

The Real Model of a Hyperelliptic Curve

Renate Scheidler

rscheidl@math.ucalgary.ca



Centre for Information Security and Cryptography



Joint work with

Mike Jacobson (CISaC, University of Calgary) and Andreas Stein (University of Wyoming)

Research supported in part by NSERC of Canada

Hyperelliptic Curves

$$C : v^2 + h(u)v = f(u)$$

$f, h \in \mathbb{F}_q[u]$; $h = 0$ if q odd; non-singular;

● Imaginary Model

- f monic and $\deg(f) = 2g + 1$,
- $\deg(h) \leq g$ if q even;

● Real Model

- If q odd: f monic and $\deg(f) = 2g + 2$;
- If q even: h monic, $\deg(h) = g + 1$ and
 - $\deg(f) \leq 2g + 1$ or
 - $\deg(f) = 2g + 2$, $\text{sgn}(f) = e^2 + e$ ($e \in \mathbb{F}_q^*$).

g is the *genus* of C .

Notation

- $\mathbb{F}_q[C] = \mathbb{F}_q[u, v]$
coordinate ring of C (ring of regular functions, maximal order of C);
- $\mathbb{F}_q(C) = \mathbb{F}_q(u, v) = \text{Quot}(\mathbb{F}_q[C])$
function field of C ;
- $\text{Cl}(\mathbb{F}_q[C])$
ideal class group of $\mathbb{F}_q[C]$ (group of fractional $\mathbb{F}_q[C]$ -ideals modulo principal ideal equivalence);
- $\text{Jac}_q(C)$
Jacobian of C over \mathbb{F}_q (group of degree zero divisors defined over \mathbb{F}_q modulo principal divisor equivalence).

Properties

Imaginary Model

- C has one point ∞ at infinity of degree 1 that is totally ramified;
- The unit group $\mathbb{F}_q[C]^*$ of $\mathbb{F}_q[C]$ is simply \mathbb{F}_q^* ;
- $\text{Jac}_q(C)$ is isomorphic $\text{Cl}(\mathbb{F}_q[C])$;

Real Model

- C has two (opposite) points at infinity, ∞_+ and ∞_- , both of degree 1 and unramified;
- The class of the degree zero divisor $\infty_+ - \infty_-$ has finite order R , the *regulator* of C ;
- $\mathbb{F}_q[C]^* \cong \mathbb{F}_q^* \times \langle R(\infty_+ - \infty_-) \rangle$;

Degree Zero Divisors

Every degree zero divisor has a unique representation of the following form:

Imaginary Model

$$D = D_{\text{finite}} - \deg(D_{\text{finite}}) \infty$$

Real Model

$$D = D_{\text{finite}} - \deg(D_{\text{finite}}) \infty_- + v_+(D)(\infty_+ - \infty_-)$$

Reduced Divisors:

- $\deg(D) \leq g$
- $0 \leq -v_+(D) < R$ (real model only)

(Semi-)Reduced Divisors

Semi-reduced divisors: $D = (a, b)$ where

- $a, b \in \mathbb{F}_q[x]$, a monic
- a unique, $b \pmod{a}$ unique
- $a \mid f + hb - b^2$

Imaginary Model:

- Every degree zero divisor class has a unique reduced representative

Real Model:

- Every degree zero divisor class has a unique reduced representative with $0 \leq -v_+(D) \leq g - \deg(D_{\text{finite}})$

Infrastructures, Real Model

Fix any divisor class C

- Reduced divisors in C have *distinct* v_+ values
- The *infrastructure* of C is the set \mathcal{R}_C of reduced divisors in C :

$$\mathcal{R}_C = \left\{ D_1 = D, D_2, \dots, D_{|\mathcal{R}_C|} \right\}$$

- \mathcal{R}_C is an ordered set under the *distance*

$$\delta(D_i) = v_+(D - D_i)$$

- $|\mathcal{R}_C| \approx R \approx q^g$.
- *Principal* infrastructure \mathcal{R}_0 : $C = \mathbf{0}$, $D_1 = 0$,
 $\delta(D_i) = -v_+(D_i)$

