Efficient Arithmetic on Binary Hyperelliptic Curves

Peter Birkner, Department of Mathematics, Technical University of Denmark

Abstract

Since hyperelliptic curve cryptosystems (HECC) gain similar attention as their elliptic counterparts, it is very interesting to investigate, whether ideas and methods can be transferred from the elliptic to the hyperelliptic case. The most important operation used by elliptic curves cryptosystems (ECC) is scalar multiplication which is composed of point addition, doubling and sometimes halving. These operations are well investigated and it is likely that the present formulae are the most efficient ones. For HECC explicit formulae for addition, doubling and hence scalar multiplication of divisor classes are also known [1,4]. In addition to that we present an efficient halving algorithm for divisor classes.

Basic Notions

- **Hyperelliptic Curve**

  Let K be a field and let $\overline{K}$ be the algebraic closure of K. A curve C, given by an equation of the form

  $$C: y^2 + h(x)y = f(x), \qquad (1)$$

  where $h \in K[x]$ is a polynomial of degree at most g and $f \in K[x]$ is a monic polynomial of degree $2g+1$, is called a *hyperelliptic curve of genus g over K* if no point on the curve over $\overline{K}$ satisfies both partial derivatives $2y + h = 0$ and $f' - h'y = 0$.

  The last condition ensures that the curve is nonsingular. In our case we concentrate on hyperelliptic curves of genus 2 over finite fields of characteristic 2. In this case we need a non-zero polynomial h in the curve equation.

- **Divisor Class Group**

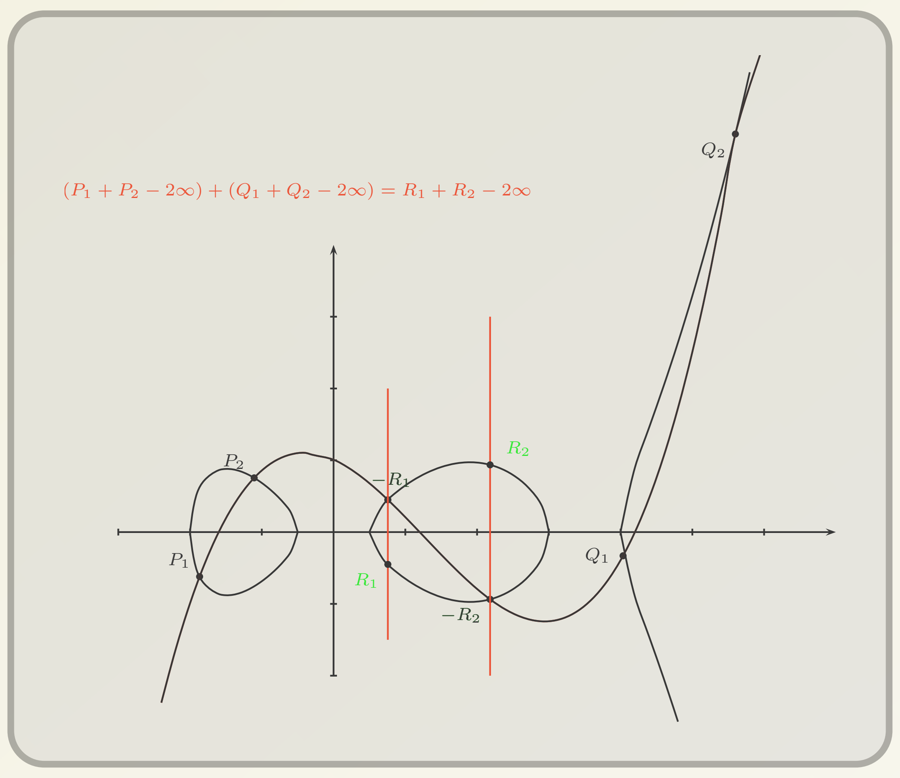
  Let C be a hyperelliptic curve of genus g over a field K. The group of degree zero divisors of C is denoted by Div_C^0. The quotient group of Div_C^0 by the group of principal divisors of C is called the *divisor class group of C* and is denoted by Pic_C^0. It is also called the *Picard group of C*.

