Chapter 8

Summary

The fundamental computation in ECC is the scalar multiplication which, in the most straightforward case, computes $[m]P$ from $m \in \mathbb{Z}$ and $P \in E$ via a double-and-add routine. Computing the Miller loop in the Tate pairing $e(P, Q)$ can be thought of as an extension of this computation by stipulating that the line functions used in the scalar multiplication of P are evaluated at Q and accumulated as we proceed to compute $[m]P$. Thus, those who understand ECC related computations should find a relatively easy transition to the basics of pairing computation. This is why we started with a general overview of ECC in Chapter 2, which included an elementary description of the group law, as well as many optimisations like that of adopting projective coordinates or the GLV technique which exploits endomorphisms to accelerate the computation of $[m]P$. Carrying many ECC related improvements over to the context of PBC is straightforward, whilst translating other optimisations requires a firm knowledge of the functions involved in the pairing computation. For example, one could not hope to thoroughly understand how or why the (optimal) ate pairing works without knowing the basics of divisor theory. In Chapter 3 we presented all the divisor theory that is necessary in preparation for the description of the Weil, Tate and ate-like pairings. We gave a very detailed description of the r-torsion group on E in Chapter 4, and illustrated that the availability of different (efficiently computable) maps between order r subgroups give rise to different pairing types. We adopted the widely accepted argument that Type 3 pairings are most commonly the preferred setting, thereby defining $\mathbb{G}_1$ and $\mathbb{G}_2$ as the base field subgroup and trace-zero subgroup respectively. We finished that chapter by detailing an efficient method of working in $\mathbb{G}_2$, namely by exploiting the isomorphism between the trace-zero subgroup $\mathbb{G}_2$ on E and the trace-zero subgroup $\mathbb{G}_2'$ on the twisted curve E', which is defined over a smaller field. In Chapter 5 we defined the Weil and Tate pairings and described Miller's algorithm which makes cryptographic pairing computations practical. Having described an efficient algorithm to compute pairings, Chapter 6 looked at the complementary arena of generating pairing-friendly curves. We discussed that pairing-friendly curves are very special in general, and cannot be found by searching at random, before giving a general overview of the many clever methods that have been developed in the last decade to facilitate their construction. We finished in Chapter 7 by bringing the reader up to speed with some of the major milestones in efficient pairing computation, most notably the BKLS-GHS algorithm for the Tate pairing, and the impressive work on loop shortened versions of the Tate pairing which was pinnacled by the optimal ate pairing.