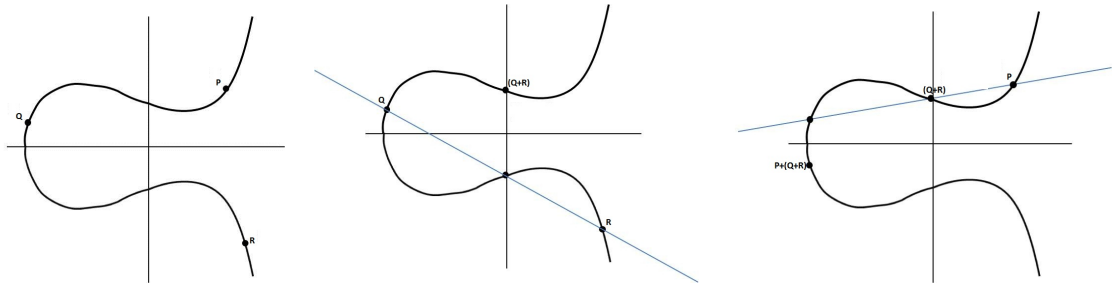
1. P_1 and P_2 are distinct and do not lie on a vertical line
2. $P_1 = P_2$
3. P_1 and P_2 lie on a vertical line

There are still a couple of cases to consider. How can we add an ordinary point P and $\mathcal{O}$? The line that goes through P and $\mathcal{O}$ is a vertical line. It intersects the curve at a point Q, and the reflection of Q across the x-axis is the point P. Therefore we define $P + \mathcal{O} = P$ for any point $P \neq \mathcal{O}$. How about $\mathcal{O} + \mathcal{O}$? You can probably guess that we define this sum as $\mathcal{O}$. In this case, we define a special line, called *the line at infinity* that intersects the curve three times at $\mathcal{O}$. Hence, we have $P + \mathcal{O} = P$ for all points P on the elliptic curve. It is also easy to see, both geometrically and algebraically, that this addition is commutative. So, we have $P + \mathcal{O} = P$ for all points $P \in E$. This means that the point $\mathcal{O}$ is the identity for this operation. What about inverses? Given a point $P = (x, y)$ on E what is its inverse $-P$? The inverse must have the property that when added to P the result is the identity. From our definitions so far it is not hard to see that $-P$ must be the reflection of P across the x-axis, i.e., $-P = (x, -y)$. The final axiom to be shown is associativity of this addition. It is tedious but not too hard to show this algebraically. Instead we will give a geometric illustration in the following figures (Figure 5 and Figure 6).



(a) A curve with 3 points P, Q, R (b) $P + Q$ (c) $(P + Q) + R$

Figure 5: Illustration of $(P + Q) + R$



(a) A curve with 3 points P, Q, R (b) $Q + R$ (c) $P + (Q + R)$

Figure 6: Illustration of $P + (Q + R)$

4.5 Additional Facts About Elliptic Curves over Finite Fields

We have seen that the set E of rational points on an elliptic curve forms a commutative group. For the purpose of cryptography we only consider curves over finite fields. A finite field has q elements for a prime power $q = p^r$ (p is a prime, r is a positive integer). Up to isomorphism there is only one finite field of order q, and it is denoted by $\mathbb{F}_q$ or $GF(q)$. Every finite field of order $q = p^r$ contains $\mathbb{Z}_p$, the field of integers $\mod p$, as a subfield. We say that the finite field $\mathbb{F}_q$, $q = p^r$ has *characteristic* p. This means that $p \cdot x = 0$ for all $x \in \mathbb{F}_q$. A field F for which there is no positive integer such that $p \cdot x = 0$ for all $x \in F$ is called a field of *characteristic zero*. Well-known examples of fields of characteristic zero are $\mathbb{R}$:real numbers, and $\mathbb{C}$: complex numbers. Any field of characteristic zero must contain $\mathbb{Q}$, the field of rational numbers. We call the fields $\mathbb{Z}_p$ and $\mathbb{Q}$ *prime fields*. Note that any field must contain one of these.

