The HECTOR BAT

Peter Birkner and Peter Schwabe

Department of Mathematics and Computer Science,
Eindhoven University of Technology, The Netherlands

HECTOR (Hyperelliptic Curve with Two-Rank One) is an example implementation of the Diffie-Hellman key exchange protocol using a hyperelliptic curve of genus 2 over a finite field of characteristic 2. The implementation consists of the following parts:

Choice of Parameters

We use the hyperelliptic curve $C : y^2 + xy = x^5 + t^{55}x^3 + x^2 + t^{53}$ over the field $\mathbb{F}_2[t]/(t^{113} + t^9 + 1)$ which was generated by Wouter Castryck, Katholieke Universiteit Leuven, Belgium. The choice of the finite field takes into account three aspects:

Firstly, it allows for an order of the divisor class group of appropriate size for the desired security level.

Secondly, the extension degree of $\mathbb{F}_2$ was chosen to be prime to make a Weil descent attack impossible. We point out that we explicitly avoid Mersenne and Fermat primes as extension degrees here. When performing the Weil descent using those primes, the transfer can lead to a probably easier problem, for instance when the genus of curve in the Weil restriction is minimal. A Mersenne prime is a prime number of the form $2^n - 1$. In the cryptographically important range the Mersenne primes are $3, 7, 31$ and 127. A Fermat prime is a prime number of the form $2^{2^n} + 1$. In the cryptographically important range $3, 5, 17$ and 257 are Fermat primes. For more details on this see page 533 in [1] and [5].

As a third aspect we considered the unused bits when representing field elements as arrays of long integers, on both, 32 and 64 bit architectures this number is 15.

As for the choice of the hyperelliptic curve we have chosen a genus two curve of 2-rank one. The 2-rank of a genus two curve can be $0, 1$ or 2. The 2-rank equals the degree of h and since we are in the genus two case, we have $0 \leq \deg(h) \leq 2$. We a priori exclude curves of 2-rank zero since they are supersingular as the polynomial h is constant. Supersingular curves are weak under the Frey-Rück attack [2]. We prefer curves of 2-rank one over those of 2-rank two because there exist faster arithmetic in the divisor class group, more precisely the addition and doubling of divisor classes takes less operations in the underlying field. For explicit formulas see Section 14.5 in [1] and [4].

The Finite Field Implementation

The implementation of the finite field is based on the $\mathtt{mp}\mathbb{F}_q$ library [3]. More than just a library, $\mathtt{mp}\mathbb{F}_q$ is a code generator for finite field arithmetic. Algorithms for reduction can hence be optimised for just one special finite field. The library makes extensive use of the SSE2 processor extensions, so HECTOR will only run on machines, where this extension is available.

The Arithmetic on the Curve

The arithmetic is chiefly based on three important points: (1) the method of computing a multiple of a divisor class, (2) the addition and doubling formulas and (3) the decision between affine and projective coordinates.

The method we use to compute a scalar multiple of a divisor class is wNAF, i.e. we use a signed bit representation and pre-computed windows. The addition and doubling formulas are the fasted yet known ones for our specific situation (cf. [1] and [4]).

As for the choice of the coordinate system we need to mention that this heavily depends on the machine. The important number to look at in this context is the I/M-ratio. The question is how expensive a field inversion compared to a field multiplication is. If this ratio is rather low, then one might want to use affine coordinates using 1 field inversion but only a relatively small amount of multiplications. If, on the other hand, one inversion is expensive, one might want to avoid inversions completely at the cost of more multiplications. In this case we recommend using inversion-free projective coordinates.

Group Exponentiation

For the group exponentiation we use two different algorithms. To compute multiples of the generator we use the 2-table comb method with 512 precomputed multiples in total. For the computation of multiples of points other than the generator we use the windowed NAF method with a window size of 5.

Performance Evaluation

References

[1] Roberto M. Avanzi, Henri Cohen, Christophe Doche, Gerhard Frey, Tanja Lange, Kim Nguyen and Frederik Vercauteren: Handbook of Elliptic and Hyperelliptic Curve Cryptography. Chapman & Hall/CRC, 2005.

[2] Gerhard Frey and Hans-Georg Rück: A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves. Mathematics of Computation, vol. 62, pp. 865–874, 1994.

[3] Pierrik Gaudry, Emmanuel Thomé: The $\mathtt{mp}\mathbb{F}_q$ library and implementing curve-based key exchanges. Proceedings of SPEED workshop (Amsterdam), 2007.

[4] Tanja Lange: Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. Applicable Algebra in Engineering, Communication and Computing, vol. 15, no. 5, pp. 295–328, 2005.

[5] Alfred J. Menezes and Minghua Qu: Analysis of the Weil descent attack of Gaudry, Hess and Smart. Topics in Cryptology – CT-RSA 2001, Lecture Notes in Computer Science, vol. 2020, pp. 308–318, Springer-Verlag, 2001.