

# Introduction to Elliptic Curve Cryptography

Elisabeth Oswald  
Institute for Applied Information Processing and Communication  
A-8010 Inffeldgasse 16a, Graz, Austria  
Elisabeth.Oswald@iaik.at

July 29, 2005

## Abstract

This document should be considered as a tutorial to elliptic curve cryptography. It is assumed that the reader has a basic understanding of cryptography and additionally, has a basic understanding of abstract algebra and elementary number theory. A lot of different aspects are discussed, but lots of things are spared. The motivation of this document is to provide the reader with facts about the use of elliptic curves in cryptography, so lots of (beautiful) maths is not included for the sake of concreteness. Enjoy reading!

## 1 Why are we interested in elliptic curve cryptography?

Since the beginning of public key cryptography there are two major cryptosystems (*RSA* and *El-Gamal*) that seem to defeat all attacks. For this reason, these two cryptosystems are the most respected and wildly used public key cryptosystems nowadays. One can use both cryptosystems for encryption/decryption and digital signatures. All important security standards cover those cryptosystems, so it should be safe to use implementations of them. The question is now, why do we look for new cryptosystems?

- Is it only mathematical curiosity?
- Do we want to make new products and earn more money?
- Do we know more about ECC and can make cryptosystems with trapdoors to spy out others?
- Are we paid by Certicom? ...
- or is there a real need for new cryptosystems?

First of all, elliptic curve cryptosystems (ECC) aren't that new anymore. They were invented around 1985 independently by Miller and Koblitz. Since their introduction a broad discussion on their security and efficiency has been carried on. It is this very efficiency that makes them so interesting for us today. This is due to the fact that information technology is developing very fast. For example, most computers today don't look like the old fashioned personal computers anymore. We use handhelds, and mobile phones and of course we have a need in securing communications on these devices. But in this case there have to be several constraints taken into account: there's very limited memory and computing power on these devices, and

it is not possible to spend much bandwidth for communications overhead. What we need is a cryptosystem with small keys, and a small signature size. Efficient encryption/decryption is not so important because these operations are usually done with a private key cryptosystem.

ECC has exactly the desired properties. This comes from the fact, that there are no subexponential algorithms for the ECDLP (elliptic curve discrete logarithm problem) known today. This means that we can use shorter keys (compared to other cryptosystems) for high security levels.

In the next sections we will discuss several aspects of elliptic curves and cryptography. First we will treat them more as mathematical objects, and then we investigate more cryptography related stuff. Everything is written in a cryptographers view. This means, we know some properties that a cryptosystem must have, and we assert that they hold for ECC .

## 2 Basic Facts about Elliptic Curves

We investigate what an elliptic curve is, and what it's (algebraic) properties are. Please note that although we often work in finite fields we write "=" instead of "≡". This is because we use the same notation as in the books of N. Koblitz ([6],[7]). If we talk about derivatives in the context of a finite field, we mean the "formal derivative" which is defined through the  $nX^{n-1}$  rule (of course not as a limit). When we represent congruence classes we choose the least positive number in the class as a representative.

**Definition 1.** *An elliptic curve  $E$  over the field  $\mathbb{F}$  is a smooth curve in the so called "long Weierstrassform"*

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6, \quad a_i \in \mathbb{F}. \quad (1)$$

We let  $E(\mathbb{F})$  denote the set of points  $(x, y) \in \mathbb{F}^2$  that satisfy this equation, along with a "point at infinity" denoted  $\mathcal{O}$ .

Remember that smooth means that there is no point in  $E(\overline{\mathbb{F}})$  where both partial derivatives vanish. The definition given above is valid for any field. But in cryptography we are only interested in finite fields. Considering only finite fields we get an "easier" equation. Two finite fields are of particular interest. The finite field  $\mathbb{F}_p$  with  $p \in \mathbb{P}$  elements, because of it's structure, and the finite field  $\mathbb{F}_{q^m}$  with  $q = p^r$  Elements, since setting  $p = 2$  the arithmetic in this field will be well suited for implementations in hardware.

### 2.1 Elliptic Curves over Prime Finite Fields

We start with  $\mathbb{F}_p$  ( $p \in \mathbb{P}, p > 3, \text{char}(\mathbb{F}_p) \neq 2, 3$ )<sup>1</sup> and perform the following change of variables :

$$\begin{aligned} X &\longrightarrow X - \frac{a_2}{3} \\ Y &\longrightarrow Y - \frac{a_1X + a_3}{2} \end{aligned}$$

<sup>1</sup>If adding the multiplicative identity 1 to itself in  $\mathbb{F}$  never gives 0 then we say  $\mathbb{F}$  has characteristic 0; in this case  $\mathbb{F}$  contains a copy of  $\mathbb{Q}$ . Otherwise, there is a prime number  $p$  such that  $pn = 0$ , for all  $n \in \mathbb{F}$ , and  $p$  is called characteristic of the field  $\mathbb{F}$ . In that case  $\mathbb{F}$  contains a copy of the field  $\mathbb{F}_p$ , which is called its prime field.

Let's take a look what is happening to the left side after the substitution for  $Y$  :

$$(Y - (a_1X + a_3)/2)^2 + a_1X(Y - (a_1X + a_3)/2) + a_3(Y - (a_1X + a_3)/2) = \dots$$

$$\dots = Y^2 - a_1^2X^2/4 - a_1a_3X/2 - a_3^2/4$$

Both,  $XY$  and  $Y$  have vanished, so their coefficients  $a_1$  and  $a_3$  must equal zero! That reduces the left side to a single  $Y^2$ . If we make the substitution for  $X$  and take a look at the right side of (1) we get :  $(X - a_2/3)^3 + a_2(X - a_2/3)^2 + a_4(X - a_2/3) + a_6 = \dots$

$$\dots = X^3 + (a^2/9 + a_4)X + 2a_2^3/27 - a_2/3a_4a_6.$$

Setting  $(\frac{1}{9}a^2 + a_4) = a$  and  $\frac{2}{27}a_2^3 - \frac{1}{3}a_2a_4a_6 = b$  we have the much nicer form  $X^3 + aX + b$ . In  $\mathbb{F}_p$  equation (1) reduces to

$$Y^2 = X^3 + aX + b. \tag{2}$$

What can we say about the smoothness of this equation? Consider the partial derivative of the equation  $y^2 = f(x)$ , which is  $f'(x) = 2y \frac{dy}{dx}$ . The expression of  $\frac{dy}{dx}$  is undefined in  $(x_0, y_0)$  if and only if  $f'(x_0) = f(x_0) = y_0 = 0$ . In other words, the function  $f(x)$  must have a multiple root at the point  $x_0$ . In the case that  $f(x) = x^3 + ax + b$ , this is equivalent to  $disc(f(x)) = -(4a^3 + 27b^2) = 0$ . We give now our definition for an elliptic curve over the finite field  $\mathbb{F}_p$ :

**Definition 2.** An elliptic curve  $E$  over the finite field  $\mathbb{F}_p$  is given through an equation of the form

$$Y^2 = X^3 + aX + b, \quad a, b \in \mathbb{F}_p, \quad \text{and} \quad -(4a^3 + 27b^2) \neq 0 \tag{3}$$

Please note that as stated in the beginning of the section, the "=" should be replaced by an "≡" in the above definition. Another remark is that when we talk about partial derivatives we mean the "formal partial derivate" which can be defined (see beginning of this section) over an arbitrary field.

