example we had $5\mathbf{M} + 6\mathbf{S}$ for doublings and $12\mathbf{M} + 2\mathbf{S}$ for additions.

The Jacobi-quartic curves discussed above are just one example of dozens of models that have been successful in achieving fast group law computations, and therefore fast cryptographic implementations. Other well known models include Edwards curves [Edw07,BL07b], Hessian curves [JQ01,Sma01] and Montgomery curves [Mon87]. We refer to the EFD [BL07a] for a catalogue of all the fastest formulas for the popular curve models, and to Hisil's thesis [His10] for a general method of (automatically) deriving fast group law algorithms on arbitrary curve models. For any reader wishing to delve even further into group law arithmetic on elliptic curves, we also recommend the recent, advanced works by Castryck and Vercauteren [CV11], and by Kohel [Koh11].

2.2 Torsion, endomorphisms and point counting

We now turn our focus to the behaviour of elliptic curve groups, as they are used in cryptography. We start by importantly discussing the possible structures exhibited by the finite group $E(\mathbb{F}_q)$. It turns out that $E(\mathbb{F}_q)$ is either itself cyclic, or isomorphic to a product of two cyclic groups $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ with $n_1 \mid n_2$ [ACD$^+$05, Prop. 5.78]. In cryptography, we would like the group $E(\mathbb{F}_q)$ to be *as cyclic as possible*, so we usually prefer the former case, or at the very least for n_1 to be very small. In most cases of practical interest, we can generate curves that are cyclic with relative ease, so throughout this thesis it is to safe assume that $E(\mathbb{F}_q)$ is cyclic (but to see the real depth of this question in general, we refer to [MS07]). The following example illustrates that $E(\mathbb{F}_q) = \langle P \rangle$ obeys all the usual rules that apply to cyclic groups, and introduces the important notion of *r-torsion*.

Example 2.2.1 (Magma script). Consider $E/\mathbb{F}_{101} : y^2 = x^3 + x + 1$. The group order is $\#E(\mathbb{F}_q) = 105 = 3 \cdot 5 \cdot 7$, and $P = (47, 12) \in E$ is a generator. Lagrange's theorem says that points (and subgroups) over the base field will have order in $\{1, 3, 5, 7, 15, 21, 35, 105\}$. Indeed, to get a point of order $r \mid 105$, we simply multiply P by the appropriate *cofactor*, which is $h = \#E/r$. For example, a point of order 3 is $[35](47, 12) = (28, 8)$, a point of order 21 is $[5](47, 12) = (55, 65)$, and a point of order 1 is $[105](47, 12) = \mathcal{O}$ (which is the only such point). By definition, a point is "killed" (sent to $\mathcal{O}$) when multiplied by its order. Any point over the full closure $E(\overline{\mathbb{F}}_q)$ that is killed by r is said to be in the r-torsion. So, the point $(55, 65)$ above is in the 21-torsion, as is the point $(28, 8)$. There are exactly 21 points in $E(\mathbb{F}_q)$ in the 21-torsion, but there are many more in $E(\overline{\mathbb{F}}_q)$.

The whereabouts and structure of r-torsion points in $E(\overline{\mathbb{F}}_q)$ (alluded to at the end of Example 2.2.1) plays a crucial role in pairing-based cryptography; we will be looking at this in close detail in Chapter 4.

In ECC we would like the group order $\#E(\mathbb{F}_q)$ to be as close to prime as possible. This is because the (asymptotic) complexity of the ECDLP that attackers face is dependent on the size of the largest prime subgroup of $E(\mathbb{F}_q)$. Even if the particular instance of the discrete logarithm problem uses a generator of the whole group, the attacker can use the known group order to solve smaller instances in subgroups whose orders are pairwise prime, and then reconstruct the answer using the Chinese Remainder Theorem (CRT). We make this clear in the following two examples: the first is a toy example, whilst the second shows the difference between two curves of the same cryptographic size; one that is currently considered secure and one that is completely breakable using modern attacks.