  For a pictorial description see below.

How to perform the group operation in the divisor class group?

- Use Cantors algorithm (see picture) to implement the addtion of divisor classes.
- **Disadvantage**: Cantor is to slow for efficient implementations!
- **Solution**: Consider additions and doublings separately to make them faster

This is a graphical description how to perform the addtion of two divisor classes (each represented by two affine points on the hyperelliptic curve) using Cantors general algorithm.



Scalar multiplication is the most important operation in DL based cryptosystems!

That operation is most often implemented using algorithms like Double-and-Add or windowing methods.

Implementations that use a Double-and-Add algorithm need a fast double operation!

Algorithm 1 Double-and-Add

INPUT: An element x of G and an integer $n = (n_{l-1} \dots n_0)_2$

OUTPUT: The element $n \cdot x$ in G

$y \leftarrow 1$ and $i \leftarrow l-1$
while $i \geq 0$
  $y \leftarrow 2 \cdot y$
  if $n_i = 1$ **then** $y \leftarrow y + x$
  $i \leftarrow i - 1$
return y

Doubling of Divisor Classes (affine)

Let $\overline{E} = [x^2 + u_1'x + u_0', v_1'x + v_0']$ be a divisor class. We can compute the doubled divisor class $\overline{D} = 2\overline{E} = [x^2 + u_1x + u_0, v_1x + v_0]$ using Lange and Steven's explicit formulae (see [4]):

$$u_1 = \left(\frac{{u_0'}^2}{f_0 + {v_0'}^2}\right)^2,$$

$$u_0 = \left(\left({u_1'}^2 + f_3\right)\left(\frac{{u_0'}^2}{f_0 + {v_0'}^2}\right) + u_1'\right)^2 + \left(\frac{{u_0'}^2}{f_0 + {v_0'}^2}\right),$$

$$v_0 = \left(\frac{{u_0'}^2}{f_0 + {v_0'}^2} + {u_1'}^2 + f_3\right)u_0 + {u_0'}^2,$$

$$v_1 = \left(\frac{{u_0'}^2}{f_0 + {v_0'}^2} + {u_1'}^2 + f_3\right)\left({u_1'}^2 + f_3\right)\left(\frac{{u_0'}^2}{f_0 + {v_0'}^2}\right) + \left(\frac{{u_0'}^2}{f_0 + {v_0'}^2}\right)u_1 + f_2 + {v_1'}^2.$$

Algorithm 1 Divisor Class Halving (HLV)

INPUT: Divisor class $\overline{D} = [u, v]$, where $u = x^2 + u_1x + u_0$, $v = v_1x + v_0$ and the pre-computed values $f_3^2, \sqrt{f_0}$

OUTPUT: Halved divisor class $\overline{E} = [u', v']$ such that $\overline{D} = 2\overline{E}$

$q_1 \leftarrow \sqrt{u_1},\ q_2 \leftarrow 1/q_1,\ q_3 \leftarrow q_2^2,\ q_4 \leftarrow u_0q_3,\ q_5 \leftarrow \sqrt{q_2}$ ▷ 1I, 1M, 2SR, 1S
$q_6 \leftarrow \sqrt{q_4},\ c \leftarrow u_1(q_6 + q_5 + f_3)$ ▷ 1SR, 1M
$t' \leftarrow \sum_{i=0}^{(d-3)/2} c^{2^{(2i+1)}}$ ▷ 1HT
$u_1' \leftarrow t'q_2,\ t \leftarrow u_1'^2,\ s_1 \leftarrow v_0 + (q_1 + t + f_3)u_0$ ▷ 2M, 1S
$u_0' \leftarrow \sqrt{s_1},\ b \leftarrow \mathrm{Trace}(u_1'(u_0' + t + f_3))$ ▷ 1M, 1S, 1TR
if $b = 0$ **then**
  $v_0' \leftarrow q_5u_0' + \sqrt{f_0}$ ▷ 1M
else
  $t \leftarrow t + q_3,\ u_1' \leftarrow u_1' + q_2$
  $u_0' \leftarrow u_0' + q_6,\ v_0' \leftarrow q_5u_0' + \sqrt{f_0}$ ▷ 1M
end if
$v_1' \leftarrow \sqrt{v_1 + q_1\big((q_1 + t + f_3)(t + f_3) + u_1\big) + f_2}$ ▷ 2M, 1SR
return $[x^2 + u_1'x + u_0',\ v_1'x + v_0']$ ▷ Total: 1I, 8M, 4SR, 3S, 1HT, 1TR