While the set E is infinite for curves over $\mathbb{R}$, it is finite for curves over finite fields. Let E be the group of rational points of the curve $y^2 = x^3 + ax + b \mod p$. Two natural questions to ask for the group E of an elliptic curve over a finite field field are

1. How many points does E contain?
2. What is the structure of the group E?

These questions are answered by the following two theorems that we state without proofs. However, we can get some intuition about these theorems. For the first question, to have a solution x for the equation $y^2 = x^3 + ax + b \mod p$, the quantity $x^3 + ax + b$ must be a square $\mod p$. It is well known that half of the non-zero

elements of $\mathbb{Z}_p$ are squares ($\frac{p-1}{2}$ of them). So, we would expect $x^3 + ax + b$ to be a square for about half of the elements of $\mathbb{Z}_p$. When that is the case (non-zero square), there are two square roots: y and $-y$. So, for about half of the elements of $\mathbb{Z}_p$ we get no solution for y, for the other half two solutions for y. Therefore, it would be reasonable to expect that the number solutions be around p. Hasse's theorem gives a more accurate answer with a proof.

Hasse's Theorem. Let E be the group of an elliptic curve over $\mathbb{Z}_p$. Then,

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p} \quad \text{or equivalently} \quad ||E| - p - 1| \leq 2\sqrt{p}$$

For the second question, since E is a finite abelian group, the Fundamental Theorem of Finite Abelian Groups says that it is isomorphic to a group of the form $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_k}$ where $m_i | m_{i+1}$ for all $i = 1, .., k-1$. The following theorem says for the group of an elliptic curve over $\mathbb{Z}_p^*$ we only need at most two factors in this decomposition.

Theorem 4.5.1 *Let E be the group of an elliptic curve over $\mathbb{Z}_p$ for some prime $p > 3$. Then E is isomorphic to $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ where $m_1 | m_2$ and $m_1 | (p-1)$.*

Exercise 4.5.2 *Determine what group is the group of elliptic curve defined by $y^2 = x^3 - 4x$ over $\mathbb{Z}_5$ isomorphic to? Justify your answer.*

5 Elliptic Curves in Cryptography

There are elliptic curves versions of many cryptosystems especially for those systems that involve the discrete logarithm problem. The main advantage of elliptic curves over other systems is that it is possible to achieve the comparable level of security with smaller primes. For example, solving elliptic curve version of the DLP for 163-bit primes is as hard as factoring 1024-bit integers (RSA system) [5]. This is useful for practical considerations. Moreover, key sizes in elliptic systems are required to be much smaller than key sizes required in many other systems. In using elliptic curves in cryptographic applications, we typically need an element of very large order. So we need a subgroup E of large order. This is not hard to find because by Theorem 4.5.1 E is either cyclic or is a direct product of two cyclic groups.

5.1 Elliptic Curve Version of the DLP

The classical Discrete Logarithm Problem is defined over a finite field where the group operation is multiplication. Since the group operation on an elliptic curve is

addition, the elliptic curve version of the DLP, ECDLP, is really an additive version of the classical, multiplicative DLP. The points of E in the elliptic curve version of the DLP play a role similar to that played by integers in the original DLP.

Exercise 5.1.1 *Formulate the elliptic curve version of the DLP.*

5.2 Elliptic Curve Version of ElGamal Cryptosystem

By changing the multiplicative group $\mathbb{Z}_p^*$ to the additive group of an elliptic curve E, we can directly modify the original ElGamal cryptosystem to obtain the elliptic curve version as follows:

First Bob chooses an elliptic curve $E \mod p$ for a large prime p, and chooses a point α on E (of large order). He also chooses a random and secret integer b and computes $\beta := b \cdot \alpha$. He then makes the following information public: (E, α, β).

Alice wants to send a message m to Bob. First she needs to express it as a point on E (we will discuss how to do this later). She chooses a random and secret integer a and computes $R := a \cdot \alpha$, $M := m + a \cdot \beta$. She sends the pair (R, M) to Bob.

To decrypt the Alice's message, Bob simply computes $M - bR$ using his secret key b. This works because $M - bR = m + a\beta - ba\alpha = m + ab\alpha - ab\alpha = m$.

As before, to be able to break the system, Eve needs to solve the DLP over the elliptic curve. Her life is harder this time because DLP over an elliptic curve of a given size p is harder than the DLP over a finite field of the same size.