## 2.2 Elliptic Curves over Binary Finite Fields

Now we work in the field  $GF(2^m)$  where we have characteristic  $\neq 2$ . Here we only consider so called "nonsupersingular curves". They have the property  $a_1 \neq 0$ . So we can make the following change of variables:

$$X \longrightarrow a_1^2X + \frac{a_3}{a_1}$$

$$Y \longrightarrow a_1^3Y + \frac{a_1^2a_4 + a_3^2}{a_1^3}$$

This leads us to the following definition.

**Definition 3.** A (nonsupersingular) elliptic curve  $E$  over the finite field  $\mathbb{F}_{2^m}$  is given through an equation of the form

$$Y^2 + XY = X^3 + aX^2 + b, \quad a, b \in \mathbb{F}_{2^m}. \tag{4}$$

Before starting with the arithmetic of the points on an elliptic curve, we take a final look at the coefficients in equation (1). The subscripts of these coefficients seem to be a little bit strange. But consider following : For big values of  $X$  we can say that the equation is very close to  $F : Y = X^{3/2}$ . This function can be parameterized by setting  $X = T^2; Y = T^3$ . One says, "X has degree 2" and "Y has degree 3". The subscripts of the coefficients in equation (1) indicate the degrees that must be given to the coefficients in order that the equation be homogeneous (this means that each term has the same total degree which is 6 in this case).

## 2.3 Addition Law

In order to define a cryptosystem on the set of points on an elliptic curve, we need to define an algebraic structure on the points. The easiest algebraic structure which provides us with all necessary tools is the group. Therefore we need to define a neutral element, inverse elements, and the addition of two elliptic curve points which needs to be associative.

**Definition 4.** Let  $E$  be an elliptic curve over  $\mathbb{F}_p$  or  $\mathbb{F}_{2^m}$ , and let  $P$  and  $Q$  be two points on  $E$ .

1. **Zero element:** If  $P$  is the point  $\mathcal{O}$ , then we define  $-P$  to be  $\mathcal{O}$ . For any point  $Q$  we define  $\mathcal{O} + Q$  to be  $Q$ . In  $\mathbb{F}_p$  we can visualize  $\mathcal{O}$  as sitting infinitely far up the  $y$ -axis (in the next section we give a more natural way to introduce this point at infinity).
2. **Inverse element:** In  $\mathbb{F}_p$  we define the negative of the point  $P = (x, y)$  to be  $-P = (x, -y)$ . If  $Q = -P$ , then we define  $P + Q = \mathcal{O}$ . For  $\mathbb{F}_{2^m}$  we define  $-P = (x, x + y)$ .
3.  **$P+Q$ :** If  $P \neq Q$ , then we shall soon show that the line  $l = \overline{PQ}$  intersects the curve in exactly one more point  $R$ . Then we define  $P + Q$  to be  $-R$ , that is the inverse of the third point of intersection.
4.  **$2P$ :** Let  $l$  be the tangent line to the curve at  $P$ , let  $R$  be the only (the third) point of intersection of  $l$  with the curve, and define  $2P = -R$ .

This set of rules can be summarized in the following succinct manner:

**The sum of the three points where a line intersects the curve is zero.**

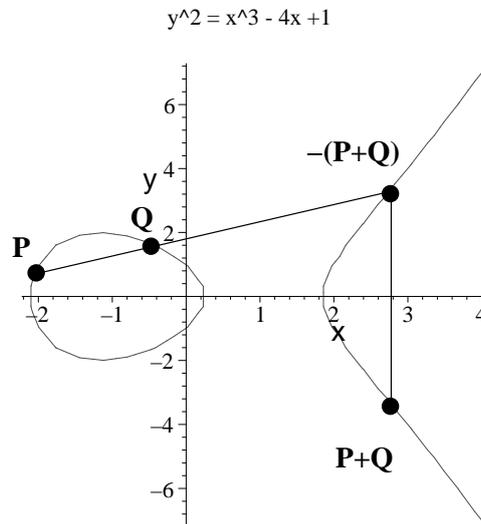


Figure 1: Point addition

We now show, why there is exactly one more point where the line  $l$  through  $P$  and  $Q$  intersects the curve, and we derive formulas for the point addition. We restrict our calculations

here on the field  $\mathbb{F}_p$  because the other case can be treated in the same fashion. Let  $P = (x_1, y_1), Q = (x_2, y_2), R = (x_3, y_3)$ . We'd like to express  $x_3$  and  $y_3$  in terms of  $x_1, y_1, x_2, y_2$ . We first discuss the case where  $P \neq Q$ . Let  $y = \alpha x + \beta$  be the equation of the line through  $P$  and  $Q$ . Then we have  $\alpha = \frac{y_2 - y_1}{x_2 - x_1}$ , and  $\beta = y_1 - \alpha x_1$ . A point  $(x, \alpha x + \beta)$  lies on the elliptic curve if and only if  $(\alpha x + \beta)^2 = x^3 + ax + b$ . Thus, there is one intersection point for each root of the cubic equation. We already know the two roots  $x_1$  and  $x_2$ , because they correspond to the points  $P$  and  $Q$  on the curve. Since there are at most three roots of the cubic equation we conclude that the third root must equal  $x_3$ . It is easy to show that the sum of the roots of a monic polynomial is equal to minus the coefficient of the second-to-highest power, so we conclude that this third root is  $x_3 = \alpha^2 - x_1 - x_2$ . (Hint: Compare the coefficients of the equations  $(x - x_1)(x - x_2)(x - x_3)$  and  $x^3 - (\alpha x + \beta)^2 + ax + b$ ). We also know that  $y_3 = -(\alpha x_3 + \beta) = \alpha(x_1 - x_3) - y_1$ .

The case when  $P = Q$  is similar, except that  $\alpha$  is now the derivative  $dy/dx$  at  $P$ . Implicit differentiation of equation (3) leads to  $\alpha = (3x_1^2 + a)/2y_1$ , and we obtain the formulas for the coordinates of  $2P$ . The following table lists all obtained formulas together with the formulas for  $\mathbb{F}_{2^m}$ :

	$\mathbb{F}_p$	$\mathbb{F}_{2^m}$
P+Q	$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$ $y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$	$x_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a$ $y_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1$
2P	$x_3 = ((3x_1^2 + a)/2y_1)^2 - 2x_1$ $y_3 = -y_1 + ((3x_1^2 + a)/2y_1)(x_1 - x_3)$	$x_3 = x_1^2 + \frac{b}{x_1^2}$ $y_3 = x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right)x_3 + x_3$
-P	$(x, -y)$	$(x, x + y)$

Table 1: Addition formulas in affine coordinates

Now we have to prove, that the above definition of  $P + Q$  makes the points on an elliptic curve into an (abelian) group. The only group law that is not an immediate consequence of the geometrical rules is the associative law. It can be proved with following proposition (without proof).

**Proposition 1.** *Let  $l_1, l_2, l_3$  be three lines that intersect a cubic in nine points  $P_1, \dots, P_9$  (counting multiplicity), and let  $l'_1, l'_2, l'_3$  be three lines that intersect the cubic in nine points  $Q_1, \dots, Q_9$ . If  $P_i = Q_i$  for  $i = 1, \dots, 8$ , then also  $P_9 = Q_9$ .*

To proof the associativity we choose three points on  $E$ , namely  $P, Q$  and  $R$  and let

$$\begin{aligned}
 l_1 &= \overline{P, Q, -(P + Q)} \\
 l_2 &= \overline{\mathcal{O}, R, -R} \\
 l_3 &= \overline{-P, -(Q + R), P + (Q + R)} \\
 l'_1 &= \overline{Q, R, -(Q + R)} \\
 l'_2 &= \overline{\mathcal{O}, P, -P} \\
 l'_3 &= \overline{-R, -(P + Q), (P + Q) + R}.
 \end{aligned}$$

We just proof the case where  $P, Q$  and  $R$  are distinct points. We want to show that  $P + (Q + R) = (P + Q) + R$ . We have three lines  $l_1, l_2, l_3$  that intersect the curve in nine points.

