walks records every 10th point, for example, in the first sequence and then checks whether the second sequence matches any of these points. In this case, the first sequence is called a tame kangaroo, and the second is called a wild kangaroo. The idea is to use the tame kangaroo to catch the wild kangaroo.

The λ method is expected to find a match in at most a constant times $\sqrt{N}$ steps. If it is run in parallel with many starting points, the running time can be improved significantly.

Finally, we should point out a difference between the baby step, giant step method and the ρ and λ methods. The baby step, giant step method is **deterministic**, which means that it is guaranteed to finish within the predicted time of a constant times $\sqrt{N}$. On the other hand, the ρ and λ methods are **probabilistic**, which means that there is a very high probability that they will finish within the predicted time, but this is not guaranteed.

5.2.3 The Pohlig-Hellman Method

As before, P, Q are elements in a group G and we want to find an integer k with $Q = kP$. We also know the order N of P and we know the prime factorization

$$N = \prod_i q_i^{e_i}$$

of N. The idea of Pohlig-Hellman is to find $k \pmod{q_i^{e_i}}$ for each i, then use the Chinese Remainder theorem to combine these and obtain $k \pmod{N}$.

Let q be a prime, and let q^e be the exact power of q dividing N. Write k in its base q expansion as

$$k = k_0 + k_1 q + k_2 q^2 + \cdots$$

with $0 \le k_i < q$. We'll evaluate $k \pmod{q^e}$ by successively determining $k_0, k_1, \ldots, k_{e-1}$. The procedure is as follows.

1. Compute $T = \left\{ j\left(\frac{N}{q}P\right) \mid 0 \le j \le q-1 \right\}$.
2. Compute $\frac{N}{q}Q$. This will be an element $k_0\left(\frac{N}{q}P\right)$ of T.
3. If $e = 1$, stop. Otherwise, continue.
4. Let $Q_1 = Q - k_0 P$.
5. Compute $\frac{N}{q^2}Q_1$. This will be an element $k_1\left(\frac{N}{q}P\right)$ of T.
6. If $e = 2$, stop. Otherwise, continue.
7. Suppose we have computed $k_0, k_1, \ldots, k_{r-1}$, and $Q_1, \ldots, Q_{r-1}$.

8. Let $Q_r = Q_{r-1} - k_{r-1}q^{r-1}P$.

9. Determine k_r such that $\frac{N}{q^{r+1}}Q_r = k_r\left(\frac{N}{q}P\right)$.

10. If $r = e - 1$, stop. Otherwise, return to step (7).

Then

$$k \equiv k_0 + k_1 q + \cdot + k_{e-1}q^{e-1} \pmod{q^e}.$$

Why does this work? We have

$$\begin{aligned}\frac{N}{q}Q &= \frac{N}{q}(k_0 + k_1 q + \cdots)P \\ &= k_0\frac{N}{q}P + (k_1 + k_2 q + \cdots)NP = k_0\frac{N}{q}P,\end{aligned}$$

since $NP = \infty$. Therefore, step (2) finds k_0. Then

$$Q_1 = Q - k_0 P = (k_1 q + k_2 q^2 + \cdots)P,$$

so

$$\begin{aligned}\frac{N}{q^2}Q_1 &= (k_1 + k_2 q + \cdots)\frac{N}{q}P \\ &= k_1\frac{N}{q}P + (k_2 + k_3 q + \cdots)NP = k_1\frac{N}{q}P.\end{aligned}$$

Therefore, we find k_1. Similarly, the method produces $k_2, k_3, \ldots$. We have to stop after $r = e - 1$ since N/q^{e+1} is no longer an integer, and we cannot multiply Q_e by the noninteger N/q^{e+1}. Besides, we do not need to continue because we now know $k \bmod q^e$.

Example 5.4

Let $G = E(\mathbf{F}_{599})$, where E is the elliptic curve given by $y^2 = x^3 + 1$. Let $P = (60, 19)$ and $Q = (277, 239)$. The methods of Section 4.3.3 can be used to show that P has order $N = 600$. We want to solve $Q = kP$ for k. The prime factorization of N is

$$600 = 2^3 \cdot 3 \cdot 5^2.$$

We'll compute k mod 8, mod 3, and mod 25, then recombine to obtain k mod 600 (the Chinese Remainder Theorem allows us to do this).