5.3 Elliptic Curve Diffie-Hellman Key Exchange

Recall that we started this module with a description of Diffie-Hellman Key Exchange protocol whose security is based on the DLP. At this point it should not be hard for you to modify that protocol so that it is based on an elliptic curve.

Exercise 5.3.1 *Describe the elliptic curve version of the Diffie-Hellman Key Exchange protocol.*

5.4 Computational Complexity of the DLP and ECDLP

The key advantage of the elliptic key cryptosystem over earlier systems (such as RSA and ElGamal system over finite fields) is that we can achieve comparable security with a smaller key size, equivalently working with a smaller prime for the size q of the finite field $\mathbb{F}_q$. This is due to the fact that the best known algorithm for the ECDLP has the complexity of $O(\sqrt{q})$, which is exponential in the size of the

input $\log(q)$ (number of bits needed to store the integer q), $O\left(\sqrt{\exp(\log q)}\right) = O\left(\exp(\frac{1}{2}\log q)\right)$. On the other hand the best known method for solving DLP in a finite field (which is known as "index calculus") has subexponential complexity of $O(\exp(c(\log q)^{\frac{1}{3}})(\log(\log q))^{\frac{2}{3}})$, for some constant $c < 2$. For large values of q, the difference between these quantities is substantial. In fact,

Exercise 5.4.1 *Show that* $\lim_{q\to\infty} \frac{f(q)}{g(q)} = \infty$ *where* $f(q) = \sqrt{\exp(\log q)}$ *and* $g(q) = \exp\left(c(\log q)^{\frac{1}{3}})(\log^{\frac{2}{3}}(\log q)\right)$

Taking a specific value of c, say 1.5, you can verify that (as the following graph illustrates) $f(q) = g(q)$ around $q = 1.4046 \times 10^7$. For larger values of q the difference between f and g gets significantly large. Currently, for the RSA system and the DLP over a finite field, a secure value of q is assumed to be around $q = 2^{1024}$ (around a 1024-bit integer). However, the function f can attain the value of $g(2^{1024})$ at a much smaller value. For example, for the specific value of $c = 1.5$, you can verify that the ratio $\frac{f(2^{153})}{g(2^{1250})}$ is very close to 1 (≈ 1.07). As the key sizes are required to be larger due to technological advancements in computing speeds, the difference between these systems will be more significant.