Also the three lines  $l'_1, l'_2, l'_3$  intersect the curve in nine points. If we count how many of these points are equal, we count eight different points:  $P, Q, P + Q, -(P + Q), R, Q + R, -(Q + R)$  and  $\mathcal{O}$ . According to the proposition the ninth points also have to be equal, so  $P + (Q + R) = (P + Q) + R$  must hold.

### 3 Projective Space

In the last section we introduced the "point at infinity" in a rather unappealing way. In projective space we get that point very naturally. We first give the definition of the projective  $n$ -dimensional space, but continue working with the projective plane (we don't need higher dimensions).

**Definition 5.** *The projective space (over a field  $K$ ) is the set of equivalence classes of tuples  $(X_0, X_1, \dots, X_n)$  (not all components zero) where two tuples are said to be equivalent if they are scalar multiples of one another, i.e.*

$$\mathbb{P}^n(K) = \{(X_0, X_1, \dots, X_n) - (0, 0, \dots, 0) \mid (tX_0, tX_1, \dots, tX_n) \sim (X_0, X_1, \dots, X_n), t \in K \setminus \{0\}\}$$

For example the *projective plane* (over  $\mathbb{Z}$ ) is the set of equivalence classes  $(X, Y, Z)$  (not all components zero), where two triples are said to be equivalent if they are scalar multiples of one another. For instance,  $(X, Y, Z) \sim (2X, 2Y, 2Z) \sim (5X, 5Y, 5Z) \sim (nX, nY, nZ), n \in \mathbb{Z} \setminus \{0\}$ . Such an equivalence class is called a *projective point*. Another example is the projective line  $\mathbb{P}^1(\mathbb{R})$ . It is the set of points  $(x, y)$  excluding  $(0, 0)$  with the points  $(tx, ty)$  identified with  $(x, y)$ . If we select  $P = (x, y)$ , then all the points  $(tx, ty)$  are on the line joining  $P$  to the origin. This is visualized in figure 2.

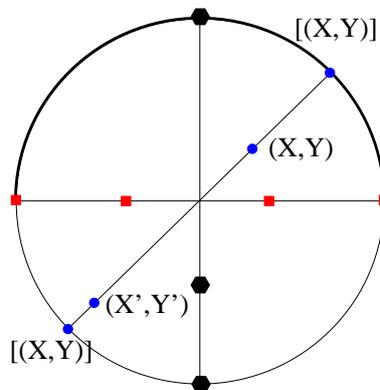


Figure 2: Projective Line

Points with the same shape are equivalent. For every equivalence class we can choose a point lying on the unit circle as a representative. The projective line  $\mathbb{P}^1(\mathbb{R})$  is then represented by the unit circle with diagonally opposite points identified together.

What can we say about the relationship between the affine (ordinary) plane and the projective plane? If a projective point has nonzero  $Z$ , then there is one and only one triple in its equivalence class of the form  $(x, y, 1)$ ; simply set

$$x = X/Z, y = Y/Z. \quad (5)$$

On the other hand if we have a point in affine space  $(x, y)$  we can embed this point in projective plane by setting the  $Z$  coordinate to one;  $(x, y) \in k^2 \mapsto (x, y, 1) \in \mathbb{P}^2(k^2)$ . Thus the projective plane can be identified with all points  $(x, y)$  of the affine plane plus the points for which  $Z = 0$ . The latter points make up what is called the *line at infinity*; roughly speaking; and can be visualized as the "horizon" on the plane.

To get the "projective equation" of our elliptic curve  $Y^2 = X^3 + aX + b$  we apply transformation (5) on the equation and multiply by a power of  $Z$  to clear the denominators :

$$Y^2Z = X^3 + aXZ^2 + bZ^3. \quad (6)$$

This equation is satisfied by all projective points with  $Z \neq 0$  for which the corresponding affine points satisfy the affine equation (3). Which points on the line at infinity satisfy equation (6)? Setting  $Z = 0$  in the equation leads to  $0 = X^3$ , i.e.  $X = 0$ . The only equivalence class with both  $X$  and  $Z$  zero is the class  $(0, 1, 0)$ . This is the point we called  $\mathcal{O}$  in the last section. It is the point on the intersection of the  $y$ -axis with the line at infinity.

What other advantages do we have when calculating in projective space? First of all, we have arithmetic advantages. When we look at the formulas for point addition and point doubling in projective coordinates, we see that we don't have to calculate the inverse of a coordinate anymore. As this is an "expensive" operation, this is a major advantage in practical applications. Figure 2 shows the formulae for projective point addition in  $\mathbb{F}_p$ . The formulas given here use the so called *weighted projective coordinates* which are more efficient than homogeneous projective coordinates. Actually there exist many forms of "projective coordinate representations" having each positive and negative aspects for practical issues.

$P_3 = P_1 + P_2$	
$\mathbb{F}_p$	$\mathbb{F}_{2^m}$
$\lambda_1 = X_1 Z_2^2$	$\lambda_1 = X_1 Z_2^2$
$\lambda_2 = X_2 Z_1^2$	$\lambda_2 = X_2 Z_1^2$
$\lambda_3 = \lambda_1 - \lambda_2$	$\lambda_3 = \lambda_1 + \lambda_2$
$\lambda_4 = Y_1 Z_2^3$	$\lambda_4 = Y_1 Z_2^3$
$\lambda_5 = Y_2 Z_1^3$	$\lambda_5 = Y_2 Z_1^3$
$\lambda_6 = \lambda_4 - \lambda_5$	$\lambda_6 = \lambda_4 + \lambda_5$
$\lambda_7 = \lambda_1 + \lambda_2$	$\lambda_7 = Z_1 \lambda_3$
$\lambda_8 = \lambda_4 + \lambda_5$	$\lambda_8 = \lambda_6 X_2 + \lambda_7 Y_2$
$Z_3 = Z_1 Z_2 \lambda_3$	$Z_3 = Z_2 \lambda_7$
$X_3 = \lambda_6^2 - \lambda_7 \lambda_3^2$	$\lambda_9 = \lambda_6 + Z_3$
$\lambda_9 = \lambda_7 \lambda_3^2 - 2X_3$	$X_3 = aZ_3^2 + \lambda_6 \lambda_9 + \lambda_3^2$
$Y_3 = (\lambda_9 \lambda_6 - \lambda_8 \lambda_3^3)/2$	$Y_3 = \lambda_9 X_3 + \lambda_8 \lambda_7^2$

Table 2: Weighted projective point addition

The condition  $P_1 = \pm P_2$  is equivalent to  $\lambda_3 = 0$  in figure 2. Additionally, if  $\lambda_6 = 0$ , i.e. if  $P_1 = P_2$ , the following point doubling routine will be used.

$P_3 = 2P_1$	
$\mathbb{F}_p$	$\mathbb{F}_{2^m}$
$\lambda_1 = 3X_1 + aZ_1^4$	$Z_3 = X_1Z_1^2$
$Z_3 = 2Y_1Z_1$	$X_3 = (X_1 + bZ_1^2)^4$
$\lambda_2 = 4X_1Y_1^2$	$\lambda = Z_3 + X_1^2 + Y_1Z_1$
$X_3 = \lambda_1^2 - 2\lambda_2$	$Y_3 = X_1^4Z_3 + \lambda X_3$

Table 3: Weighted projective point doubling

The following tables list the cost of point addition and point doubling for both, affine and projective coordinates (These formulas correspond not exactly to the formulas for point arithmetic given before! Take a look at the P1363a standard for the arithmetic). Thereby  $I$  denotes the inversion and  $M$  denotes the multiplication. Projective coordinates stands always short for weighted projective coordinates.