Example 2.2.2 (Magma script). Consider $E/\mathbb{F}_{1021} : y^2 = x^3 + 905x + 100$, with group order $\#E(\mathbb{F}_q) = 966 = 2 \cdot 3 \cdot 7 \cdot 23$, and generator $P = (1006, 416)$. Suppose we are presented with an instance of the ECDLP: namely, we are given $Q = (612, 827)$, and we seek to find k such that $[k]P = Q$. For the sake of the example, suppose our best "attack" is trivial: trying every multiple $[i]P$ of P until we hit the correct one ($i = k$). Rather than seeking i in the full group ($2 \leq i \leq 965$), we can map the instance into each prime order subgroup by multiplying by the appropriate cofactor, and then solve for $k_j \equiv k \bmod j$, $j \in \{2, 3, 7, 23\}$. For $j = 2$, we have $P_j = P_2 = [966/2]P = [483](1006, 416) = (174, 0)$, and $Q_j = Q_2 = [483](612, 827) = (174, 0)$, so $Q_2 = [k_2]P_2$ gives $k_2 = 1$. For $j = 3$, we have $P_3 = [322]P = (147, 933)$ and $Q_3 = [322]P = \mathcal{O}$, so $Q_3 = [k_3]P_3$ gives $k_3 = 3$. For $j = 7$, we have $P_7 = [138]P = (906, 201)$ and $Q_7 = [138]Q = (906, 201)$, so $Q_7 = [k_7]P_7$ gives $k_7 = 1$. For $j = 23$, we have $P_{23} = [42]P = (890, 665)$ and $Q_{23} = [42]Q = (68, 281)$. For $Q_{23} = [k_{23}]P_{23}$, we exhaust $k_{23} \in \{1, .., 22\}$ to see that $k_{23} = 20$. Now, we can use the Chinese Remainder Theorem to solve

$$k \equiv k_2 = 1 \bmod 2; \quad k \equiv k_3 = 0 \bmod 3; \quad k \equiv k_7 = 1 \bmod 7; \quad k \equiv k_{23} = 20 \bmod 23,$$

which gives $k \equiv 687 \bmod \#E$, solving the ECDLP instance. Notice that the hardest part was exhausting the set $\{1,..,22\}$ to find $k_{23} = 20$, so the largest prime order subgroup becomes the bottleneck of the algorithm, giving intuition as to why the largest prime order subgroup defines the attack complexity when groups of a cryptographic size are used.

Example 2.2.3 (Magma script). For our real world example, we take the curve P-256 from the NIST recommendations [NIS99], which currently achieves a similar security level (resistance against best known attacks) to the 128-bit Advanced Encryption Standard (AES) for symmetric encryption. The curve is defined as $E/\mathbb{F}_q : y^2 = x^3 - 3x + b$, with prime order $r = \#E$, and generator $G = (x_G, y_G)$, where

$$
\begin{aligned}
q &= 115792089210356248762697446949407573530086143415290314195533631308867097853951,\\
r &= 115792089210356248762697446949407573529996955224135760342422259061068512044369,\\
b &= 41058363725152142129326129780047268409114441015993725554835256314039467401291,\\
x_G &= 48439561293906451759052585252797914202762949526041747995844080717082404635286,\\
y_G &= 36134250956749795798585127919587881956611106672985015071877198253568414405109,\\
x_H &= 53987601597021778433910548064987973235945515666715026302948657055639179420355,\\
y_H &= 53690949263410447908824456005055253553237881490194075871737490561466076234637.
\end{aligned}
$$

We give another point $H = (x_H, y_H)$ to pose $H = [k]G$ as an intractable instance of the ECDLP; this 256-bit prime field (and group order) is far beyond the reach of current attacks. For example, there is currently a campaign underway to solve a discrete logarithm problem over a 130-bit field using a cluster of servers that have already been running for two years (see `http://ecc-challenge.info/`), so (assuming the best known attacks stay exponential) it seems the above ECDLP should be safe for a while yet. We remark that the prime characteristic q is given by $q = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$; such primes are preferred in ECC as they allow for faster finite field multiplication and reduction routines, greatly enhancing the speed of $\mathbb{F}_q$ arithmetic. We now give a curve over the same field $\mathbb{F}_q$, for which the ECDLP is well within reach of the best known attacks. Namely, consider the alternative curve with $b = 0$, namely $\tilde{E}/\mathbb{F}_q : y^2 = x^3 - 3x$, whose group order $n = \#\tilde{E}$ is given as

$$\begin{aligned}n &= 115792089210356248762697446949407573530086143415290314195533631308867097853952,\\ &= 2^{96} \cdot 7 \cdot 274177 \cdot 67280421310721 \cdot 11318308927973941931404914103.\end{aligned}$$

This time, the largest prime divisor of the group order is only 94 bits long, and the complexity of solving the ECDLP in $\tilde{E}(\mathbb{F}_q)$ is governed by the difficulty of solving the ECDLP instance in this largest prime subgroup, which could be done in a small amount of time on a desktop computer.

