# 13 CHAPTER Other Systems

The security of public-key cryptosystems is based on the intractability of certain computational problems. The security of the RSA and Rabin schemes is based on the hardness of integer factorization. The security of the ElGamal protocols and of DSA is based on the intractability of computing discrete logarithms in finite prime fields. However, none of those computational problems is provably intractable. Algorithmic progress has almost always been faster than predicted and it is known that quantum computers will make integer factorization and discrete logarithm computation in the relevant groups easy. Therefore, it is necessary to find public-key cryptosystems that are based on new intractable problems. In particular, it is necessary to find public key cryptosystems that remain secure even when quantum computers can be built.

As we have described in Sections 8.5.5 and 12.5.9, the ElGamal cryptosystem and signature scheme can be implemented in groups in which the discrete logarithm problem is hard to solve. In this chapter we describe a few possible groups. However, also those new discrete logarithm problems can efficiently be solved by quantum computers. Cryptosystems that may be resistant against quantum attacks are based on the intractability of findig short vectors in lattices. For an overview over lattice based cryptography we refer to [50]. For other alternative cryptosystems see also [40].

## 13.1 Finite Fields

We show that the ElGamal algorithms can be implemented in the unit group of any finite field, not only of the prime field $\mathbb{Z}/p\mathbb{Z}$ for a prime $p$.

### 13.1.1 DL problem

Let $p$ be a prime number and let $n$ be a positive integer. In Theorem 2.21.1, we have shown that the unit group of the finite field $\mathrm{GF}(p^n)$ is cyclic. Its order is $p^n - 1$. If this order has only small prime factors, then the Pohlig-Hellman DL algorithm will efficiently compute discrete logarithms in this group (see Section 10.5). Otherwise, an index calculus algorithm can be applied (see Section 10.6). For fixed $n$, the number field sieve can be applied. For fixed $p$ and growing $n$, the function field sieve is used. An overview can be found in [63]. Both algorithms have running time $L_q[1/3, c+o(1)]$ (see Section 9.4), where $c$ is a constant and $q = p^n$. If $p$ and $q$ grow simultaneously, then the best-known algorithm has only running time $L_q[1/2, c + o(1)]$.

## 13.2 Elliptic Curves

Elliptic curves can be defined over any field. In cryptography, elliptic curves over finite fields are of particular interest. To make things simple, we only describe elliptic curves over prime fields. For more details concerning elliptic curve cryptosystems we refer to [39], [48], and [11].

### 13.2.1 Definition

Let $p$ be a prime number, $p > 3$ and let $a, b \in \mathrm{GF}(p)$. Consider the equation

$$y^2z = x^3 + axz^2 + bz^3. \tag{13.1}$$

Its *discriminant* is

$$\Delta = -16(4a^3 + 27b^2). \tag{13.2}$$

We assume that $\Delta$ is nonzero. If $(x, y, z) \in \mathrm{GF}(p)^3$ is a solution of this equation, then for any $c \in \mathrm{GF}(p)$ also $c(x, y, z)$ is a solution. Two solutions $(x, y, z)$ and $(x', y', z')$ are called *equivalent* if there is a nonzero $c \in \mathrm{GF}(p)$ with $(x, y, z) = c(x', y', z')$. This defines an equivalence relation on the set of all solutions of (13.1). The equivalence class of $(x, y, z)$ is denoted by $(x : y : z)$. The *elliptic curve* $E(p; a, b)$ is the set of all equivalence classes of solutions of (13.1). Each element of this set is called a *point* on the curve.

**Example 13.2.1**
We work in GF(11). The elements are represented by their smallest nonnegative representatives. Over this field, we consider the equation

$$y^2 = x^3 + x + 6. \tag{13.3}$$

We have $a = 1$ and $b = 6$. The discriminant is $\Delta = -16*(4+27*6^2) = 4$. Hence, (13.3) defines an elliptic curve over GF(11). It is

$$\begin{aligned} E(11; 1, 6) = \{&\mathcal{O}, (2, 4), (2, 7), (3, 5), (3, 6), (5, 2), (5, 9), (7, 2),\\ &(7, 9), (8, 3), (8, 8), (10, 2), (10, 9)\}. \end{aligned}$$

We simplify the description of the elliptic curve. If $(x', y', z')$ is a solution of (13.1) and if $z' \neq 0$, then $(x' : y' : z')$ contains exactly one element $(x, y, 1)$. Here $(x, y)$ is a solution of the equation

$$y^2 = x^3 + ax + b. \tag{13.4}$$

Conversely, if $(x, y) \in \mathrm{GF}(p)^2$ is a solution of (13.4), then $(x, y, 1)$ is a solution of (13.1). Moreover, there is exactly one equivalence class of solutions of (13.1) which are all of the form $(x, y, 0)$. In fact, if

$z = 0$, then we also have $x = 0$, so this equivalence class is $(0 : 1 : 0)$. Hence, we can write the elliptic curve as

$$E(p; a, b) = \{(x : y : 1) : y^2 = x^3 + ax + b\} \cup \{(0 : 1 : 0)\}.$$

We also write $(x, y)$ instead of $(x : y : 1)$ and $\mathcal{O}$ instead of $(0 : 1 : 0)$, so

$$E(p; a, b) = \{(x, y) : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

### 13.2.2 Group structure

Let $p$ be a prime number, $p > 3$, $a, b \in \mathrm{GF}(p)$ and let $E(p; a, b)$ be an elliptic curve. We define the addition of points on that curve.