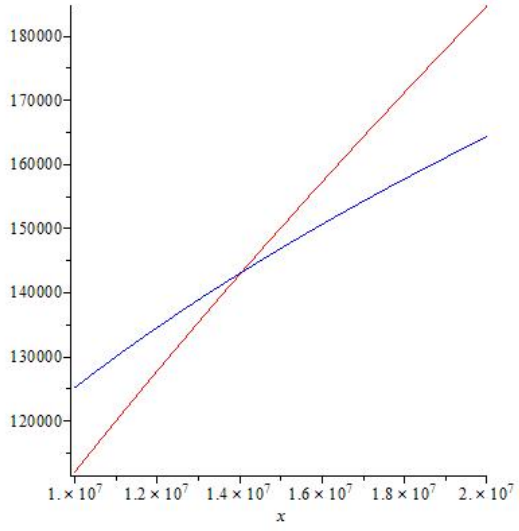


Figure 7: Comparisons of complexities of DLP (blue) and ECDLP (red) algorithms

5.5 Representation of Plaintext as Points on an Elliptic Curve

In earlier cryptosystems, we have needed to express a plaintext message as a number (modulo some large integer). For an elliptic curve implementation we need to express our message m as a point on an elliptic curve E, given by an equation of the form $y^2 = x^3 + cx + d$ over $\mathbb{Z}_p^*$. As a first step we convert the message to a number m. If there is a point on the elliptic curve with first component equal to m then we could use such a point as the representation of the plaintext. However, there may not be such a point on the curve because the quantity $m^3 + cm + d$ is not necessarily a square in the finite field $\mathbb{Z}_p^*$. It has about 50% chance of being a square. When $m^3 + cm + d$ is not a square, one way of finding a point on E whose x coordinate is obtained from m is to try integers $x_1, x_2, ...$ obtained from m by adjoining a few extra bits and check if $x_i^3 + cx_i + d$ is a square. If we try x_i for $i = 1, 2, ..., t$ then the probability of not being able to find a point on the curve is about $\frac{1}{2^t}$. Clearly, we can make this probability is small as we like by taking t large enough. Here is a method from Koblitz, described in [11]. Let t be an integer so that the failure rate of $\frac{1}{2^t}$ is acceptably low, and that $(m + 1)t < p$. Let $x_i = mt + i$ for $i = 0, 1, ..., t - 1$. For each x_i compute $x_i^3 + cx_i + d$ and see if it is a square. If it is, find one of its square roots y_i and let the point (x_i, y_i) represent the message m. If $x_i^3 + cx_i + d$ is not a square try the next value of x_i.

Exercise 5.5.1 *Explain how to recover the message m from the point (x, y) found using the algorithm above.*

Exercise 5.5.2 *Let E be the elliptic curve over $\mathbb{Z}_{31}$ given by the equation $y^2 = x^3 + 5x + 11$. Let $m = 5$ and $t = 5$. Show how to find the point on E that represents m according to the algorithm described above.*

5.6 A Maple Implementation of Elliptic Curve version of the ElGamal Cryptosystem

Here is one implementation of the ElGamal cryptosystem in Maple. Parts of the implementation are left as an exercise. Suppose Bob chooses the prime
$p := 47745199971245512067$, and the curve $E : y^2 = x^3 + cx + d$, where $c = 8$, and $d = 15$. He also chooses the element $\alpha = (25, 35838802608384708202)$ as the generator of a cyclic group. His secret key is $b = 437$.

```
> p := 47745199971245512067:
> c := 8: d := 15:
> eqn := x^3+c*x+d:
> Alpha := [25, 35838802608384708202]:
> b := 437:
> Bt := epower(Alpha, b, c, p);
          [29147934348216782453, 24974979216270861020]
```

So, his public key is
$(E, \alpha, \beta) = (E, (25, 35838802608384708202), (29147934348216782453, 24974979216270861020))$.
The function "epower" is a user-defined function that computes b-th "power of an element" α in the group of an elliptic curve, i.e. computes $b \cdot \alpha$. The definition of this function is left as an exercise.

Suppose Alice's plaintext message is "east", and her private key is $a = 1203$. She first converts the plaintext to a number, and test to see if that number satisfies the equation of E.

```
> a := 1203;
> pt := "east";
                              "east"
> ptN := convert(pt, bytes);
                       [101, 97, 115, 116]
> m := convert(ptN, base, 256, p);
                           [1953718629]
> yc := `mod`(subs(x = m[1], eqn), p);
                      44076639168269100330
> `mod`(Power(yc, (p-1)*(1/2)), p);
                                1
```

Luckily it does (after all there is a 50% chance! (But it would not have been a big deal if it did not. We know how to handle that situation). Note that if $y^{\frac{p-1}{2}} = 1 \mod p$, then y is a square. Next, she finds the square root of 44076639168269100330. Since $p \equiv 3 \mod 4$, the square roots of y are given by $\pm x$, where $x = y^{\frac{p+1}{4}} \mod p$.

```
> syc := `mod`(Power(xc, (p+1)*(1/4)), p);
                      45622981855386415246
> mess := [m[1], syc];
              [1953718629, 45622981855386415246]
```

So, the point that represents her message on the elliptic curve is $(1953718629, 45622981855386415246)$. She then computes

```
> R := epower(Alpha, a, c, p);
          [5578279319746943067, 39756276951217044356]
> M1 := epower(Bt, a, c, p);
          [33481033851846984254, 9651503015562240975]
> M := ecadd(M1, mess, c, p);
          [21022544373167168187, 20202453460108310425]
```

Here, the function "ecadd"' is a user defined function that is used to add to points on an elliptic curve. Its definition is left as an exercise. So, Alice sends the pair $(R, M) =$
$([5578279319746943067, 39756276951217044356], [21022544373167168187, 20202453460108310425])$.
Finally, Bob does the following computations to decipher her message.

```
> Rb := epower(R, b, c, p);
          [33481033851846984254, 9651503015562240975]
> NegRb := [Rb[1], `mod`(p-Rb[2], p)];
          [33481033851846984254, 38093696955683271092]
> Dc := ecadd(M, NegRb, c, p);
               [1953718629, 45622981855386415246]
> decipherMod256 := convert([Dc[1]], base, p, 256);
                      [101, 97, 115, 116]
> decipherText := convert(decipherMod256, bytes);
                            "east"
```