Operation	Coordinates	
	affine	projective
Addition	1I+3M	16M
Doubling (arbitrary $a$ )	1I+4M	10M
Doubling ( $a = -3$ )	1I+4M	8M

Table 4: Cost of point addition ( $\mathbb{F}_p$ )

In the next table, we denote by  $S$  the subtraction. The cost of field additions, as well as multiplications by small constants (e. g. 2 and 3 in the computation of  $\lambda$ ) are neglected in the tables.

Operation	Coordinates	
	affine	projective
Addition ( $a \neq 0$ )	1I+2M+1S	15M+5S
Addition ( $a = 0$ )	1I+2M+1S	14M +4S
Doubling	1I+2M+1S	5M+5S

Table 5: Cost of point addition ( $\mathbb{F}_{2^m}$ )

The next remark is too brief to be understandable and for readers with a special interest in maths only : There is an easy method for transporting the group law if it is only defined in  $\mathbb{Q}$  to  $\mathbb{F}_p$ . One can define a "reduction" which has the nice property that it respects the group law. By using the projective equation we can define a reduction modulo  $p$  which respects the group law. Therefore one chooses a "reduced" point  $\bar{P}$  in  $\mathbb{P}^2(\mathbb{Q})$  (reduced means that  $\text{ggt}(X, Y, Z) = 1$ ). This point can then be reduced modulo  $p$  :  $r_p(P) := (X \pmod{p}, Y \pmod{p}, Z \pmod{p})$ .  $r_p$  is in  $\mathbb{P}^2(\mathbb{F}_p)$  because one of  $X, Y, Z$  is not zero. The reduction of  $E$  is defined as  $r_p(E) \rightarrow E_p : Y^2 = X^3 + aX + b \pmod{p}$ . One can show that for  $P_1, P_2 \in E(\mathbb{Q})$ , and  $P_1 + P_2 = Q$  then also  $\bar{P}_1 + \bar{P}_2 = \bar{Q}$ .

## 4 Properties of Cryptographic Interest

In the last part of section (2) we discuss some topics which are of interest for cryptographers. We already defined an algebraic structure called "elliptic curve" and found a way to make the set of points on such a curve to an abelian group. Basically we could start building a cryptosystem, but there are some more questions that should be considered in advance:

- How "big" is an elliptic curve?
- How do I find a point on an elliptic curve?
- How do I find a curve?

In the rest of this section we will mostly deal with the finite field  $\mathbb{F}_q$ , where  $q = p^r$ .

### 4.1 Calculating the number of points on an elliptic curve over $\mathbb{F}_q$

With the following easy counting method it is possible to give an lower and an upper bound for the number of  $\mathbb{F}_q$  points. Therefore we choose an  $x \in \mathbb{F}_q$  and assert if there is a corresponding  $y$  on the curve, i.e. we look if  $f(x) = x^3 + ax + b$  is a square in  $\mathbb{F}_q$ . We restrict our considerations to the case  $q = p$ . Then we can introduce the (easier)<sup>2</sup> notation which is used to work with squares, or how mathematicians call them "quadratic residues" (short QR).

**Definition 6 (The Legendre symbol).** Let  $a$  be an integer and  $p > 2$  be a prime. The Legendre symbol  $\left(\frac{a}{p}\right)$  is defined as follows

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p|a; \\ 1, & \text{if } a \text{ is quadratic residue modulo } p; \\ -1, & \text{if } a \text{ is not a quadratic residue modulo } p. \end{cases}$$

The Legendre symbol tells us whether or not an integer is a quadratic residue modulo  $p$ . A simple method to compute the value of the Legendre symbol is given in the next proposition.

**Proposition 2.**

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

*Proof.* For the trivial case where  $a = 0$  both sides are  $\equiv 0 \pmod{p}$ . Suppose  $a > 0$  and  $p \nmid a$ . Fermat's little theorem states that the square of  $a^{(p-1)/2}$  is 1, so  $a^{(p-1)/2}$  itself is  $\pm 1$ . Let  $g$  be a generator of  $F_p^*$ , and let  $a = g^j$ .  $a$  is a quadratic residue if and only if  $j$  is even. And  $a^{(p-1)/2} = g^{j(p-1)/2}$  is 1 if and only if  $j(p-1)/2$  is divisible by  $(p-1)$ , i. e. if and only if  $j$  is even. Thus both sides are  $\pm 1$  in  $F_p$ , and each side is  $+1$  if and only if  $a$  is a square.  $\square$

Depending whether  $f(x)$  is a quadratic residue or not modulo  $q$ , we can have the following cases

- **f(x) is QR:** Then there are two points  $(x, \pm y)$ .
- **f(x) divides p:** Then there is a single point  $(x, 0)$ .

<sup>2</sup>For the more general case, i. e.  $q = p^r$ , one has to take the quadratic character of  $\mathbb{F}_q$ .

- **f(x) in not QR:** Then there is no point.

Putting all three cases into one formula results in :

$$\#E(\mathbb{F}_q) = 1 + \sum_{x \in \mathbb{F}_q} \left( 1 + \left( \frac{f(x)}{q} \right) \right) = q + 1 + \sum_{x \in \mathbb{F}_q} \left( \frac{f(x)}{q} \right).$$

$f(x)$  denotes the right side of equation (3) resp. (4). The upper and lower bound for  $\#E(\mathbb{F}_q)$  is therefore  $1 \leq \#E(\mathbb{F}_q) \leq 2q + 1$ . For an exact calculation of the number of points, the Legendre symbol must be evaluated for every  $x \in \mathbb{F}_q$ . This is inefficient for a big field. For this purpose one uses Schoof's algorithm which is beyond the scope of this introduction. We continue instead with our consideration of upper and lower boundaries, and expand that considerations to (arbitrary) extension fields. If an elliptic curve is defined over  $\mathbb{F}_q$  then it is also defined over  $\mathbb{F}_{q^r}$ . So it is meaningful to look at solutions in extension fields of the defining equation of the curve (and of course it is even more interesting for  $q = 2$ ! Let  $N_r$  denote the number of  $\mathbb{F}_{q^r}$  points on  $E$  (so  $N_1 = N = \#E(\mathbb{F}_q)$ ). One forms a generating series  $Z(E/\mathbb{F}_q; T)$ , which is the formal power series defined by setting

$$Z(E/\mathbb{F}_q; T) = e^{\sum N_r T^r / r}, \quad (7)$$

in which  $T$  is an indeterminate. The sum is over all  $r \in \mathbb{N}$ . This series is also called *zeta-function* of the elliptic curve (over  $\mathbb{F}_q$ ). Although the definition looks rather difficult, the following theorem shows, that the zeta-function has a really simple form.

**Theorem 1.** *The zeta-function of an elliptic curve is a rational function of  $T$  having the form*

$$Z(E/\mathbb{F}_q; T) = \frac{1 - aT + qT^2}{(1 - T)(1 - qT)}, \quad (8)$$

where only the coefficients of  $T$  in the numerator depends on the particular elliptic curve  $E$ . This coefficient is related to  $N = N_1$  as follows:  $N = q + 1 - a$ . In addition, the discriminant of the quadratic polynomial in the numerator is negative or zero (that is,  $a^2 \leq 4q$ ) and so this polynomial has two complex conjugate roots  $\alpha, \bar{\alpha}$  both of absolute value  $\sqrt{q}$ . (More precisely,  $1/\alpha$  and  $1/\bar{\alpha}$  are the roots, and  $\alpha, \bar{\alpha}$  are the "reciprocal roots".)