The above example provides clear motivation as to the importance of counting points on elliptic curves. The largest prime factor of the group order determines the difficulty that attackers face when trying to solve the ECDLP, so we would like to be able to count points on curves quickly enough to find those whose order is prime or almost prime (i.e. has a small cofactor), or have methods of prescribing such a group order before searching for the curve. Fortunately, on elliptic curves we have efficient algorithms to do both.

We start our brief discussion on elliptic curve point counting by referring back to the two group orders in Example 2.2.3, and observing that both group orders share the first half of their digits with those of the field characteristic q. This suggests that the number of points on an elliptic curve is close to q, which is indeed the case in general; the *Hasse bound* [Sil09, Ch. 5, Th. 1.1] says the most that $\#E(\mathbb{F}_q)$ can differ from $q+1$ is $2\sqrt{q}$, i.e. $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$. This offset between $\#E(\mathbb{F}_q)$ and $(q+1)$ is called the *trace of Frobenius*, and is denoted by t, so

$$\#E(\mathbb{F}_q) = q + 1 - t, \qquad |t| \leq 2\sqrt{q} \tag{2.6}$$

We will discuss where t comes from and provide some more intuition behind the above formula in a moment, but what the Hasse bound tells us is that the group order lies somewhere in the interval $[q+1-2\sqrt{q}, q+1+2\sqrt{q}]$. In fact, Deuring [Deu41] showed that when q is prime[2], then every value $N \in [q+1-2\sqrt{q}, q+1+2\sqrt{q}]$ can be found as a group order $\#E(\mathbb{F}_q)$ for some E.

Example 2.2.4 (Magma script). Let $q = 23$, so that the Hasse interval becomes $[q+1-2\sqrt{q}, q+1+2\sqrt{q}] = [15, 33]$, meaning that there are exactly 19 different

[2] When q is a prime power, there are a very small number of explicitly described exceptions.

group orders taken by elliptic curves over $\mathbb{F}_{23}$. For example, $E/\mathbb{F}_{23} : y^2 = x^3 + 18x + 3$ has $\#E = 15$, whilst $\tilde{E}/\mathbb{F}_{23} : y^2 = x^3 + 13x + 7$ has $\#\tilde{E} = 33$. We give 19 (a, b) pairs such that the corresponding curves $E : y^2 = x^3 + ax + b$ have group orders in ascending order spanning the whole interval, as follows: $(18, 3)$, $(7, 22)$, $(19, 14)$, $(17, 17)$, $(12, 5)$, $(7, 12)$, $(8, 10)$, $(17, 18)$, $(20, 20)$, $(2, 3)$, $(20, 3)$, $(6, 8)$, $(16, 8)$, $(16, 22)$, $(9, 16)$, $(19, 6)$, $(20, 8)$, $(22, 9)$, $(13, 7)$.

A rough (but elementary and instinctive) argument as to why $\#E \approx q$ is that approximately half of the values $x \in [0, .., q - 1]$ will give a quadratic residue $x^3 + ax + b \in \mathrm{QR}(q)$, which gives rise to two points $(x, \pm\sqrt{x^3 + ax + b}) \in E(\mathbb{F}_q)$, the only exception(s) being when $x^3 + ax + b = 0$ which obtains one point. The sophisticated explanation requires a deeper knowledge than our introduction offers, but for the purposes of this introductory text we get almost all that we need from Equation (2.6); the derivation of which makes use of the following definition. If E is defined over $\mathbb{F}_q$, then the *Frobenius endomorphism* π is defined as

$$\pi : E \to E, \qquad (x, y) \mapsto (x^q, y^q). \tag{2.7}$$

We note that the Frobenius endomorphism maps any point in $E(\overline{\mathbb{F}}_q)$ to a point in $E(\overline{\mathbb{F}}_q)$, but the set of points fixed by π is exactly the group $E(\mathbb{F}_q)$. Thus, π only acts non-trivially on points in $E(\overline{\mathbb{F}}_q) \setminus E(\mathbb{F}_q)$, and more generally, $\pi^i : (x, y) \mapsto (x^{q^i}, y^{q^i})$ only acts non-trivially on points in $E(\overline{\mathbb{F}}_q) \setminus E(\mathbb{F}_{q^i})$.