Baby Steps & Distances, Real Model

A *baby step* is the operation

$$\boxed{D_i \rightarrow D_{i+1}}$$

Properties of the Distance:

$$\delta(D_1) = 0$$

$$\delta(D_{i+1}) - \delta(D_i) = g + 1 \quad \text{if } D_i = 0$$

$$1 \leq \delta(D_{i+1}) - \delta(D_i) \leq g \quad \text{if } D_i \neq 0$$

For $x \in [0, R)$, the divisor $D_i \in R_C$ below x can be defined via

$$\delta(D_i) \leq x < \delta(D_{i+1})$$

Giant Steps, Real Model

By applying *exactly* the same algorithm as used for adding divisor classes via reduced representatives in the *imaginary* model to two divisors $D', D'' \in \mathcal{R}_0$, we can define a reduced divisor

$$D = D' \oplus D'' \in \mathcal{R}_0$$

The operation $\boxed{(D', D'') \rightarrow D' \oplus D''}$ is called a *giant step*. We have

$$d(D' \oplus D'') = \delta(D') + \delta(D'') - d \text{ with } 0 \leq d \leq 2g.$$

Complexity of a giant step: $O(g^2)$ field operations.

Properties of Baby Steps

Real & Imaginary Model

- Applying at most $\lceil (\deg(a) - g)/2 \rceil$ baby steps to a semi-reduced divisor $D = (a, b)$ produces a reduced divisor.
- If D is the sum of two reduced divisors, this requires at most $\lceil g/2 \rceil$ baby steps.
- The q_i are the partial quotients of the continued fraction expansion of $(b + ev)/a$ ($e = 0$ or 1).

Real Model Only

- Baby steps move forward through \mathcal{R}_C .
- $|\mathcal{R}_C|$ baby steps applied to any divisor of \mathcal{R}_C produce all of \mathcal{R}_C (*not recommended!*)

Computing Baby Steps

Baby step $D_i = (a_{i-1}, b_{i-1}) \rightarrow D_{i+1} = (a_i, b_i)$:

$$q_{i-1} = \left[\frac{b_{i-1} + e_{i-1}v}{a_{i-1}} \right] \quad \text{where}$$

$$e_{i-1} = \begin{cases} 1 & \text{if } C \text{ is real and } \deg(a_{i-1}) \leq g + 1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_i = h + q_{i-1}a_{i-1} - b_{i-1}$$

$$a_i = \frac{f + hb_i - b_i^2}{a_{i-1}}$$

($[\cdot]$ is the polynomial part of the Laurent series in u^{-1} .)

Complexity of a baby step: $O(g)$ field operations.

Divisor Addition

If $D' = (a', b')$ and $D'' = (a'', b'')$, then

$D' + D'' = (a, b) + (s)$ where

$$s = \gcd(a', a'', b' + b'')$$

$$= Va' + Wa'' + X(b' + b'')$$

$$a = \frac{a'a''}{s^2}$$

$b = b'' + Ua''/s$ where

$$U \equiv W(b' - b'') + X \frac{f - (b'')^2}{a''} \left(\text{mod } \frac{a'}{s} \right)$$

Giant Steps à la Cantor

To obtain $D' \oplus D''$:

1. Compute $D' + D''$ using the well-known divisor addition (ideal multiplication) algorithm;
2. Apply at most $2g$ baby steps to $D' + D''$ to obtain the $\left\{ \begin{array}{c} \text{first} \\ \text{unique} \end{array} \right\}$ divisor in the $\left\{ \begin{array}{c} \text{real} \\ \text{imaginary} \end{array} \right\}$ case. This is $D' \oplus D''$.

