A. Speeding up addition and doubling

Let K be a field and let C be a smooth Weierstraß curve over K given by

$$x_2^2 + a_1x_1x_2 + a_3x_2 = x_1^3 + a_2x_1^2 + a_4x_1 + a_6. \tag{A.1}$$

Then all K-rational points of C together with ∞, the point at infinity, can be interpreted as an abelian group. This group will be denoted by $C(K)$, the addition in this group by $\oplus$, the opposite elements by $\ominus$, and $[m]$ will be used when the addition is repeated m-times. The neutral element of $C(K)$ is the point at infinity ∞. Thus $\alpha \oplus \infty = \infty \oplus \alpha$ for all $\alpha \in C(K)$.

The group $C(K)$ may be also interpreted as a group on all projective K-rational points of C. Under this approach every affine K-rational point (α_1, α_2) is identified with $(\alpha_1 : \alpha_2 : 1)$, and ∞ with $(0 : 1 : 0)$.

Suppose that $\alpha = (\alpha_1, \alpha_2)$ and $\beta = (\beta_1, \beta_2)$ are K-rational affine points of C. Then:

$$\ominus\alpha = (\alpha_1, -\alpha_2 - \alpha_1a_1 - a_3). \tag{A.2}$$

If $\beta = \ominus\alpha$, then $\beta \oplus \alpha = \infty$. Suppose that $\beta \neq \ominus\alpha$. To define $\gamma = \alpha \oplus \beta$, $\gamma = (\gamma_1, \gamma_2)$, first set

$$\lambda = \frac{3\alpha_1^2 + 2a_2\alpha_1 - a_1\alpha_2 + a_4}{2\alpha_2 + a_1\alpha_1 + a_3} \text{ if } \alpha = \beta, \text{ and } \lambda = \frac{\beta_2 - \alpha_2}{\beta_1 - \alpha_1} \text{ if } \alpha \neq \beta. \tag{A.3}$$

The value of γ_1 depends upon λ, α_1, β_1, a_1 and a_2, and γ_2 depends upon λ, γ_1, a_1 and a_3:

$$(\gamma_1, \gamma_2) = (-\alpha_1 - \beta_1 + \lambda^2 + a_1\lambda - a_2,\ \lambda(\alpha_1 - \gamma_1) - \alpha_2 - a_1\gamma_1 - a_3). \tag{A.4}$$

The formulas above describe what is known as the *chord and tangent process*. Let us recall its properties:

(CT1) For each $\alpha = (\alpha_1, \alpha_2) \in C(K)$ there is at most one $\beta = (\beta_1, \beta_2) \in C(K)$ such that $\alpha_1 = \beta_1$ and $\beta \neq \alpha$. If such a β exists, then $\beta = \ominus\alpha$. If no such β exists, then $[2]\alpha = \alpha \oplus \alpha = \infty$ and, thus, $\ominus\alpha = \alpha$. The latter happens if and only if $x_1 = \alpha$ yields the tangent line of C at α.

(CT2) Suppose that $\beta \neq \ominus\alpha$. The choice of λ in (A.3) is such that there exists a (unique) $\mu \in K$ for which $x_2 = \lambda x_1 + \mu$ describes a line that is (1) the tangent of C at α, provided $\alpha = \beta$, and (2) connects α and β, provided $\alpha \neq \beta$.

(CT3) Assume $\beta \neq \ominus\alpha$ and $\gamma = \alpha \oplus \beta = (\gamma_1, \gamma_2)$. We have $\alpha_2 = \lambda\alpha_1 + \mu$, $\mu = \alpha_2 - \lambda\alpha_1$ and $\gamma = (\gamma_1, -(\lambda\gamma_1 + \mu) - a_1\gamma_1 - a_3)$. Therefore $\ominus\gamma = (\gamma_1, \lambda\gamma_1 + \mu)$, by (A.2). All of the points α, β and $\ominus\gamma$ are incident to the line given by $x_2 = \lambda x_1 + \mu$. Denote this line by L. It is a fact that $L \cap C = \{\alpha, \beta, \ominus\gamma\}$.

(CT4) These possibilities can occur:

- The points α, β and $\ominus\gamma$ are pairwise distinct.
- $\alpha \neq \beta$ and $\beta = \ominus\gamma$. Then $\alpha \oplus [2]\beta = \infty$ and $\gamma = \ominus\beta$.
- $\alpha \neq \beta$ and $\alpha = \ominus\gamma$. Then $[2]\alpha \oplus \beta = \infty$ and $\gamma = \ominus\alpha$.
- $\alpha = \beta$ and $\alpha \neq \ominus\gamma$. Then $\gamma = [2]\alpha$.
- $\alpha = \beta = \ominus\gamma$. Then $[3]\alpha = \infty$ and $\gamma = \ominus\alpha = [2]\alpha$.