Example 2.2.5 (Magma script). Let $q = 67$, and consider $E/\mathbb{F}_q : y^2 = x^3+4x+3$, and let $\mathbb{F}_{q^2} = \mathbb{F}_q(u)$ where $u^2 + 1 = 0$, and further let $\mathbb{F}_{q^3} = \mathbb{F}_q(v)$ where $v^3 + 2 = 0$. For $P_1 = (15, 50) \in E(\mathbb{F}_q)$, we have $\pi_q(P_1) = (15^q, 50^q) = (15, 50)$. For $P_2 = (2u + 16, 30u + 39)$, we have $\pi_q(P_2) = ((2u + 16)^q, (30u + 39)^q) = (65u + 16, 39 + 37u)$; it is easy to see in this example that computing $\pi_q(Q)$ for any $Q \in E(\mathbb{F}_{q^2})$ involves a simple "complex conjugation" on each coordinate, which also agrees with $\pi_q^2(Q) = Q$. Let $P_3 = (15v^2 + 4v + 8, 44v^2 + 30v + 21)$, $\pi_q(P_3) = (33v^2+14v+8, 3v^2+38v+21)$, $\pi_q^2(P_3) = (19v^2+49v+8, 20v^2+66v+21)$, and $\pi_q^3(P_3) = P_3$.

We can now return to sketch the derivation of Equation (2.6) by skimming over results that are presented in full in Silverman's book [Sil09, Ch. V, Th. 1.1]. We now know that $P \in E(\mathbb{F}_q)$ if and only if $\pi(P) = P$ (i.e. $([1] - \pi)P = \mathcal{O}$), and thus $\#E(\mathbb{F}_q) = \#\mathrm{ker}([1] - \pi)$. It is not too hard to show that the map

$[1]-\pi$ is separable, which means that $\#E(\mathbb{F}_q) = \#\ker([1]-\pi) = \deg([1]-\pi)$. We can then make use of (a special case of) a version of the Cauchy-Schwarz inequality [Sil09][Ch. V, Lemma 1.2], to give $|\deg([1]-\pi)-\deg([1])-\deg(\pi)| \leq 2\sqrt{\deg([1])\deg(\pi)}$, from which Equation (2.6) follows from $\deg(\pi)=q$.

The theory of elliptic curves makes constant use of the *endomorphism ring* of E, denoted $\mathrm{End}(E)$, which (as the name suggests) is the ring of all maps from E to itself; addition in the ring is natural, i.e. $(\psi_1+\psi_2)(P) = \psi_1(P)+\psi_2(P)$, and multiplication in $\mathrm{End}(E)$ is composition $(\psi_1\psi_2)(P) = \psi_1(\psi_2(P))$. The *multiplication-by-m* map $[m]$ is trivially in $\mathrm{End}(E)$ for all $m \in \mathbb{Z}$, and when E is defined over a finite field, then clearly π is too, so we are usually interested in any extra endomorphisms that shed more light on the behaviour of E.

Example 2.2.6 (Magma script). Consider $E/\mathbb{F}_q : y^2 = x^3+b$. The map ξ, defined by $\xi : (x,y) \mapsto (\xi_3 x, y)$ with $\xi_3^3 = 1$ and $\xi_3 \neq 1$, is a non-trivial endomorphism on E, so $\xi \in \mathrm{End}(E)$. If $\xi_3 \in \mathbb{F}_q$, then ξ will be defined over $\mathbb{F}_q$, otherwise $\xi_3 \in \mathbb{F}_{q^2}$ in which case ξ is not *defined over* $\mathbb{F}_q$, but over $\mathbb{F}_{q^2}$. We will observe both cases. Firstly, cubic roots of unity will be defined in $\mathbb{F}_q$ if and only if $q \equiv 1 \bmod 3$, so let us take $q \equiv 19$, $b = 5$, which gives $E/\mathbb{F}_{19} : y^2 = x^3+5$. Let $\xi_3 = 7$ so that $\xi_3^3 = 1$ (we could have also taken $\xi_3^2 = 11$), so that $\xi : (x,y) \mapsto (7x,y)$ is an endomorphism on E. Applying this to, say $P = (-1,2)$, gives $\xi(P) = (-7,2) \in E$. Taking the same curve over $\mathbb{F}_{23}$, i.e. $E/\mathbb{F}_{23} : y^2 = x^3+5$, for which $P = (-1,2)$ is a again a point, we no longer have a non-trivial $\xi_3 \in \mathbb{F}_{23}$, so we must form a quadratic extension $\mathbb{F}_{q^2}(u)$, $u^2+1=0$. Now, we can take $\xi_3 = 8u+11$ (the other option is $\xi_3^2 = 15u+11$), so that $\xi(P) = (-(8u+11),2) = (15u+12,2) \in E(\mathbb{F}_{q^2})$. Notice that P started in $E(\mathbb{F}_q)$, but landed in $E(\mathbb{F}_{q^2})$ under ξ. The endomorphism ξ has an inverse ξ^{-1} (which is defined the same way but with ξ_3^2 instead), so ξ is actually an automorphism of E, written as $\xi \in \mathrm{Aut}(E)$.