The proof of this theorem is also beyond the scope of this tutorial. For a proof see Silverman ([12]). An immediate consequence of this theorem is the next proposition. We just have to write the numerator on the right side of equation (8) in the form  $(1 - \alpha T)(1 - \bar{\alpha} T)$ . Then we replace the left side by the right side of equation (7), and take the logarithm of both sides. We use then the identity  $\ln(1 - cT) = -\sum c^r T^r / r$ .

**Corollar 1.** *Let  $N_r$  denote the number of  $\mathbb{F}_{q^r}$  points on  $E$ , and set  $N = N_1$  and  $a = q + 1 - N$ . Let  $\alpha$  and  $\bar{\alpha}$  be the roots of the quadratic polynomial  $T^2 - aT + q$ . Then*

$$N_r = |\alpha^r - 1|^2 = q^r + 1 - \alpha^r - \bar{\alpha}^r, \quad (9)$$

where  $|\cdot|$  denotes the complex absolute value.

This corollary shows how to determine all of the  $N_r$  once you know  $N = N_1$ .

**Example 1.** We calculate the zeta-function of the elliptic curve  $y^2 + y = x^3$  over  $\mathbb{F}_2$ . Since there are exactly three  $\mathbb{F}_2$ -points ( $P_1 = (0, 0)$ ,  $P_2 = (0, 1)$ ,  $P_3 = \mathcal{O}$ ) on this curve, we obtain the value of  $a$  which is 0 (using the relationship  $N = q + 1 + a$  given in theorem 1). The zeta-function can then be written as

$$Z(E/\mathbb{F}_2; T) = \frac{1 + 2T^2}{(1 - T)(1 - 2T)}. \quad (10)$$

Using corollary 1 with  $\alpha = i\sqrt{2}$  and  $\bar{\alpha} = -i\sqrt{2}$  we get  $N_r = 2^r + 1 - (i\sqrt{2})^r - (-i\sqrt{2})^r$ . This leads to the formula

$$N_r = \begin{cases} 2^r + 1, & \text{if } r \text{ is odd;} \\ 2^r + 1 - 2(-2)^{r/2}, & \text{if } r \text{ is even.} \end{cases} \quad (11)$$

Another corollary, which is sometimes called "Hasse's theorem" is the following fact.

**Corollar 2.** The number  $N$  of  $\mathbb{F}_q$ -points on an elliptic curve defined over  $\mathbb{F}_q$  lies in the interval

$$q + 1 - 2\sqrt{q} \leq N \leq q + 1 + 2\sqrt{q}. \quad (12)$$

This follows because  $|q + 1 - N| = |a| \leq \sqrt{4q}$ .

Before we proceed we give another definiton.

**Definition 7.** The order of an elliptic curve is defined as the number of points on the curve.

## 4.2 Obtaining a point on an elliptic curve over $\mathbb{F}_q$

One open questions is, how to find a point on an elliptic curve, and especially, how to find "generator" points, i.e. points which scalar multiples give all other elliptic curve points. We consider only the case for  $q = p \in \mathbb{P}$ . Otherwise one should consult one of the various textbooks.

Suppose we'd like to find an arbitray point. The straightforward method would be choosing a  $x$ -coordinate and look if there's a  $y$  satisfying the elliptic curve equation (i.e. evaluate the Legendre symbol for  $f(x)$ ). If there is no  $y$  then we choose another  $x$ . Now lets assume that we have found an  $x$  for which we can be sure that a  $y$  exists. We have to find that  $y$ , which means that we have to find the square root of a given number  $y^2 \equiv z \pmod{q}$ . If  $q \equiv 3 \pmod{4}$  it is very simple. One just has to choose  $y \equiv z^{(q+1)/4}$  because with Fermat's little theorem one has  $y^2 = z^{(q+1)/2} = zz^{(q-1)/2} \equiv z \pmod{q}$ .

The more interesting case is  $q \equiv 1 \pmod{4}$ . The most widely used method is the *approximation method* due to *Shanks*. We first write  $q$  as  $q - 1 = 2^s t$ ,  $t$  odd,  $s \leq 2$ . We choose a non-square element  $u$  in  $\mathbb{F}_q^*$  (i.e.  $u^{(q-1)/2} = -1$ ). Then we set  $v = u^t$ , then  $v$  is a primitive  $2^s$ -th root of unity in  $F_q$  (it generates the subgroup  $G$  with  $2^s$  elements). The last sentence needs to be proved.

**Proposition 3.** With the above setup of variables,  $v = u^t$  is a primitive  $2^s$ -th root of unity.

*Proof.* We first show that  $v = u^t$  is a  $2^s$ -th root of unity. This is equivalent to  $2^s t \equiv 0 \pmod{q-1}$  and is true due to our setup. Now we proof that it is primitive too. Assume the opposite, i. e. let  $v$  be not primitive. Then there has to be a lower power of  $v$  (a divisor of  $2^s$ ) that is congruent 1. But then  $v$  would be an even power of a  $2^s - th$  root of unity and therefore

a square in  $\mathbb{F}_q^*$ . This leads to the equation  $1 = \left(\frac{v}{p}\right) = \left(\frac{u}{p}\right)^t = -1$  (because  $v$  is a quadratic residue so the evaluation of the legendre symbol must give 1, but as  $u$  is chosen to be a quadratic non-residue and  $t$  is odd the evaluation of the right side gives -1).  $\square$

We get an "approximate" solution  $y_1 = z^{(t+1)/2}$ . This is "close" to the exact solution  $z$  because if we look at  $y_1^2 z^{-1}$  then we get the equation

$$(y_1^2 z^{-1})^{2^{s-1}} = (z^{t+1} z^{-1})^{2^{s-1}} = z^{t2^{s-1}} = z^{t2^s/2} = z^{(q-1)/2} = \left(\frac{z}{q}\right) = 1.$$

So if we have a "correction" term (a  $2^s$ -th root of unity  $v^{-l}$ ) we can convert  $y_1$  to a square root of  $z$ . Let's write  $l$ 's binary digits as  $l = l_0 + l_1 2 + l_2 2^2 + \dots + l_{s-2} 2^{s-2}$ . The naive method of finding  $l$  would be to try all possible values until the correct one is found. That's pretty boring, so we proceed with Shank's method. We determine the binary digits inductively starting with  $l_0$ .

- **Step 1.** Raising both sides of  $z = y^2 = y_1^2 v^{-2l} = z^t$  to the  $2^{s-2} - th$  power we see that  $v^{2l_0 2^{s-2}} = \left(\frac{y_1^2}{z}\right)^{2^{s-2}} = \pm 1$  and  $l_0 = 0$  if and only if we obtain 1 on the right side, otherwise  $l_0 = 1$  (because if  $l_0 = 1$  then we know from the long equation above that the right side could be both  $\pm 1$ ). In other words  $l_0$  is chosen in such a way that  $(v^{l_0} y_1)^2 / z$  is a  $2^{s-2}$ -th root of unity.
- **Step  $k$ .** Assume that we are in step  $k$ , so that we already have the digits  $l_0, \dots, l_{k-1}$  with the property that  $(v^{l_0 + l_1 2 + \dots + l_{k-1} 2^{k-1}} y_1)^2 / z$  is a  $2^{s-k-1}$ -th root of unity. So we take the  $2^{s-k-2}$  power of this expression which is  $\pm 1$ . Again we set  $l_k = 0$  if it is 1 and  $l_k = 1$  if it is  $-1$ . So  $(v^{l_0 + l_1 2 + \dots + l_k 2^k} y_1)^2 / z$  is a  $2^{s-k-2}$ -th root of unity.
- **Last step.** If we are in step  $k = s - 2$  we will calculate  $l_{s-2}$  and we will get the square root of  $z$  because  $v^{l_0 + \dots + 2^{s-2} l_{s-2}} y_1)^2 / z = 1$ .