For hardware implementations one needs inversionsfree doubling formulae!

Solution: Use projective or recent coordinates instead of affine coordinates

Advantage: This allows fast and inversionsfree doubling of divisor classes!

Projective Coordinates

In projective coordinates a divisor class of the Jacobian of a HEC is written as $[U_1, U_0, V_1, V_0, Z]$ which represents the affine divisor class $[x^2 + U_1/Zx + U_0/Z, V_1/Zx + V_0/Z]$.

Recent Coordinates

In Recent coordinates a divisor class of the Jacobian of a HEC is written as $[U_1, U_0, V_1, V_0, Z]$ which represents the affine divisor class $[x^2 + U_1/Zx + U_0/Z, V_1/Z^2x + V_0/Z^2]$. These are so called weighted coordinates. The variables U_i have weight 1 and the V_i have weight 2.

Algorithm 1 Divisor Class Doubling in Recent Coordinates

INPUT: Divisor class $\overline{D} = [u, v, z]$, where $u = x^2 + u_1x + u_0$, $v = v_1x + v_0$

OUTPUT: Doubled divisor class $\overline{E} = [u', v', z']$ such that $\overline{E} = 2\overline{D}$

$z_2 \leftarrow z^2,\ z_4 \leftarrow z_2^2,\ t1 \leftarrow f_0z_4 + v_0^2$
$t_2 \leftarrow u_1^2 + f_3z_2$
$a_1 \leftarrow u_0^2,\ a_2 \leftarrow a_1z,\ a_3 \leftarrow a_1z_4$
$q_1 \leftarrow (t_2a_2 + u_1t_1)^2 + t_1a_3$
$q_2 \leftarrow t_1^2,\ q_3 \leftarrow q_2^2,\ q_4 \leftarrow a_1z_2$
$q_5 \leftarrow t_1t_2,\ q_6 \leftarrow (a_3 + q_5)t_1$
$u_0' \leftarrow q_1$
$u_1' \leftarrow a_3q_4$
$v_0' \leftarrow q_6q_1 + q_3q_4$
$v_1' \leftarrow q_4(q_5q_6 + a_3^2t_1) + q_3(f_2z_4 + v_1^2)$
$z' \leftarrow q_2z_2$
return $[x^2 + u_1'x + u_0',\ v_1'x + v_0']$

References

[1] Roberto Avanzi, Henri Cohen, Christophe Doche, Gerhard Frey, Tanja Lange, Kim Nguyen, and Frederik Vercauteren.The Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, 2005.

[2] Peter Birkner. Efficient Divisor Class Halving on Genus Two Curves. To appear in: Proceedings of Selected Areas in Cryptography - SAC 2006.

[3] Izuru Kitamura, Masanobu Katagi, and Tsuyoshi Takagi. A Complete Divisor Class Halving Algorithm for Hyperelliptic Curve Cryptosystems of Genus Two. In: Information Security and Privacy - ACISP 2005, Vol. 3574 of Lecture Notes in Computer Science, p. 146-157. Springer-Verlag, 2005.

[4] Tanja Lange and Marc Stevens. Efficient Doubling for Genus Two Curves over Binary Fields. In Selected Areas in Cryptography - SAC 2004, Vol. 3357 of Lecture Notes in Computer Science, p. 170-181. Springer-Verlag, 2005.