The definition of $\xi : (x,y) \mapsto (\xi_3 x, y)$ in the above example gives an endomorphism on $E : y^2 = x^3+b$ regardless of the field that E is defined over. If there exists a non-trivial map (like ξ) for an elliptic curve E, we say E has *complex multiplication*. To be more precise, all elliptic curve endomorphism rings trivially contain $\mathbb{Z}$, since every $m \in \mathbb{Z}$ corresponds to the multiplication-by-m map $[m] \in \mathrm{End}(E)$. However, if non-trivial endomorphisms exist that make $\mathrm{End}(E)$ strictly larger than $\mathbb{Z}$, then we say E has complex multiplication (CM). Thus, by this definition, every elliptic curve defined over $\mathbb{F}_q$ has CM, because the existence of the Frobenius endomorphism $\pi \in \mathrm{End}(E)$ makes $\mathrm{End}(E)$ larger than $\mathbb{Z}$.

However, if we discuss whether E has CM without yet stipulating the underlying finite field, then the question becomes non-trivial in general, because the answer depends on the existence of non-trivial maps. We use Silverman's example to illustrate [Sil09, Ch. 3, Eg. 4.4].

Example 2.2.7 (Magma script). Consider $E/K : y^2 = x^3 + ax$. The map $\zeta : (x, y) \mapsto (-x, iy)$, where $i^2 = -1$ in K is an endomorphism, so E has CM. Clearly, ζ will be defined over K if and only if $i \in K$. Observe that $\zeta \circ \zeta(x, y) = \zeta(-x, iy) = (x, -y) = -(x, y)$, so $\zeta \circ \zeta = [-1]$ (i.e. ζ^2 is equivalent to negation). Thus, there is a ring homomorphism $\mathbb{Z}[i] \longrightarrow \text{End}(E)$ defined by $m + ni \mapsto [m] + [n] \circ \zeta$. If $\text{Char}(K) \neq 0$, then this map is an isomorphism, thus $\text{End}(E) \cong \mathbb{Z}[i]$, and $\text{Aut}(E) \cong \mathbb{Z}[i]^*$.

The trace of Frobenius t in Equation (2.6) is named so because of the role it plays in the characteristic polynomial satisfied by π, which is given as

$$\pi^2 - [t] \circ \pi + [q] = 0 \qquad \text{in End}(E), \tag{2.8}$$

meaning that for all $(x, y) \in E(\overline{\mathbb{F}}_q)$, we have

$$(x^{q^2}, y^{q^2}) - [t](x^q, y^q) + [q](x, y) = \mathcal{O}. \tag{2.9}$$

Example 2.2.8 (Magma script). We use our results from Example 2.2.5 to illustrate, so as before $E/\mathbb{F}_{67} : y^2 = x^3 + 4x + 3$, $\mathbb{F}_{q^2} = \mathbb{F}_q(u)$ where $u^2 + 1 = 0$, and $\mathbb{F}_{q^3} = \mathbb{F}_q(v)$ where $v^3 + 2 = 0$. The trace of Frobenius is $t = -11$, so $\#E(\mathbb{F}_q) = q + 1 - t = 79$. For $P_1 = (15, 50) \in E(\mathbb{F}_q)$, we trivially had $\pi^2(P_1) = \pi(P_1) = P_1$, so $P_1 - [t]P_1 + [q]P_1 = ([1] - [t] + [q])P_1 = [\#E(\mathbb{F}_q)]P_1 = \mathcal{O}$. For $P_2 = (2u+16, 30u+39)$, we had $\pi^2(P_2) = P_2$ and $\pi(P_2) = (65u+16, 37u+39)$, so we are computing $P_2 - [-11]\pi(P_2) + [67]P_2 = [68](2u+16, 30u+39) + [11](65u+16, 37u+39)$, which is indeed $\mathcal{O}$. $P_3 \in E(\mathbb{F}_{q^3})$ is the only case where both π and π^2 act non-trivially, so we compute $(19v^2+49v+8, 20v^2+66v+21) - [-11](33v^2+14v+8, 3v^2+38v+21) + [67](15v^2+4v+8, 44v^2+30v+21)$, which is $\mathcal{O}$.

