2. D. Husemöller, *Elliptic Curves*, Springer–Verlag, 1987.
3. N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, 2nd ed., Springer–Verlag, 1993.
4. N. Koblitz, "Why study equations over finite fields?," *Math. Magazine* **55** (1982), 144–149.
5. S. Lang, *Elliptic Curves: Diophantine Analysis*, Springer–Verlag, 1978.
6. J. Silverman, *The Arithmetic of Elliptic Curves*, Springer–Verlag, 1986.

## 2 Elliptic curve cryptosystems

In § IV.3 we saw how the finite abelian group $\mathbf{F}_q^*$ — the multiplicative group of a finite field — can be used to create public key cryptosystems. More precisely, it was the difficulty of solving the discrete logarithm problem in finite fields that led to the cryptosystems discussed in § IV.3. The purpose of this section is to make analogous public key systems based on the finite abelian group of an elliptic curve $E$ defined over $\mathbf{F}_q$.

Before introducing the cryptosystems themselves, there are some preliminary matters that must be discussed.

**Multiples of points.** The elliptic curve analogy of multiplying two elements of $\mathbf{F}_q^*$ is *adding* two points on $E$, where $E$ is an elliptic curve defined over $\mathbf{F}_q$. Thus, the analog of raising to the $k$-th power in $\mathbf{F}_q^*$ is multiplication of a point $P \in E$ by an integer $k$. Raising to the $k$-th power in a finite field can be accomplished by the repeated squaring method in $O(log\, k\, log^3 q)$ bit operations (see Proposition II.1.9). Similarly, we shall show that the multiple $kP \in E$ can be found in $O(log\, k\, log^3 q)$ bit operations by the method of repeated doubling.

**Example 1.** To find $100P$ we write $100P = 2(2(P+2(2(2(P+2P)))))$, and end up performing 6 doublings and 2 additions of points on the curve.

**Proposition VI.2.1.** *Suppose that an elliptic curve $E$ is defined by a Weierstrass equation (equation (1), (2) or (3) in the last section) over a finite field $\mathbf{F}_q$. Given $P \in E$, the coordinates of $kP$ can be computed in $O(log\, k\, log^3 q)$ bit operations.*

**Proof.** Note that there are fewer than 20 computations in $\mathbf{F}_q$ (multiplications, divisions, additions, or subtractions) involved in computing the coordinates of a sum of two points by means of equations (4)–(5) (or the analogous equations in Exercise 6 of §1). Thus, by Proposition II.1.9, each such addition (or doubling) of points takes time $O(log^3 q)$. Since there are $O(log\, k)$ steps in the repeated doubling method (see the proof of Proposition I.3.6), we conclude that the coordinates of $kP$ can be calculated in $O(log\, k\, log^3 q)$ bit operations.

**Remarks. 1.** The time estimate in Proposition VI.2.1 is not the best possible, especially in the case when our finite field has characteristic $p = 2$. But we shall be satisfied with the estimates that result from using the most obvious algorithms for arithmetic in finite fields.