This probabilistic algorithm takes time  $O(\ln^3 q)$ . Don't ask for a deterministic polynomial-time algorithm for finding points (other than the point at infinity) on an elliptic curve since no such algorithm is known in general. The main obstacle to a deterministic polynomial-time algorithm is not the problem of taking the square root. Rather, it is finding  $x \in \mathbb{F}_q$  such that  $f(x)$  is a square. Although about the half of the elements have this property, no efficient deterministic way is known to find such an element except some special cases.

Now that we know how to determine an arbitrary point, one can ask for generator points, resp. points with large order (which is sufficient for our purposes), too. The *order*  $n$  of a point is the smallest positive integer such that  $nP = \mathcal{O}$  (of course such a finite  $n$  need not to exist for elliptic curves defined over an arbitrary field). If  $k$  and  $l$  are integers, then  $kP = lQ$  if and only if  $k \equiv l \pmod{n}$ . For a finite field this order always exists and divides the order of the curve. This leads us to the following problem: Given a finite field, find an elliptic curve defined over that field whose order is divisible by a sufficiently large prime  $r$ .

### 4.3 Constructing an elliptic curve over a given finite field

We motivate this section with the following example.

**Example 2.** Let  $E$  be the elliptic curve given through the equation  $y^2 = x^3 + 3x + 1$  over  $\mathbb{F}_p$ ,  $p = 10^7 + 19$ . The order of the curve is  $n = \#E(\mathbb{F}_p) = 9999846 = 2 \cdot 3^2 \cdot 347 \cdot 1601 = 18 \cdot 347 \cdot 1601$ . We take the curve points  $P = (2, 4417259)$  and  $Q = (1, 866032)$  with  $Q = xP$ ,  $x$  unknown. If we want to determine  $x$  we have to solve the elliptic curve discrete logarithm problem which is the analog of the discrete logarithm problem and will be defined more formally in the next section. This problem is the underlying problem of all cryptosystems based on elliptic curves and has to be hard! But know look at this: Since we know the factorization of  $n$  we get the following set of equations.

$$x \frac{n}{18} P = \frac{n}{18} Q \quad \text{gives } x \equiv 14 \quad (18)$$

$$x \frac{n}{347} P = \frac{n}{347} Q \quad \text{gives } x \equiv 81 \quad (347)$$

$$x \frac{n}{1601} P = \frac{n}{1601} Q \quad \text{gives } x \equiv 854 \quad (1601)$$

We know for example that  $18 \cdot \frac{n}{18} P = nP = \mathcal{O}$  so we can reduce the space for the possible values of  $x$  significantly (we just have to look between zero and 17). Now we can easily check every  $x$  and get the relation  $x \equiv 14 \pmod{18}$ . According to this method we can obtain all of the left equations. Then just have to apply the chinese remainder theorem to get  $x = 5553122$ .

The method sketched in the example is called Silver-Pohlig-Hellman method. It works so well in the example because  $n$  has small divisors (one says  $E$  has "smooth" order). It is clear that for cryptographic purposes one should avoid those curves!

This makes our effort in obtaining curves even more difficult. We have to find a curve (over a given finite field) which does not have a smooth order and additionally has a base point with large (prime) order. This can be accomplished by the following four approaches:

- Select a curve equation at random, compute its order directly, and repeat this process until an appropriate order is found.
- Select curve coefficients with particular (desired) properties, compute the curve order directly, and repeat the process until an appropriate order is found.
- If  $q = 2^m$  where  $m$  is divisible by a "small" integer  $d$ , then select a curve defined over  $\mathbb{F}_{2^d}$  and compute its order over  $\mathbb{F}_{2^m}$ . Repeat if possible until an appropriate order is found.
- Search for an appropriate order, and construct a curve of that order.

We already realized in the last section that counting points is not as easy as it seems, but we fairly got an idea how to do it. The last point is the only idea we didn't discuss. But unfortunately, this method (which is called *complex multiplication* method) is also far beyond the scope of this tutorial.

Before we continue in the next section with cryptography, there is a list of terms that are worth to be explored (but not in this tutorial).

What is :

- the  $j$ -invariant of an elliptic curve,
- a torsion point,

- the structure of the group of points,
- complex multiplication,
- the relationship between elliptic curves and Fermat's Last Theorem?

By studying all of the cited books, some of these questions could be answered.

## 5 Elliptic Curve Cryptosystems

All readers familiar with public key cryptography know the definition of the discrete logarithm problem in the multiplicative group of a finite field. We can give an analog definition for the group of points on an elliptic curve.

**Definition 8.** *If  $E$  is an elliptic curve over  $\mathbb{F}_q$  and  $B$  is a point of  $E$ , then the discrete log problem on  $E$  (to the base  $B$ ) is the problem, given a point  $P \in E$ , of finding an integer  $x \in \mathbb{Z}$  such that  $xB = P$  if such an integer  $x$  exists.*

What can we say about the hardness of this problem? Until 1990, the only discrete log algorithms known for an elliptic curve cryptosystem were the ones that work in any group. These are exponential time algorithms, provided that the order of the group is divisible by a large prime factor. Menezes, Okamoto and Vanstone found a new approach to the discrete log problem on an elliptic curve. They used the Weil pairing to embed the group of points on  $E$  into the multiplicative group of some extension field  $\mathbb{F}_{q^k}$ . It is essential for the extension degree  $k$  to be small. The only elliptic curves for which  $k$  is small are the so-called "supersingular" curves which we omitted since the beginning of this tutorial.

### 5.1 Embedding plaintext on an elliptic curve

Suppose we'd like to encrypt some plaintext with ECC. There has to be a method, which takes some arbitrary text and embeds it in an elliptic curves, i.e. which gives a bijection between the points on an elliptic curve and a plaintext block. We sketch such an algorithm.

**1. Step:** We choose an alphabet with  $N$  letters and fix the length  $l$  of a plaintext block. The characters of the alphabet are then identified with the numbers  $0, \dots, N-1$ . With the following assignment we get a bijection between the plaintext blocks  $w$  and the numbers  $0 \leq x_w \leq N^l$ :

$$w = (a_0 a_1 \dots a_{l-1}) \mapsto x_w = a_0 N^{l-1} + a_1 N^{l-2} + \dots + a_{l-2} N + a_{l-1}, \quad 0 \leq x_w \leq N^l$$

**Idea:** For such an  $x_w$  there need not be a point on the elliptic curve. But it should be possible to find the "next" curve point  $x_1$  close to  $x_w$  efficiently. Given a number  $k$  we'd like to have a high probability (i.e.  $1 - (1/2)^k$ ) for  $x_w \leq x_1 < x_w + k$ .

**2. Step:** We choose an appropriate  $k$ , i.e. that the success probability is high and that  $q > kN^l$ . For each  $j$  we obtain an element of  $\mathbb{F}_q$  through  $kx_w + j$ . We take the first curve point ( $j \geq 0$ )  $P_w$  with x-coordinate  $\geq kx_w$ , i.e.  $P_w = (kx_w + j, *) \in E(\mathbb{F}_q)$ .