5.7 Efficient Computation of Multiples of a Point

Recall that efficient implementation of the exponentiation, that is computing quantities of the form $m^e \mod n$ for large integers m, n and e, was crucial for a practical implementation of the RSA cryptosystem. In [1] we noted that the straightforward method of computing exponents was not efficient. Instead, there was an efficient method called "Square and Multiply" algorithm. In the case of an elliptic curve cryptosystem, our operation is addition instead of multiplication (additive group of

E vs. multiplicative group of $\mathbb{Z}_n^*$). The additive version of the operation "$m^e \mod n$ in $\mathbb{Z}_n^*$" is "$k \cdot P$ for some point $P \in E$ and integer k". You might wonder if there is a version of the "Square and Multiply" algorithm for elliptic curves. Yes, there is and quite naturally it is called Double-and-Add algorithm in which the multiplicative operation of squaring $m \to m^2$ is replaced with the additive version of doubling: $P \to 2P$.

Exercise 5.7.1 *Describe the Double-and-Add algorithm.*

It turns out that we can modify this the Double-and-Add algorithm to make it more efficient. This more efficient algorithm is called Double-and-(Add OR Subtract) algorithm. The description below is adopted from [10]. This algorithm is based on a *signed binary representation* of an integer k given by $k = \sum_{i=0}^{t} k_i 2^i$ where each k_i is 0, 1 or -1. We sometimes use the vector $(k_t, \ldots, k_0)$ to denote the representation where the right-most digit is the coefficient of the lowest order term. Whereas the standard binary representation of an integer is unique, a signed binary representation is not. In fact, every binary representation is also a signed binary representation. For example, $(0, 1, 1, 0, 1)$, $(1, 0, 0, -1, -1)$ and $(1, 0, -1, 0, 1)$ are all signed binary representations of 13. By remembering that subtraction is nothing more than addition with the additive inverse, i.e. $P - Q$ is the same as $P + (-Q)$, modify the Double-and-Add algorithm to obtain Double-and-(Add OR Subtract) algorithm.

Exercise 5.7.2 *Describe the Double-and-(Add OR Subtract) algorithm.*

Although a signed binary representation of an integer is not unique, there is a special form of it that is unique. A signed representation $(k_t, \ldots, k_0)$ of k is said to be in *non-adjacent form* if no two consecutive k_i's are non-zero. Such a representation is called a NAF representation. For example, of the three representations of 13, $(1, 0, -1, 0, 1)$ is the only one in NAF form. It is not too difficult to convert a standard binary representation of an integer into a NAF representation.

Exercise 5.7.3 *Describe an algorithm to convert a standard binary representation of an integer into NAF form. (Hint: Replace strings of the form* $(0, 1, 1, \ldots, 1)$ *by* $(1, 0, \ldots, 0, -1)$ *and show that this substitution does not change the value of the number.)*

Exercise 5.7.4 *Find the standard binary representation of the integer 749, then apply your algorithm to find its representation in NAF form.*

Exercise 5.7.3 shows that every positive integer has a representation in NAF form. It is also possible to show that such a representation is unique.

Exercise 5.7.5 *Show that a signed binary representation of a positive integer in NAF form is unique.*

Given this result, we can speak of the NAF representation of a positive integer. But what is the benefit of the NAF representation? Because of the requirement that no two consecutive digits can be non-zero in the NAF representation, we expect that on average the NAF representations of integers contain more zero bits than the standard binary representation. In fact, it can be shown that on average a t-bit integer contains $\frac{t}{2}$ zeroes in its binary representation but $\frac{2t}{3}$ zeroes in its NAF representation. This reduces the number of operations needed to implement the Double-and-Add algorithm.