The natural question is how to perform efficiently both the addition $\alpha \oplus \beta$, and the doubling $[2]\alpha$. Note that the elliptic curve cryptography requires a computation of $[n]\alpha$ for very large n. The point α is usually denoted by P. It remains stable, while n varies. Standard algorithms, e.g. the sliding window, require many applications of doubling. The doubling hence deserves the same attention as the addition of distinct arguments.

For the rest of this section we shall assume that $\mathrm{char}(K) \neq 2$ and that C is given by $x_2^2 = x_1^3 + ax_1 + b$. Thus $a = a_4$, $b = a_6$, $a_1 = a_2 = a_3 = 0$ and $4a^3 + 27b^2 \neq 0$.

Then

$$\ominus(\alpha_1, \alpha_2) = (\alpha_1, -\alpha_2). \tag{A.5}$$

This means that opposite elements are symmetric along the axis x_1 (the line with $x_2 = 0$), and that (α_1, α_2) is of order two if and only if $\alpha_2 = 0$. An element of order two is sometimes called an *involution*.

If $\alpha \oplus \beta \neq \infty$, then there exists $\gamma = (\gamma_1, \gamma_2)$ such that $\gamma = \alpha \oplus \beta$ and

$$\gamma_1 = \lambda^2 - \alpha_1 - \beta_1, \quad \gamma_2 = \lambda(\alpha_1 - \gamma_1) - \alpha_2, \text{ where} \tag{A.6}$$

$$\lambda = \frac{\alpha_2 - \beta_2}{\alpha_1 - \beta_1} \text{ if } \alpha_1 \neq \beta_1, \text{ and } \lambda = \frac{3\alpha_1^2 + a}{2\alpha_2} \text{ if } \alpha_1 = \beta_1. \tag{A.7}$$

Note that the parameter $b = a_6$ has no bearing upon any of the formulas above.

Let us now consider the time needed to perform $\alpha \oplus \beta$, $\alpha \neq \beta$, and to perform $[2]\alpha$. The time will be quantified in the number of needed arithmetical operations over the field K. Typically, K is equal to $\mathbb{F}_p$ for p a large prime. This implies that these operations are not built-in, but have to be algorithmically computed. If $\xi, \eta \in K$, then there exist algorithms which compute ξ^2 somewhat more quickly then $\xi\eta$. We shall use S for squaring ξ^2, M for multiplying $\xi\eta$, and I for inversion ξ^{-1}. An addition $\xi + \eta$ and/or a subtraction $\xi - \eta$ will be neglected since it is much more quicker than multiplication.

The cost of $\alpha \oplus \beta$ is I + 2M + S. Indeed, an inversion is needed to compute $(\alpha_1 - \beta_1)^{-1}$. If this is done, then a multiplication is needed to get λ. A squaring appears when computing γ_1, and one more multiplication appears in the formula that expresses γ_2. Small multiples can be replaced by additions. That makes the cost of doubling I + 2M + 2S.

To find an inversion modulo a prime means to employ the extended Euclidean algorithm. This includes many multiplications. Hence replacing I by kM, where k is fixed (and not too big) causes a significant speed-up. Such a speed-up is possible, but at a price. The price is that a point $\alpha = (\alpha_1, \alpha_2)$ may be addressed in several ways (using a triple or a quadruple instead of the pair (α_1, α_2)). That may pay off only if there are many intermediary stages at which the lack of uniqueness of point identification does not cause a difficulty. At the end an inversion usually cannot be avoided if the goal is to get a uniquely determined result. However, when computing $[n]P$, say in a cryptographic application, then the computation uses many additions and doublings that are of intermediary character. For such situations projective or Jacob or Chudonovski coordinates may be used.

A.1. **Projective coordinates.** The projective description of C is by the equation

$$X_2^2 X_3 = X_1^3 + aX_1X_3^2 + bX_3^3. \tag{A.8}$$

Let $\alpha = (\alpha_1 : \alpha_2 : \alpha_3) = (\alpha_1/\alpha_3 : \alpha_2/\alpha_3 : 1)$ and $\beta = (\beta_1 : \beta_2 : \beta_3) = (\beta_1/\beta_3, \beta_2/\beta_3 : 1)$ be two distinct points on C. Assume that $\alpha \oplus \beta \neq (0 : 1 : 0)$. Then $\alpha \oplus \beta = \gamma = (\gamma_1 : \gamma_2 : \gamma_3) = (\gamma_1/\gamma_3 : \gamma_2/\gamma_3 : 1)$. By (A.6)

$$\frac{\gamma_1}{\gamma_3} = \lambda^2 - \frac{\alpha_1}{\alpha_3} - \frac{\beta_1}{\beta_3} \text{ and } \frac{\gamma_2}{\gamma_3} = \lambda\left(\frac{\alpha_1}{\alpha_3} - \frac{\gamma_1}{\gamma_3}\right) - \frac{\alpha_2}{\alpha_3}, \tag{A.9}$$

where, by (A.7),

$$\lambda = \frac{\alpha_2/\alpha_3 - \beta_2/\beta_3}{\alpha_1/\alpha_3 - \beta_1/\beta_3} = \frac{\alpha_2\beta_3 - \beta_2\alpha_3}{\alpha_1\beta_3 - \beta_1\alpha_3}.$$

