Arithmetic geometry and key exchange : compact Diffie–Hellman with efficient endomorphisms

Benjamin Smith
INRIA and École polytechnique, France
smith@lix.polytechnique.fr

Diffie–Hellman key exchange is one of the fundamental primitives in public-key cryptography. If G is an abelian group (written additively), then the Diffie–Hellman protocol in G is composed of four computations in the form $P \longmapsto [m]P = P + \cdots + P$ (m times) for various points P and integers m; optimising this scalar multiplication operation is crucial.

In practice, the most efficient contemporary Diffie–Hellman implementations are based on elliptic curves, or Jacobians of genus 2 curves. But in these groups, computing $-P$ is extremely efficient, so we can use the fact that $[m](\pm P) = \pm([m]P)$ to simplify and speed up the protocol, identifying P with $-P$ (formally, working in the quotient set $G/\langle\pm 1\rangle$). These "compact" systems offer significant savings in both space, which translates into slightly shorter keys, and in computing time, through simpler pseudo-group law formulae. In the elliptic curve context, this amounts to using only x-coordinates of points and Montgomery's pseudo-group law. Bernstein's Curve25519 software, which has become a de facto reference implementation of Diffie–Hellman at the 128-bit security level, is a practical example of these techniques in practice. The genus 2 analogue is Kummer surface arithmetic, where we can use particularly efficient formulae developed by the Chudnovskys, and popularized in cryptography by Gaudry.

Recent years have seen renewed interest in the Gallant–Lambert–Vanstone (GLV) technique for computing $[m]P$ in G. Here, we suppose our elliptic curve (or our genus 2 Jacobian) has an efficiently computable non-integer endomorphism ϕ, which when applied to elements of G acts like $[\lambda]$ (for some large eigenvalue λ). Suppose we want to compute $[m]P$: first we use the Euclidean algorithm to compute much smaller integers a and b such that $a + b\lambda \equiv m \pmod{\#G}$, and then we compute $[m]P = [a]P + [b]\phi(P)$. The running time of the multiexponentiation depends on the size of a and b, while traditional scalar multiplication depends on the size of m. In practice, a and b have half the bitlength of m, which means that GLV and its variants can offer us a significant speedup.

In this talk, we will discuss the adaptation of GLV techniques to x-coordinate-only and Kummer surface systems. On the practical side, we will present recent experimental results for a new elliptic-curve based implementation. On the more theoretical side, we will present some new formulae for Kummer surface systems.