Exercise 5.7.6 *Assuming the result stated above about expected number of 0's in the two representations, and assuming that doubling takes about the same amount of time as adding or subtracting, quantify the expected amount of speed up by switching to the NAF representation from the binary representation.*

In the projects section at the end of the module, you are asked to write an algorithm to compute multiples of a point on an elliptic curve based on the NAF representation. We close the module by investigating some combinatorial properties of the NAF representation.

Exercise 5.7.7 *Let N_i denote the set of positive integers that have exactly i coefficients in their NAF representation. (Note that the leading term in a NAF representation must be 1). Let $n_i = |N_i|$. Show that*

a) $n_1 = 1$, $n_2 = 1$, and n_i's satisfy the following recurrence relation

$$n_{i+1} = 2(n_1 + n_2 + \cdots + n_{i-1}) + 1 \text{ for } i \geq 2.$$

b) Derive a second degree recurrence relation for n_i, and obtain an explicit formula for n_i.

c) Fill in the following table

i	n_i
5	
10	
25	
50	

6 Projects

6.1 Computational Projects

1. Write a Maple procedure to perform addition on a given elliptic curve

2. Write a Maple procedure to compute the "power" of an point P on an elliptic curve, that is, given a point P on an elliptic curve E and a positive integer k the procedure computes $k \cdot P = \underbrace{P + P + \cdots + P}_{k-times}$. Use the most efficient method we discussed based on the NAF representation of the integer k.

3. Use the two functions you wrote above, to implement the ElGamal cryptosystem in Maple on a specific example. Choose a prime p that is as large as possible and that Maple can still handle the computations in short amounts of times (say no more than 2 minutes per computation). Choose an elliptic curve over $\mathbb{Z}_p$ to implement the algorithm. Do at least two versions of the exercise. In one version the prime p should be $p \equiv 3 \mod 4$, and in the other version take $p \equiv 1 \mod 4$.

6.2 Research Projects

You may find the book [7] as a useful reference book and a good starting point for these research problems

1. Research the index calculus method to solve the the discrete logarithm problem in a finite field. Write a paper describing the algorithm. As a bonus, implement the algorithm on Maple.

2. Research the attacks on elliptic curve cryptosystems. What precautions should be taken in a practical implementation of an elliptic curve cryptosystem?

3. Research the integer factorization algorithms that use elliptic curves. Write a paper describing the algorithms. As a bonus, implement the algorithm on Maple.

4. Find an actual, real-life use of a cryptosystem based on elliptic curves. Describe the system in as much detail as you can.

7 Solutions to Exercises

2.0.1 Eve finds a from α^a and b from α^b. She then computes $k = \alpha^{ab}$.

2.0.2 Here is a solution

- Alice, Bob and Carol agree on a large prime p, and a generator α of $\mathbb{Z}_p^*$.
- Alice, Bob and Carol generate random, private integers a, b, c respectively.
- Alice computes α^a and sends it Bob.
- Bob computes $(\alpha^a)^b = \alpha^{ab}$ and sends it Carol.
- Carol computes $(\alpha^{ab})^c = \alpha^{abc}$ and uses it as her secret key.
- Carol computes α^c and send it to Alice and Bob.
- Alice computes $(\alpha^c)^a = \alpha^{ac}$ and sends it Bob.
- Bob computes $(\alpha^{ac})^b = \alpha^{abc}$ and uses it as his secret key.
- Bob computes $(\alpha^c)^b = \alpha^{bc}$ and sends it to Alice.
- Alice computes $(\alpha^{bc})^a = \alpha^{abc}$ and uses it as her secret key.

3.1.1 Bob simply computes $M \cdot R^{-b}$. This correctly recovers the original message because $M \cdot R^{-b} = \beta^a m(\alpha^a)^{-b} = (\alpha^b)^a m\alpha^{-ab} = m\alpha^{ab}\alpha^{-ab} = m$

3.2.1 Signature verification scheme is correct because: Since $S \equiv k^{-1}(m - aR) \mod p-1$, we have $m \equiv Sk + aR \mod p-1$. Then
$V_1 = \beta^R R^S \equiv (\alpha^a)^R(\alpha^k)^S \equiv \alpha^{Sk+aR} \equiv \alpha^m \equiv V_2$. Here we used the fact that for a prime p, if $x \equiv y \mod p-1$ then $\alpha^x \equiv \alpha^y \mod p$.