We now give a brief sketch of Schoof's algorithm for counting points on elliptic curves [Sch85]. Understanding the algorithm is not a prerequisite for understanding pairings, but it certainly warrants mention in any overview text on elliptic curves in cryptography, since it is essentially the algorithm that made ECC practical. Before Schoof's polynomial-time algorithm, all algorithms for point counting on elliptic curves were exponential and therefore cryptographically impractical. Besides, to sketch his idea, we need to introduce the notion of *division polynomials*, which are a useful tool in general. Put simply, division polynomials are polynomials whose roots reveal torsion points: namely, for odd[3] ℓ, the ℓ-th division polynomial $\psi_\ell(x)$ on E solves to give the x-coordinates of the points of order ℓ. They are defined recursively and depend on the curve constants a and b, but rather than giving the recursions here, we point the reader to [Sil09, Ch. III, Exer. 3.7], and opt instead for an example that illustrates their usefulness.

Example 2.2.9 (Magma script). Recall the curve $E/\mathbb{F}_{101} : y^2 = x^3 + x + 1$ from Example 2.2.1 with group order $\#E(\mathbb{F}_q) = 105 = 3 \cdot 5 \cdot 7$. The x-coordinates of the points of order 2 are found as the roots of $\psi_2(x) = 4x^3 + 4x + 4$, which is irreducible in $\mathbb{F}_q[x]$, so there are no 2-torsion points in $E(\mathbb{F}_q)$. For $r = 3$, $\psi_3(x) = 3x^4+6x^2+12x+100 \in \mathbb{F}_q[x]$ factors into $\psi_3(x) = (x+73)(x+84)(x^2+45x+36)$, so we get two solutions over $\mathbb{F}_q$, namely $x = 17$ and $x = 28$. This does not mean that the points implied by both solutions are in $\mathbb{F}_q$: namely, $x = 28$ gives $x^3 + x + 1 \in \text{QR}(q)$, so two points in the 3-torsion follow as $(28, 8)$ and $(28, 93)$. Conversely, $x = 17$ gives $x^3+x+1 \notin \text{QR}(q)$, so the two points implied by $x = 17$ will be defined over $\mathbb{F}_{q^2}$. For $\psi_5(x) = 5x^{12} + ... + 16$, the factorisation in $\mathbb{F}_q[x]$ is $\psi_5(x) = (x + 15)(x + 55)(x^5 + ... + 1)(x^5 + ... + 100)$, which gives $x = 46$ and $x = 86$ as solutions. This time, both x values give rise to two points, giving four non-trivial 5-torsion points in total: $(46, 25)$, $(46, 76)$, $(86, 34)$, $(86, 67)$. $\psi_7(x)$ is degree 24, and gives three linear factors in $\mathbb{F}_q[x]$, all of which result in two 7-torsion points, giving 6 non-trivial torsion points in total: $(72, 5)$, $(72, 96)$, $(57, 57)$, $(57, 44)$, $(3, 43)$, $(3, 58)$. Other division polynomials have roots in $\mathbb{F}_q$, but these roots will not give rise to points defined over $\mathbb{F}_q$. For example, $\psi_{11}(x)$ has 5 roots over $\mathbb{F}_q$ (13, 18, 19, 22, 63), but none of them give points in $E(\mathbb{F}_q)$, meaning we will have to extend to $E(\mathbb{F}_{q^2})$ to collect any 11-torsion points. The only division polynomials whose roots produce points defined over $\mathbb{F}_q$ are the $\psi_d(x)$ with $d \mid 105$. This generalises to imply that the only division polynomials whose roots produce points defined over $\mathbb{F}_{q^n}$ are $\psi_d(x)$, where $d \mid \#E(\mathbb{F}_{q^n})$.

