2. If we happen to know the number N of points on our elliptic curve E, and if $k > N$, then since $NP = O$ we can replace k by its least nonnegative residue modulo N before computing kP; in this case we can replace the time estimate by $O(log^4 q)$ (recall that $N \leq q + 1 + 2\sqrt{q} = O(q)$). There is an algorithm due to René Schoof which computes N in $O(log^8 q)$ bit operations.

Imbedding plaintexts. We shall want to encode our plaintexts as points on some given elliptic curve E defined over a finite field $\mathbf{F}_q$. We want to do this in a simple systematic way, so that the plaintext m (which we may regard as an integer in some range) can readily be determined from knowledge of the coordinates of the corresponding point P_m. Notice that this "encoding" is not the same thing as encryption. Later we shall discuss ways to encrypt the plaintext points P_m. But an authorized user of the system must be able to recover m after deciphering the ciphertext point.

There are two remarks that should be made here. In the first place, there is no polynomial time (in $log\, q$) *deterministic* algorithm known for writing down a large number of points on an arbitrary elliptic curve E over $\mathbf{F}_q$. However, there are probabilistic algorithms for which the chance of failure is very small, as we shall see below. In the second place, it is not enough to generate random points of E: in order to encode a large number of possible messages m, we need a systematic way to generate points that are related to m in some way, for example, the x-coordinate has a simple relationship to the integer m.

Here is one possible probabilistic method to imbed plaintexts as points on an elliptic curve E defined over $\mathbf{F}_q$, where $q = p^r$ is assumed to be large (and odd; see Exercise 8 below for $q = 2^r$). Let κ be a large enough integer so that we are satisfied with a failure probability of 1 out of 2^κ when we attempt to imbed a plaintext message unit m; in practice $\kappa = 30$ or at worse $\kappa = 50$ should suffice. We suppose that our message units m are integers $0 \leq m < M$. We also suppose that our finite field is chosen so that $q > M\kappa$. We write the integers from 1 to $M\kappa$ in the form $m\kappa + j$, where $1 \leq j \leq \kappa$, and we set up a 1-to-1 correspondence between such integers and a set of elements of $\mathbf{F}_q$. For example, we write such an integer as an r-digit integer to the base p, and take the r digits, considered as elements of $\mathbf{Z}/p\mathbf{Z}$, as the coefficients of a polynomial of degree $r-1$ corresponding to an element of $\mathbf{F}_q$. That is, the integer $(a_{r-1}a_{r-2}\cdots a_1 a_0)_p$ corresponds to the polynomial $\sum_{i=0}^{r-1} a_j X^j$, which, considered modulo some fixed degree-r irreducible polynomial over $\mathbf{F}_p$, gives an element of $\mathbf{F}_q$.

Thus, given m, for each $j = 1, 2, \ldots, \kappa$ we obtain an element x of $\mathbf{F}_q$ corresponding to $m\kappa + j$. For such an x, we compute the right side of the equation

$$y^2 = f(x) = x^3 + ax + b,$$

and try to find a square root of $f(x)$ using the method explained at the end of § II.2. (Although the algorithm was given for the prime field $\mathbf{F}_p$, it carries