4.1.1 First note that the set of squares over $\mathbb{Z}_7$ is $\{0, 1, 2, 4\}$. By checking each element of $\mathbb{Z}_7$ we see that for $x = 0, 2, 5$ we get $y^2 = 0$, hence $(0, 0), (2, 0), (5, 0)$ are on the curve. For $x = 4$ and $x = 6$ we have no solutions because we get $y^2 = 6$ and $y^2 = 3$ respectively, which are not squares in $\mathbb{Z}_7$. For $x = 1$, $y^2 = 4$ so $(1, 2)$ and $(1, 5)$ are on the curve. Finally, for $x = 3$, $y^2 = 1$ hence the points $(3, 1)$ and $(3, 6)$ are on the curve. Therefore, the set of all points on this curve are $\{(0, 0), (1, 2), (1, 5), (2, 0), (3, 1), (3, 6), (5, 0), \mathcal{O}\}$.

4.3.2 Implicitly differentiating $y^2 = x^3 + ax + b$, we get $2yy' = 3x^2 + a$, then $y' = \frac{3x^2+a}{2y}$. Substituting the point (x_1, y_1) into this equation we obtain the slope at P_1 as $M = \frac{3x_1^2+a}{2y_1}$. The rest of the computations is similar to the previous case.

4.5.2 We know that the order of this group is 4. Up to isomorphism, there are only groups of order 4: $\mathbb{Z}_4$, the cyclic group of order 4, and $\mathbb{Z}_2 \times \mathbb{Z}_2$, the Klein four-group. To distinguish between the two notice that $P + P = \mathcal{O}$ for all E. Therefore, it is isomorphic to the Klein four-group.

5.1.1 Given two points P, Q on an elliptic curve E, and given that $Q = kP = \underbrace{P + P + \cdots + P}_{k-times}$ for some positive integer k, find the value of k.

5.3.1 Alice and Bob agree on an elliptic curve E, and a generator α of E (this information can even be public). Then each one of them chooses a secret integer a, and b respectively. Alice sends $x = a\alpha$ to Bob, and Bob sends $y = b\alpha$ to Alice. Alice computes $ay = ab\alpha$ and Bob computes $bx = ba\alpha$, and they end up with the same key $k = ab\alpha$.

5.4.1 This is a Calculus problem. It suffices to show that (a simplified version) $\lim_{x\to\infty} \left(\log(x) - \sqrt{\log(x)}\right) = \infty$. We can show this using algebraic manipulations such as conjugation etc.

5.5.1 The message is $m = \lfloor \frac{x}{t} \rfloor \mod p$.

5.5.2 First note that squares $\mod 31$ are $\{0, 1, 2, 4, 5, 7, 8, 9, 10, 14, 16, 18, 19, 20, 25, 28\}$. First we compute $5^3 + 5 \cdot 5 + 11 = 6 \mod 31$, which is not a square. Then let $x_0 = 5 \cdot 5 = 25$, and compute $x_0^3 + 5 \cdot x_0 + 11 = 13 \mod 31$, which is still not a square. Next, let $x_1 = 26$, and compute $x_1^3 + 5 \cdot x_1 + 11 = 16 \mod 31$. Clearly, 16 is a square and its square roots $\mod 31$ are 4 and $-4 = 27$. We can use the pair $(26, 4)$ as a representative of the message $m = 5$ on E. Note: For primes p such that $p \equiv 3 \mod 4$, there is an easy way of computing square roots of an element y: Let $x = y^{\frac{p+1}{4}} \mod p$. Then the square roots of y are $\pm x$.