We are now in a position to shed light on Schoof's algorithm. Equation (2.6) means that computing $E(\mathbb{F}_q)$ immediately reduces to computing the (much smaller) trace of Frobenius, t. At the highest level, Schoof's idea is to compute

[3] When ℓ is even, the division polynomial is of the form $\psi_\ell(x, y) = y \cdot \tilde{\psi}_\ell(x)$ since $y = 0$ gives points of order two, which are in the ℓ-torsion.

$t_\ell \equiv t \bmod \ell$ for enough co-prime ℓ's to be able to uniquely determine t within the interval $-2\sqrt{q} \leq t \leq 2\sqrt{q}$ via the Chinese Remainder Theorem. Namely, when $\prod_\ell t_\ell \geq 4\sqrt{q}$, then we have enough relations to determine the correct t. To compute t_ℓ for various primes ℓ, Schoof looked to consider Equation (2.9) "modulo ℓ", restricting the points (x, y) to come from the ℓ-torsion, and trying to solve

$$(x^{q^2}, y^{q^2}) - [t_\ell](x^q, y^q) + [q_\ell](x, y) = \mathcal{O}, \tag{2.10}$$

for t_ℓ, where $q_\ell \equiv q \bmod \ell$. The problem for general ℓ is, that since we do not know the group order, we cannot explicitly use ℓ-torsion points in (2.10), nor do we know if they are even defined over $\mathbb{F}_q$, or where they *are* defined, so we have to work with (2.10) implicitly. Namely, we restrict (2.10) to the ℓ-torsion by working modulo $\psi_\ell(x)$: we do not work with Equation (2.10) on $E(\mathbb{F}_q)$, but rather in the polynomial ring $R_\ell = \mathbb{F}_q[x, y]/\langle \psi_\ell(x), y^2 - (x^3 + ax + b)\rangle$, where the size of the polynomials $f(x, y)$ we deal with in R_ℓ are bounded by the degrees of the division polynomials $\psi_\ell(x)$. Even for very large prime fields $\mathbb{F}_q$ of cryptographic size, the number of different primes used is small enough to keep this algorithm very practical. For example, finding the group order of the curve defined over a 256-bit prime q in Example 2.2.3 would require solving (2.10) for the 27 primes up to $\ell = 107$, at which point the product of all the primes used exceeds $4\sqrt{q}$. It is not too difficult to deduce that the asymptotic complexity of Schoof's algorithm is $O\left((\log q)^8\right)$ (see [Sil09, Ch. XI.3] for details, and further improvements).

Example 2.2.10 (Magma script). Consider $E/\mathbb{F}_{13} : y^2 = x^3 + 2x + 1$; we seek $\#E(\mathbb{F}_{13})$. Schoof's algorithm actually begins with $\ell = 3$ [Sil09, Ch. XI.3]; so since $14 < 4\sqrt{13} < 15$, we only need to solve (2.10) with $\ell = 3$ and $\ell = 5$. For $\ell = 3$, $\psi_3(x) = 3x^4 + 12x^2 + 12x + 9$, so we work in the ring $R_3 = \mathbb{F}_q[x, y]/\langle 3x^4 + 12x^2 + 12x + 9, y^2 - (x^3 + 2x + 1)\rangle$ with $q_\ell = 1$, to find that $t_3 = 0$. For $\ell = 5$, $\psi_5(x) = 5x^{12} + ... + 6x + 7$, so we work in the ring $R_5 = \mathbb{F}_q[x, y]/\langle 5x^{12} + ... + 6x + 7, y^2 - (x^3 + 2x + 1)\rangle$ with $q_\ell = 3$ to find that $t_5 = 1$. For both cases we had to compute $[q_\ell](x, y)$ in R_ℓ using the affine formulas (2.4) and (2.5), compute (x^q, y^q) and (x^{q^2}, y^{q^2}) in R_ℓ, and then test incremental values of t_ℓ until $[t_\ell](x^q, y^q)$ (also computed with the affine formulas) satisfies (2.10). The CRT with $t \equiv 0 \bmod 3$ and $t \equiv 1 \bmod 5$ gives $t \equiv 6 \bmod 15$, which combined with $-7 \leq t \leq 7$ means $t = 6$, giving $\#E = q + 1 - t = 8$.