Put $U = \alpha_2\beta_3 - \beta_2\alpha_3$ and $V = \alpha_1\beta_3 - \beta_1\alpha_3$. The cost of computing U and V is 4M. The cost of computing

$$W = U^2\alpha_3\beta_3 - V^2(\alpha_1\beta_3 + \beta_1\alpha_3)$$

is $2\mathrm{S} + 7\mathrm{M}$ since $\alpha_1\beta_3$ and $\beta_1\alpha_3$ may be regarded as precomputed. Since $\alpha_1\beta_3 + \beta_1\alpha_3 = (\beta_1\alpha_3 - \alpha_1\beta_3) + 2\alpha_1\beta_3 = -V + 2\alpha_1\beta_3$ we also have

$$W = U^2\alpha_3\beta_3 + V^3 - 2\alpha_1\beta_3V^2. \tag{A.10}$$

If this formula is followed, the cost of W is $2\mathrm{S} + 8\mathrm{M}$.

Put $\gamma_3 = V^3\alpha_3\beta_3$. Note that $\lambda = U/V$. Then

$$\gamma_1 = V(U^2\alpha_3\beta_3) - V^3(\alpha_1\beta_3 + \beta_1\alpha_3) = VW, \text{ and}$$
$$\gamma_2 = (U/V)(V^3\alpha_1\beta_3 - VW) - V^3\alpha_2\beta_3 = U(\alpha_1\beta_3V^2 - W) - \alpha_2\beta_3V^3.$$

Compute W by means of (A.10) and use precomputed values to get γ_3, γ_1 and γ_2. The cost is 1M, 1M and 2M, respectively. The overall cost of computing $\gamma = (\gamma_1, \gamma_2)$ thus amounts to $2\mathrm{S} + 12\mathrm{M}$.

Formula (A.9) can be used for the doubling as well, with $\alpha = \beta$. However, in this case

$$\lambda = \frac{3(\alpha_1/\alpha_3)^2 + a}{2\alpha_2/\alpha_3} = \frac{3\alpha_1^2 + a\alpha_3^2}{2\alpha_2\alpha_3}.$$

The form of γ_2/γ_3 suggests to choose γ_3 as $8\alpha_2^3\alpha_3^3$. Then

$$\gamma_1 = 2\alpha_2\alpha_3((3\alpha_1^2 + a\alpha_3^2)^2 - 8\alpha_1\alpha_2^2\alpha_3), \text{ and}$$
$$\gamma_2 = (3\alpha_1^2 + a\alpha_3^2)(4\alpha_1\alpha_2^2\alpha_3 - \gamma_1/2\alpha_2\alpha_3) - 8\alpha_2^4\alpha_3^2$$
$$= (3\alpha_1^2 + a\alpha_3^2)(4\alpha_1\alpha_2^2\alpha_3 - ((3\alpha_1^2 + a\alpha_3^2)^2 - 8\alpha_1\alpha_2^2\alpha_3)) - 8\alpha_2^4\alpha_3^2.$$

To compute γ_i, $1 \le i \le 3$, it may be proceeded by computing (1) α_1^2, (2) α_3^2, (3) $U = 3\alpha_1^2 + a\alpha_3^2$, (4) U^2, (5) $V = 2\alpha_2\alpha_3$, (6) α_2V, (7) V^2, (8) $\gamma_3 = V^3$, (9) $W = U^2 - 4\alpha_1\alpha_2V$, (10) $\gamma_1 = VW$, (11) $(\alpha_2V)^2$ and (12) $\gamma_2 = U(2\alpha_1\alpha_2V - W) - 2(\alpha_2V)^2$. The cost of doubling hence is $5\mathrm{S} + 7\mathrm{M}$. If a is small, then the cost of multiplying by a may be regarded as negligible. In such a case the cost of doubling is equal to $5\mathrm{S} + 6\mathrm{M}$.

When computing $[n]P$ it often happens that the point P is being added to an intermediary result. If the intermediary result is denoted by α, and the point P as β, then $\beta_3 = 1$ since P is given as an affine point. By inspecting the above procedure for computing $\alpha \oplus \beta$ it may be observed that it includes exactly three instances of multiplying by β_3. The cost of computing $\alpha \oplus \beta$ is thus reduced to $2\mathrm{S} + 9\mathrm{M}$ if $\beta_3 = 1$.