5.7.1 Given a point P on an elliptic curve E and a positive integer b with binary representation $b = \sum_{i=0}^{t} b_i 2^i$, the algorithm is as follows:

1. Set $Q \leftarrow \mathcal{O}$

2. for i from 0 to t do

   (a) $Q \leftarrow 2Q$

   (b) If $b_i = 1$, set $Q \leftarrow Q + P$

3. Return Q.

5.7.2 Given a point P on an elliptic curve E and a positive integer b with a signed binary representation $b = \sum_{i=0}^{t} b_i 2^i$, the algorithm is as follows:

1. Set $Q \leftarrow \mathcal{O}$

2. for i from 0 to t do

   (a) $Q \leftarrow 2Q$

   (b) If $b_i = 1$, set $Q \leftarrow Q + P$

   (c) If $b_i = -1$, set $Q \leftarrow Q - P$

3. Return Q.

5.7.3 Apply the substitution given in the hint as many times as needed starting from the right-most digits proceeding to the left. Note that $2^i + 2^{i-1} + \cdots + 2^j = 2^{i+1} - 2^j$.

5.7.4 The standard binary representation of 749 is $(1, 0, 1, 1, 1, 0, 1, 1, 0, 1)$, and its NAF form is $(1, 0, -1, 0, 0, 0, -1, 0, -1, 0, 1)$.

5.7.5 Suppose a positive integer m has two representations in NAF form. First note that the leading terms must be 1. Then show that leading terms must be located in the same positions. Next consider the first term (from the right) in which the two representations differ. Considering all possible cases (1 and 0, 1 and -1, etc in that position) show that it is not possible to have different digits in any position.

5.7.6 For the standard binary representation, we need, on average, t doublings and $\frac{t}{2}$ additions for a total of $\frac{3t}{2}$ operations. In the NAF representation, we need, on average, t doublings and $\frac{t}{3}$ additions or subtractions, for a total of $\frac{4t}{3}$ operations. The ratio $\frac{9}{8}$ shows that we obtain a reduction of about 11% in the number of operations needed.

5.7.7 a) It is easy to see that $n_1 = n_2 = 1$. To obtain the recurrence relation, note that a NAF expansion of the form $(1, *, *, \ldots, *)$ with a 1 at the $(i-1)$st position can be expanded to a NAF expansion of length $i+1$ in exactly two ways: $(1, 0, 1, *, *, \ldots, *)$ and $(1, 0, -1, *, *, \ldots, *)$. The same is true for NAF expansions with leading coefficients at the positions $i-2, i-3, \ldots, 1$. Finally, we also have the expansion $(1, 0, \ldots, 0)$ with a single 1 at the $(i+1)$st position that cannot be obtained from any of the smaller expansions.

b) The recurrence relation is $n_{i+1} = n_i + 2n_{i-1}$. The explicit solution is

$$n_i = \frac{(1+\sqrt{2})^{i-1}}{2} + \frac{(1-\sqrt{2})^{i-1}}{2}$$

c)

i	n_i
5	17
10	1393
25	768398401
50	2850877693509864481

References

[1] N. Aydin "Public Key Cryptography and the RSA Cryptosystem", A computational science module, supported by NSF CCLI DUE-0618252, http://www.capital.edu/cs-math/, 2009.

[2] W. Diffie and M. E. Hellman. "New directions in cryptography". *IEEE Transactions on Information Theory*. 22 (6): 644-654, 1976

[3] T. ElGamal."A public key cryptosystem and a signature scheme based on discrete logarithms". *IEEE Transactions on Information Theory*. 31 (4): 469-472, 1985.

[4] N. Koblitz. "Elliptic curve cryptosystems". *Mathematics of Computation* 48 (177): 203209, 1987.

[5] K. Lauter. "The advantages of elliptic curve cryptography for wireless security". *IEEE Wireless Communications Magazine*, 11(1), 62-67, 2004.

[6] H. W. Lenstra Jr. "Factoring integers with elliptic curves". *Annals of Mathematics* 126 (3): 649673, 1987.

[7] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, Washington D.C., 1997.

[8] V. Miller V. "Use of elliptic curves in cryptography". *CRYPTO* 85: 417426, 1985.

[9] J. H. Silverman and J. Tate. *Rational Points on Elliptic Curves*. Springer-Verlag, New York, 2010.

[10] D. R. Stinson. *Cryptography: Theory and Practice*, 3rd ed. Chapman & Hall/CRC, Boca Raton, London, New York, 2006.

[11] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*, Prentice Hall, Upper Saddle River, NJ, 2002.

[12] A. Wiles. "Modular elliptic curves and Fermat's Last Theorem". *Annals of Mathematics* 141 (3): 443551, 1995.