**3. Step:** We can recover the plain text block from the point by  $x_w = \lfloor \frac{x}{k} \rfloor$ .

## 5.2 Elliptic Curve Diffie-Hellman key exchange (ECDH)

Suppose two communication parties, Alice and Bob, want to agree upon a key which will be later used for encrypted communication in conjunction with a private key cryptosystem. They first fix a finite field  $\mathbb{F}_q$ , an elliptic curve  $E$  defined over it and a base point  $B \in E$  (with high order). To generate a key, first Alice chooses a random  $a \in \mathbb{F}_q$  (of high order) which she keeps secret. Next she calculates  $aB \in E$  which is public and sends it to Bob. Bob does the same steps, i.e. he chooses a random integer  $b$  (secret) and calculates  $bB$  which is sent to Alice. Their secret common key is then  $P = abB \in E$ .

## 5.3 Analog of El-Gamal

We start with a fixed publicly known finite field  $\mathbb{F}_q$ , an elliptic curve  $E$  defined over it and a base point  $B \in E$ . Each user chooses a random integer  $a$ , which is kept secret, and computes the point  $x = aB$  which is the public key. To send a message  $P$  to Bob, Alice chooses a random integer  $k$  and sends the pair of points  $(kB, P + k(bB))$  (where  $bB$  is Bob's public key) to Bob. To read the message, Bob multiplies the first point in the pair by his secret  $b$  and subtracts the result from the second point:  $P + k(bB) - b(kB) = P$ .

## 5.4 Elliptic Curve Digital Signature Algorithm (ECDSA)

The ECDSA is the elliptic curve analog of the DSA. ECDSA was first proposed in 1992 by Vanstone in response to NIST's (National Institute of Standards and Technology) request for comments on their first proposal for DSS. Digital signature schemes are the counterpart to handwritten signatures. A digital signature is a number that depends on the secret key only known by the signer and on the contents of the message being signed. Signatures must be verifiable without access to the signer's private key. Signatures should be existentially unforgeable under chosen-message attacks. This asserts that an adversary who is able to obtain Alice's signatures for any messages of his choice cannot forge Alice signature on a single other message.

Suppose Alice wants to send a digitally signed message to Bob. They first choose a finite field  $\mathbb{F}_q$ , an elliptic curve  $E$ , defined over that field and a base point  $G$  with order  $n$ . Alice's key pair is  $(d, Q)$ , where  $d$  is her private and  $Q$  is her public key. To sign a message  $M$  Alice does the following:

- |    |   |
|----|---|
| 1. | Choose a random number $k$ with $k : 1 \leq k \leq n - 1$ .                     |
| 2. | Compute $kG = (x_1, y_1)$ and $r = x_1 \bmod n$ . If $r = 0$ then go to step 1. |
| 3. | Compute $k^{-1} \bmod n$ .  |
| 4. | Compute $e = \text{SHA-1}(M)$ .   |
| 5. | Compute $s = k^{-1}(e + dr) \bmod n$ . If $s = 0$ then go to step 1.            |
| 6. | Alice signature for the message $M$ is $(r, s)$ .                               |

Figure 3: ECDSA Signatur Generierung

To verify Alice's signature  $(r, s)$  on the message  $m$ , Bob obtains an authentic copy of Alice's parameters and public key. Bob should validate the obtained parameters! Bob then does the following: If the signature  $(r, s)$  on the message  $m$  was indeed generated by Alice,

1.	Verify that $r, s$ are integers in the interval $[1, n - 1]$ .
2.	Compute $e = \text{SHA-1}(M)$ .
3.	Compute $w = s^{-1} \pmod n$ .
4.	Compute $u_1 = ew \pmod n$ and $u_2 = rw \pmod n$ .
5.	Compute $X = u_1G + u_2Q$ . If $X = \mathcal{O}$ then reject the signature. Otherwise compute $v = x_1 \pmod n$ where $X = (x_1, y_1)$ .
6.	Accept the signature if and only if $v = r$ .

Figure 4: ECDSA Signatur Verifikation

the  $s = k^{-1}(e + dr) \pmod n$ . With this information we have

$$k \equiv s^{-1}(e + dr) \equiv s^{-1}e + s^{-1}rd \equiv we + wrd \equiv u_1 + u_2d \pmod n.$$

Thus  $u_1G + u_2Q = (u_1 + u_2d)G = kG$  and so  $v = r$  as required.

For a discussion on known attacks and how they can be avoided consult [4].

### 5.5 Elliptic Curve Primality Proving (ECP)- Elliptic Curve Factoring Method (ECM)

The ECM method is the elliptic curve analogon to the (p-1)-method. One uses an equation  $y^2 = x^3 + ax + b$  over  $\mathbb{F}_n$ . The ECM method determines prime divisors for which the order of  $E(\mathbb{F}_n)$  is smooth. The advantage of this method is basically that there are lots of elliptic curves over a given finite field. If we can't succeed with one elliptic curve equation we can take another one; modern implementations work with several hundreds of elliptic curves in parallel.

Let's look at the idea in detail. Suppose we have a number  $n$ . We don't know if this number is prime or not. No matter if it is prime or not we can apply the formulas of section (2) to add elements on  $E$ . One of three things can happen when we add two points:

1. we get a valid point on the curve,
2. if we add two points of the form  $(x, y)$  and  $(x, -y)$  we get  $\mathcal{O}$ ,
3. the formulas are undefined, because we have a denominator which is not invertible modulo  $n$ .

Case three implies that  $\gcd(\text{denominator}, n) > 1$  which means firstly that  $n$  is not prime and secondly that we have found a nontrivial divisor of  $n$ . On the other hand, if  $n$  is definitely prime then this case will never happen. We can formulate this more precisely in the following proposition.

**Proposition 4.** *Let  $n$  be a positive integer. Let  $E$  be an elliptic curve modulo  $n$ . Let  $m$  be an integer. Suppose that there is a prime  $q$  which divides  $m$  which is greater than  $(n^{1/4} + 1)^2$ . If there exists a point  $P$  of  $E$  such that (i)  $mP = \mathcal{O}$ ; and (ii)  $(m/q)P$  is defined and not equal to  $\mathcal{O}$ , then  $n$  is prime.*

*Proof.* If  $n$  is not prime, than there exists a prime  $p \leq \sqrt{n}$  which divides  $n$ . Let  $E'$  be the elliptic curve given by the same equation as  $E$  but considered modulo  $p$ , and let  $m'$

be the order of the group of points of  $E'$ . By the inequality 2 (Hasse's theorem) we have  $m' \leq p + 1 + 2\sqrt{p} = (\sqrt{p} + 1)^2 \leq (n^{1/4} + 1)^2 < q$ , and hence  $\gcd(m', q) = 1$ . Since there exists an integer  $u$  such that  $uq \equiv 1 \pmod{m'}$ . Let  $P' \in E'$  be the point  $P$  considered modulo  $p$ . Then in  $E'$  we have  $(m/q)P' = uq(m/p)P' = umP' = \mathcal{O}$  by (i) since  $mP'$  is obtained using the same procedure as  $mP$ , only working modulo  $p$ . This contradicts (ii) since if  $(m/q)P$  is defined and  $\neq \mathcal{O}$  modulo  $n$ , the same procedure working modulo  $p$  will give  $(m/q)P' \neq \mathcal{O}$ .  $\square$

This proposition leads to following primality proving algorithm.

**Step 1.** Randomly select three integers  $a, P = (x, y)$  and set  $b \equiv y^2 - x^3 - ax \pmod{n}$ .

**Step 2.** Count the number of points  $m$  on the elliptic curve.

**Step 3.** If  $m$  cannot be written in the form  $m = kq$  then go back to step 1.

**Step 4.** Compute  $kP$  and  $mP$ .

**Step 5.** If we ever obtained an undefined expression (either at the point counting or here) we have a nontrivial factor of  $n$

**Step 6.** If  $kP = \mathcal{O}$  we are out of luck and have to back to step 1.

**Step 7.** If  $kP \neq \mathcal{O}$  and  $mP = \mathcal{O}$  we know that  $m$  is prime provided  $q$  is really a prime. This reduces the problem to proving primality of  $q$ , which has magnitude at most  $n/2$ .

