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 $\mathrm{End}(E)$ of E, and finished by showing that non-trivial elements of $\mathrm{End}(E)$ 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 $\ominus P$.