Recall that the reduction part amounts to computing a continued fraction expansion (a_i, b_i, q_i) of an algebraic function $(b + ev)/a$ where usually $\deg(b) < \deg(a) \approx 2g$ (*double-sized operands!*)

Giant steps with NUCOMP

Suppose $D' + D'' = (a, b) + (s)$

- *Idea:* Replace the partial quotients in $(b + ev)/a$ by those in the rational function $U/(a'/s)$:

$$\frac{b + ev}{a} = \frac{U}{a'/s} + \frac{b'' + ev}{a} = \frac{U}{a'/s} + O(u^{1-g})$$

- Produces the same sequence of partial quotients as Cantor giant steps, but uses only the Euclidean Algorithm (no costly a_i, b_i).
- There are formulas for recovering the coefficients of $D' \oplus D''$ at the end (Shanks, van der Poorten).
- Still $O(g^2)$ field operations, but we only work with operands of degree $\leq g$.

Scalar Multiplication in $\text{Jac}_q(C)$

Input: a divisor D and a scalar n

Output: the reduced divisor $D \oplus D \oplus \dots \oplus D$
(n times) in the divisor class of nD

1. Write $n = 2^l + b_{l-1}2^{l-1} + \dots + b_0$ (in binary)
2. Set $E = D$
3. For $i = 1$ to l do
 - (a) // double Replace E by $E \oplus E$
 - (b) // add If $b_i = 1$, replace E by $E \oplus D$
4. Output E

Average complexity: $1.5 \log_2(n)$ giant steps

“Scalar Multiplication” in \mathcal{R}_0, I

Input: a divisor D and a “scalar” n

Output: The divisor E below $n\delta(D)$

1. Write $n = 2^l + b_{l-1}2^{l-1} + \dots + b_0$ (in binary)
2. Set $E = D$
3. For $i = 1$ to l do
 - (a) // *double* Replace E by $E \oplus E$
 - (b) // *adjust* Apply at most $2g$ baby steps to reach the divisor below $2\delta(E)$
 - (c) If $b_i = 1$, then
 - i. // *add* Replace E by $E \oplus D$
 - ii. // *adjust* Apply at most $2g$ baby steps to reach the divisor below $\delta(E) + \delta(D)$
4. Output E

“Scalar Multiplication” in \mathcal{R}_0 , II

Secret key: the divisor E below $n_A n_B \delta(D)$ where

- n_A and n_B are Alice’s and Bob’s respective secret scalars (exponents),
- D is some public starting divisor.

Average complexity: $1.5 \log_2(n)$ giant steps plus up to $4g \log_2(n)$ baby steps — *slower* than imaginary case.

Problem: the extra adjustment baby steps in steps 3 (b) and 3 (c) (ii).

Improvements, Real Scenario

Idea:

- Eliminate *all* adjustment steps,
- Replace “add” giant steps in the first round by baby steps.

Requires some precomputation and some extra baby steps in the second round of the DH protocol.

Heuristics:

- $\delta(D_{i+1}) - \delta(D_i) = 1$ with prob. $1 - O(q^{-1})$
(so $\delta(D_i) = g - 1 + i$ for $i \geq 2$)
- $\delta(D' \oplus D'') = \delta(D') + \delta(D'') - \lceil g/2 \rceil$
(so giant step = divisor addition + $\lceil \frac{g}{2} \rceil$ baby steps)

Improvements to First Round, I

Input: a divisor D and a scalar n

Output: the divisor of distance

$$2^{\lfloor \log_n(n) \rfloor} (g + 1) + n + \lceil g/2 \rceil$$

Precomputed: the divisor D_0 of distance $\lceil g/2 \rceil + 3$

1. Write $n = 2^l + b_{l-1}2^{l-1} + \dots + b_0$ (in binary)
2. Set $E = D_0$
3. For $i = 1$ to l do
 - (a) // *double* Replace E by $E \oplus E$
 - (b) // *baby step* If $b_i = 1$, apply a baby step to E
4. Output E

Improvements to First Round, II

Input: a divisor D and a scalar n

Output: the divisor of distance n

Precomputed: the divisor D^* of distance
 $2^{\lfloor \log_2(n) \rfloor} (g + 1) + g$

1. Compute the divisor D_0 of distance
 $2^{\lfloor \log_n(n) \rfloor} (g + 1) + n + \lceil g/2 \rceil$
using the previous algorithm
2. Compute $E = D_0 \oplus \overline{D^*}$
3. Output E

Alice and Bob use this algorithm in the first round to compute the divisors of distance n_A and n_B .