## 6 State of the art in ECC

In the last section we will investigate the ongoing efforts in the standardization of elliptic curve cryptography. The development of standards is a very important point for the use of a cryptosystem. Standards help ensure security and interoperability of different implementations of one cryptosystem. There are several major organisations that develop standards. The most important for security in information technology are the

- International Standards Organization (ISO),
- American National Standards Institute (ANSI),
- Institute of Electrical and Electronics Engineers (IEEE),
- Federal Information Processing Standards (FIPS).

The most prominent ECC algorithm, the ECDSA was accepted 1998 as ISO standard (ISO 14888-3), 1999 as ANSI standard (ANSI X9.62), and 2000 as IEEE (P1363) and Fips (186-2) standard. Several other standardisation efforts are in progress. The most prominent is the ANSI X9.63 which includes dozens of key transport and key agreement schemes. Table 6 provides an overview of the various standards and the included algorithms. The relationship between these standards is visualized in figure 5.

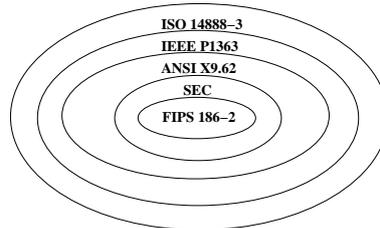


Figure 5: Standards

There is a company with a certain interest in elliptic curve cryptography which enforces the integration of ECC in several commonly used protocols. We will discuss some of the briefly. For example in the recent version of the TLS/SSL (Transport Layer Security/Secure Socket Layer) protocol basically ECC is supported. In its recent implementation of this protocol, Certicom has already included this feature.

Standard	Schemes	Status
ANSI X9.62	ECDSA	approved
ANSI X9.63	ECIES, ECDH, ECMQV	draft
FIPS 186-2	ECDSA	approved
IEEE P1363	ECDSA, ECDH, ECMQV	approved
IEEE P1363A	ECIES	draft
ISO 14888-3	ECDSA	approved
ISO 15946	ECDSA, ECDH, ECMQV	draft

Table 6: Standards and Algorithms

Another important application is the Wireless Application Protocol (WAP/WTLS). It ensures secure wireless communications and includes in its recent version ECC in the WTLS layer. Also in the ATM Security Specification 1.0 ECC is included. They used modified variants of the ANSI algorithms in order to avoid the costly inversion in the signature generation. Table 7 compares the two algorithms.

ECDSA-like	ECDSA
$r = x_1 \bmod n$	$r = x_1 \bmod n$
$s = (kr - e)d^{-1}$	$s = k^{-1}(e + dr)$

Table 7: ECDSA vs. ECDSA-like

The Secure/Multipurpose Internet Mail Extensions (S/MIME) standard specifies security mechanisms for electronic communications. S/MIME can be applied to all compatible MIME formats, such as http for example. The internet Draft Elliptic Curve S/MIME tries to embed ECC in S/MIME.

## 6.1 X.509

The most interesting application for ECC is certainly the application of ECDSA in digital signatures. X.509 is the most widely used certificate standard and is part of lots of applications like SSL or S/MIME. Figure 6 shows the basic structure of an X.509 certificate.

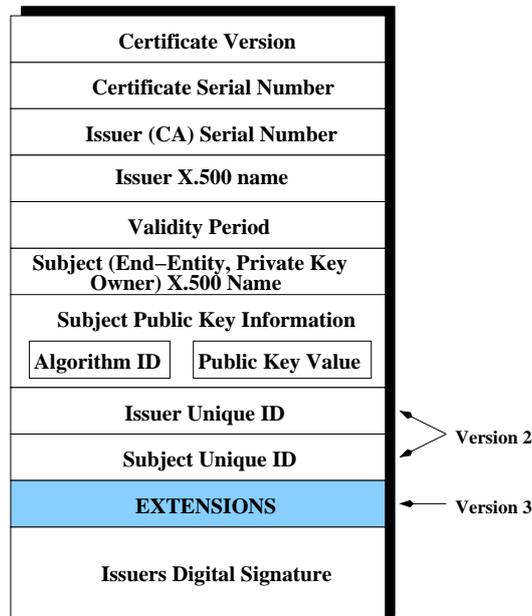


Figure 6: X.509 certificate

The certificate data is written in the Abstract Syntax Notation 1 (ASN.1) syntax and is DER (Distinguished Encoding Rules) encoded. X.509 does not directly address any algorithms, but instead uses "object identifiers". Profiles, such as PKIX or X9.55, specify the usage of algorithms. In order to support ECC one must define an ASN.1 Syntax for the algorithms and profile their use. This is basically done in ANSI X9.62. All other standards use this syntax notation. Here are some remarks towards this definitions:

- One can use the same ECC keys for both digital signing and encryption.
- Elliptic curve parameters can be given directly in the certificate or can be referenced by name.
- The key usage section for ECDSA includes : digitalSignature, nonRepudiation, keyCertSign and cRLSign.
- The key usage section for ECDH includes : keyAgreement, encipherOnly, decipherOnly (encipherOnly and decipherOnly are "exclusive or").

## 6.2 Available Software- and Hardwareimplementations

The most prominent chip manufacturers that have ECC implementations are Siemens (the well-known "Pluto-IC"), Infineon (SLE66xxP), Motorola (MPC180x) and Philips (SmartXA).

Most of these implementations use elliptic curves over  $\mathbb{F}_p$  because of the many (pending) patents from Certicom. Certicom also has the broadest palette of ECC software including their *Security Builder* crypto-library. But also their "counterpart" RSA company has included ECC in the recent version of their *Bsafe* crypto-library. Other well known companies such as Cryptomathic, Secude and Entrust have their own ECC implementations. Lots of other, also well known companies, hold a licence from Certicom. One should consult Certicom's web page for further information.

## References

- [1] I. Blake, G. Seroussi, N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society, Lecture Note Series 265, Cambridge University Press, 1999
- [2] S. Hamdy, *Anwendungen elliptischer Kurven in der Kryptologie*, Master thesis, 1998
- [3] K. Ireland, M. Rosen, *A classical introduction to modern number theory*, Graduate Texts in Mathematics, Vol. 84, Springer, 1990
- [4] D. Johnson, A. Menezes, *The Elliptic Curve Digital Signature Algorithm (ECDSA)*. Technical Report CORR 99-34, Dept. of C&O, University of Waterloo, Canada.
- [5] V. Miller, *Elliptic Curves and their use in Cryptography*, 1997
- [6] N. Koblitz, *Algebraic Aspects of Cryptography*, Springer-Verlag, ACM Volume 3, 1998
- [7] N. Koblitz, *A Course in Number Theory and Cryptography*, Second Edition, Graduate Texts in Mathematics, Vol. 114, Springer, 1994
- [8] N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, Graduate Texts in Mathematics, Vol. 97, Springer, 1984
- [9] F. Lemmermeyer, *Elliptische Kurven I*, Lecture Notes, 1998
- [10] W. M. Ruppert, *Elliptische Kurven und Kryptographie*, Lecture Notes, Summerterm 1998
- [11] W. M. Ruppert, *Elliptische Kurven und Kryptographie*, Lecture Notes, Winterterm 1999/2000
- [12] J. H. Silverman. *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, Vol. 106, Springer, 1986