k mod 8. We compute $T = \{\infty, (598, 0)\}$. Since

$$(N/2)Q = \infty = 0 \cdot \left(\frac{N}{2}P\right),$$

we have $k_0 = 0$. Therefore,

$$Q_1 = Q - 0P = Q.$$

Since $(N/4)Q_1 = 150Q_1 = (598, 0) = 1 \cdot \frac{N}{2}P$, we have $k_1 = 1$. Therefore,

$$Q_2 = Q_1 - 1 \cdot 2 \cdot P = (35, 243).$$

Since $(N/8)Q_2 = 75Q_2 = \infty = 0 \cdot \frac{N}{2}P$, we have $k_2 = 0$. Therefore,

$$k = 0 + 1 \cdot 2 + 0 \cdot 4 + \cdots \equiv 2 \pmod 8.$$

k mod 3. We have $T = \{\infty, (0, 1), (0, 598)\}$. Since

$$(N/3)Q = (0, 598) = 2 \cdot \frac{N}{3}P,$$

we have $k_0 = 2$. Therefore,

$$k \equiv 2 \pmod 3.$$

k mod 25. We have

$$T = \{\infty, (84, 179), (491, 134), (491, 465), (84, 420)\}.$$

Since $(N/5)Q = (84, 179)$, we have $k_0 = 1$. Then

$$Q_1 = Q - 1 \cdot P = (130, 129).$$

Since $(N/25)Q_1 = (491, 465)$, we have $k_1 = 3$. Therefore,

$$k = 1 + 3 \cdot 5 + \cdots \equiv 16 \pmod{25}.$$

We now have the simultaneous congruences

$$\begin{cases} x \equiv \ \ 2 \pmod 8 \\ x \equiv \ \ 2 \pmod 3 \\ x \equiv 16 \pmod{25} \end{cases}.$$

These combine to yield $k \equiv 266 \pmod{600}$, so $k = 266$. $\square$

The Pohlig-Hellman method works well if all of the prime numbers dividing N are small. However, if q is a large prime dividing N, then it is difficult to list the elements of T, which contains q elements. We could try to find the k_i without listing the elements; however, finding k_i is a discrete log problem in the group generated by $(N/q)P$, which has order q. If q is of the same order of magnitude as N (for example, $q = N$ or $q = N/2$), then the Pohlig-Hellman method is of little use. For this reason, if a cryptographic system is based on discrete logs, the order of the group should be chosen so it contains a large prime factor.

If N contains some small prime factors, then the Pohlig-Hellman method can be used to obtain partial information on the value of k, namely a congruence modulo a product of these small prime factors. In certain cryptographic situations, this could be undesirable. Therefore, the group G is often chosen to be of large prime order. This can be accomplished by starting with a group that has a large prime q in its order. Pick a random point P_1 and compute its order. With high probability (at least $1 - 1/q$; cf. Remark 5.2), the order of P_1 is divisible by q, so in a few tries, we can find such a point P_1. Write the order of P_1 as qm. Then $P = mP_1$ will have order q. As long as q is sufficiently large, discrete log problems in the cyclic group generated by P will resist the Pohlig-Hellman attack.

5.3 Attacks with Pairings

One strategy for attacking a discrete logarithm problem is to reduce it to an easier discrete logarithm problem. This can often be done with pairings such as the Weil pairing or the Tate-Lichtenbaum pairing, which reduce a discrete logarithm problem on an elliptic curve to one in the multiplicative group of a finite field.

5.3.1 The MOV Attack

The MOV attack, named after Menezes, Okamoto, and Vanstone [80], uses the Weil pairing to convert a discrete log problem in $E(\mathbf{F}_q)$ to one in $\mathbf{F}_{q^m}^{\times}$. Since discrete log problems in finite fields can be attacked by index calculus methods, they can be solved faster than elliptic curve discrete log problems, as long as the field $\mathbf{F}_{q^m}$ is not much larger than $\mathbf{F}_q$. For supersingular curves, we can usually take $m = 2$, so discrete logarithms can be computed more easily for these curves than for arbitrary elliptic curves. This is unfortunate from a cryptographic standpoint since an attractive feature of supersingular curves is that calculations can often be done quickly on them (see Section 4.6).

Recall that for an elliptic curve E defined over $\mathbf{F}_q$, we let $E[N]$ denote the set of points of order dividing N with coordinates in the algebraic closure. If $\gcd(q, N) = 1$ and $S, T \in E[N]$, then the Weil pairing $e_N(S, T)$ is an Nth root of unity and can be computed fairly quickly. The pairing is bilinear, and if $\{S, T\}$ is a basis for $E[N]$, then $e_N(S, T)$ is a primitive Nth root of unity. For any S, $e_N(S, S) = 1$. For more properties of the Weil pairing, see Sections 3.3 and 11.2.