We finish this chapter by briefly discussing one more improvement to ECC that will essentially bring the reader up to speed with major milestones that contribute to the current state-of-the-art implementations. The technique was introduced by Gallant, Lambert and Vanstone (GLV) [GLV01], and recently generalised by Galbraith, Lin and Scott (GLS) [GLS11]. It exploits the existence of an efficiently computable endomorphism ψ that allows us to instantly move P to a large multiple $\psi(P) = [\lambda]P$ of itself, so that (in the simplest case) the scalar multiplication $[m]P$ can be split into $[m]P = [m_0]P + [m_1]\psi(P)$, where if $|m| \approx r$ (the large subgroup order), then $|m_0|$, $|m_1| \approx \sqrt{r}$. The values m_0 and m_1 are found by solving a closest vector problem in a lattice [GLV01, §4]. We apply an example from the GLV paper (which was itself taken from Cohen's book [Coh96, §7.2.3]) that is actually exploiting a special case of the endomorphism we described in Example 2.2.7.

Example 2.2.11 (Magma script). Let $q \equiv 1 \bmod 4$ be prime, $E/\mathbb{F}_q : y^2 = x^3 + ax$, and let $i^2 = -1$. The map defined by $\psi : (x, y) \mapsto (-x, iy)$ and $\psi : \mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_q$ ($\psi = \zeta$ from 2.2.7). Let $P \in E(\mathbb{F}_q)$ have prime order r, then $\psi(Q) = [\lambda]Q$ for all $Q \in \langle P \rangle$, and λ is the integer satisfying $\lambda^2 = -1 \bmod r$. We give a specific example: $q = 1048589$, $E/\mathbb{F}_q : y^2 = x^3 + 2x$ with $\#E = 2r$, where $r = 524053$; we further have $i = 38993$, and $\lambda = 304425$. $P = (609782, 274272) \in E$ has $|\langle P \rangle| = r$, so we can take any element in $\langle P \rangle$, say $Q = (447259, 319154)$, and compute $\psi(Q) = (-447259, i \cdot 319154) = (601330, 117670) = [304425](447259, 319154) = [\lambda]Q$. Computing a random multiple of Q, say $[m]Q$ with $m = 103803$, can be done by decomposing m into (in this case) $(m_0, m_1) = (509, 262)$, and instead computing $[m]Q = [m_0]Q + [m_1]\psi(Q)$. Here m is 17 bits, whilst m_0 and m_1 are both 9 bits. Doing the scalar multiples $[m_0]Q$ and $[m_1]\psi(Q)$ separately would therefore give no savings, but where the GLV/GLS methods gain a substantial speed-up is in merging the doublings required in both of the multiplications by the "mini-scalars", which halves the number of doublings required overall; again, see [GLV01, GLS11] for futher details.

2.3 Chapter summary

We defined the elliptic curve group law $\oplus$ via the chord-and-tangent method, and discussed that elliptic curve groups are an attractive setting for discrete-log based cryptosystems because of the relative security obtained for the sizes of the fields they are defined over. We also exemplified many improvements in the context of cryptographic implementations, where the fundamental operation (that creates ECDLP instances) is computing large scalar multiples $[m]P$ of $P \in E$. Namely, we showed that group law computations in finite fields can be much faster in projective coordinates, i.e. computing $(X_1 : Y_1 : Z_1) \oplus (X_2 : Y_2 : Z_2)$ rather than $(x_1, y_1) \oplus (x_2, y_2)$, and that other (non-Weierstrass) curve models also offer advantages. We gave an explicit equation for the number of points in $E(\mathbb{F}_q)$, and briefly discussed Schoof's polynomial-time algorithm that facilitates point counting on curves of cryptographic size. We also introduced the notion of the endomorphism ring $\text{End}(E)$ of E, and finished by showing that non-trivial elements of $\text{End}(E)$ can be used to further accelerate ECC. A reader that is comfortable with the exposition in this chapter is equipped with many of the tools required to tackle the vast literature in this field, and is somewhat up-to-date with the state-of-the-art ECC implementations. For example, in the context of chasing ECC speed records, some authors have applied alternative projective coordinate systems to the Edwards model to give very fast scalar multiplications [HWCD08], whilst others have investigated higher dimension GLV/GLS techniques (Example 2.2.11 above was 2-dimensional) to gain big speed-ups [HLX12]; visit `http://bench.cr.yp.to/supercop.html` for comprehensive and up-to-date benchmarkings of a wide number of implementations that are pushing ECC primitives to the limit.

Relaxed notation. Our last order of business before proceeding into the next chapter is to relax some notation in order to agree with the rest of the literature. Rather than writing "$\oplus$" for the elliptic curve group law, from hereon we simply use "$+$". Similarly, for the inverse of the point P, we use $-P$ instead of $\ominus P$.