Improvements to Second Round

Input: a divisor D and a scalar n

Output: the divisor of distance $n\delta(D) + \lceil g/2 \rceil$

1. Apply $\lceil g/2 \rceil$ baby steps to D to obtain the divisor E of distance $\delta(D) + \lceil g/2 \rceil$
2. Write $n = 2^l + b_{l-1}2^{l-1} + \dots + b_0$ (in binary)
3. For $i = 1$ to l do
 - (a) // *double* Replace E by $E \oplus E$
 - (b) // *add* If $b_i = 1$, replace E by $E \oplus D$
4. Output E

Alice and Bob use this algorithm in the second round to compute the divisor E of distance $n_A n_B + \lceil g/2 \rceil$.

Operation Count

Let l be the scalar bit length, i.e. $2^l \leq n_A, n_B < 2^{l+1}$

	Imaginary	Real
Domain Parameters	Random divisor D	l gs $(l + 1) \lceil g/2 \rceil + g + 2$ bs
First Round	$1.5l$ gs	$l + 1$ gs $0.5l$ bs
Second Round	$1.5l$ gs	$1.5l$ gs $\lceil g/2 \rceil$ bs
Total Both Rounds	$3l$ gs	$2.5l + 1$ gs $0.5l + \lceil g/2 \rceil$ bs

Comparison – Real vs. Imaginary

If baby steps cost is negligible:

$$\text{Real} \approx \frac{2.5}{3} \times \text{Imaginary} \approx 0.83 \times \text{Imaginary}$$

- The exact speed-up depends on the baby step/giant step cost ratio.
- If $0.5l + \lceil g/2 \rceil$ baby steps cost less than $0.5l - 1$ giant steps, then the real scenario wins.
- We suspect (or hope?) that this is always the case, even when
 - the genus is small ($g = 2, 3$),
 - giant steps are optimally implemented.

Numerical Data

Implementation:

- Pentium IV, 2.53 GHz, Linux, GNU C++, NTL

Parameter Sizes:

- $2 \leq g \leq 6$
- $q^{g/2}$ has 80, 112, 128, 192, 256 bits

Timings (should be considered *very* preliminary):

- For $q = p$, the ratio varies between 0.82 and 0.9
- For $q = 2^n$, we get similar ratios except:
 - Slightly better ratios for $g = 4$, especially for large q^g ; as good as 0.79
 - For $g = 2$, the ratios are as good as 0.75 - 0.76

Discrete Logarithm Problem

- **Imaginary Model:**
 - Given a reduced divisor D and the reduced divisor in the divisor class of xD , find x .
- **Real Model** (four equivalent formulations):
 - Given a reduced principal divisor D and the reduced principal divisor E below $x\delta(D)$, find x ;
 - Given x , find the reduced principal divisor E below x ;
 - Given a reduced principal divisor E , find $\delta(E)$
 - Given a reduced principal $\mathbb{F}_q[C]$ -ideal \mathfrak{a} , find a generator of \mathfrak{a} (*Principal Ideal Problem*).

Security of both scenarios seems to be the same.

Conclusions and Future Work

Conclusions

- Unified framework for explicit divisor arithmetic in the both the real and imaginary scenario
- Real scenario seems faster for ephemeral Diffie-Hellman with the same level of security. This is due to the second, much faster, operation (baby steps) in the infrastructure.

Future (and Present) Work:

- Exact operation count for NUCOMP and comparison with straightforward giant steps
- Explicit formulas, NAF, windowing methods
- Other cryptographic protocols