For a point $P$ on the curve, we set

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

Hence, the point $\mathcal{O}$ is a neutral element with respect to this addition.

Let $P$ be a point different from $\mathcal{O}$, $P = (x, y)$. Then $-P = (x, -y)$ and we set $P + (-P) = \mathcal{O}$.

Let $P_1$ and $P_2$ be points on the curve that are different from $\mathcal{O}$ and satisfy $P_2 \neq \pm P_1$. Let $P_i = (x_i, y_i)$, $i = 1, 2$. Then the sum

$$P_1 + P_2 = (x_3, y_3)$$

is defined as follows. If

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{falls } P \neq Q, \\ \\ \frac{3x_1^2 + a}{2y_1}, & \text{falls } P = Q, \end{cases}$$

then

$$x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1.$$

It can be shown that with this addition $E(p; a, b)$ is an abelian group.

**Example 13.2.2**
We use the curve from Example 13.2.1 and compute the sum $(2, 4) + (2, 7)$. Since $(2, 7) = -(2, 4)$, we have $(2, 4) + (2, 7) = \mathcal{O}$. Next, we compute $(2, 4) + (3, 5)$. We obtain $\lambda = 1$ and $x_3 = -4 = 7$, $y_3 = 2$.

Hence, $(2, 4) + (3, 5) = (7, 2)$. Finally, we have $(2, 4) + (2, 4) = (5, 9)$, as the reader can easily verify.

### 13.2.3 Cryptographically secure curves

Again, let $p$ be a prime number $p > 3$, $a, b \in \mathrm{GF}(p)$ and $E(p; a, b)$ an elliptic curve. In the group $E(p; a, b)$, the Diffie-Hellman key-exchange system (see Section 8.5) and the ElGamal encryption and signature schemes (see Sections 8.5.5 and 12.5.9) can be implemented.

Those implementations are only secure if the discrete logarithm problem in $E(p; a, b)$ is difficult. Currently, the fastest DL algorithm on elliptic curves is the Pohlig-Hellman algorithm (see Section 10.5). This algorithm has exponential complexity. For special curves, the so-called *supersingular* and *anomalous* curves, faster algorithms are known.

To obtain an elliptic curve cryptosystem or signature scheme that is as secure as a 1024-bit RSA system, curves are used with approximately $2^{163}$ points. To prevent a Pohlig-Hellman attack, a prime factor $q \geq 2^{160}$ of the group order is required. We briefly describe how such a curve can be found.

The number of points on the curve $E(p; a, b)$ is estimated in the following theorem.

**Theorem 13.2.3 (Hasse)**
*We have* $|E(p; a, b)| = p + 1 - t$ *with* $|t| \leq 2\sqrt{p}$.

The theorem of Hasse guarantees that the elliptic curve $E(p; a, b)$ has approximately $p$ points. To obtain a curve with $2^{163}$ points, we choose $p \sim 2^{163}$. If $p$ is fixed, then the coefficients $a$ and $b$ are chosen at random. Then the order of $E(p; a, b)$ is determined. This is possible in polynomial time and takes a couple of minutes per curve. If the curve is supersingular, anomalous, or its order has no prime factor $q \geq 2^{160}$, then a new curve is generated. Otherwise, the curve is accepted.

There are more efficient ways of generating cryptographically secure curves.

### 13.2.4 Advantages of EC cryptosystems

There are several reasons to use elliptic curve cryptosystems.

Elliptic curve public-key systems are currently the most important alternative to RSA systems. Such alternatives are necessary since one day RSA may become insecure.

Elliptic curve systems have efficiency advantages over RSA and finite field systems. While in the latter systems modular arithmetic with 1024-bit numbers is used, the arithmetic on cryptographically secure elliptic curves works with 163-bit numbers. This is an efficiency advantage, although group operations on elliptic curves are more complicated than group operations in prime fields. Also, keys in elliptic curve systems are much smaller than keys in RSA and finite field systems.

## 13.3 Quadratic Forms

Class groups of binary quadratic forms or, more generally, class groups of algebraic number fields can also be used to implement cryptographic algorithms (see [16] and [17]).

In some respects, class groups are different from the unit groups of finite fields and point groups of elliptic curves. The order of the unit group of GF($p^n$) is $p^n - 1$. The order of an elliptic curve can be computed in polynomial time, but no efficient algorithm is known for computing the order of a class group. The known algorithms for solving this problem are closely related to DL algorithms in class groups and no more efficient. Also, class groups may be very small. However, if a class group is small it is very difficult to decide whether two elements in the class group are equal. There are cryptographic protocols that use this difficulty.

## 13.4 Exercises

**Exercise 13.4.1**
Construct the finite field GF(9) with its addition and multiplication tables.

**Exercise 13.4.2**

1. Construct GF(125) and determine a generating element for the multiplicative group GF(125)*.
2. Determine a valid secret and public key for the ElGamal signature system in GF(125)*.

**Exercise 13.4.3**
Determine the number of points on the elliptic curve $y^2 = x^3 + x + 1$ over GF(7). Is the group of points on that curve cyclic? If it is cyclic, determine a generator of this curve.

**Exercise 13.4.4**
Let $p$ be a prime number, $p \equiv 3 \bmod 4$ and let $E$ be an elliptic curve over GF($p$). Find a polynomial time algorithm that, given $x \in$ GF($p$) computes a point $(x, y)$ on $E$ if it exists. Hint: use Exercise 2.23.21. Use this algorithm to find a point $(2, y)$ on $E(111119; 1, 1)$.