

Cryptographic Aspects of Real Hyperelliptic Curves

Michael J. Jacobson, Jr.

`jacobs@cpsc.ucalgary.ca`



Centre for Information Security and Cryptography



Joint work with S. Erickson, J. Hammell, R. Scheidler, N. Shang, S. Shen, and A. Stein.

ECC 2009

Cryptographic Applications of Hyperelliptic Curves

Hyperelliptic curves over \mathbb{F}_q :

- divisor class group is finite abelian, can use for generic cryptographic protocols (Diffie-Hellman, El Gamal, etc...)
- discrete logarithm problem believed to be hard for small genus
- can achieve same security as elliptic curves, but with smaller q
- performance competitive and sometimes superior to elliptic curves

Real hyperelliptic curves:

- more general than widely-used imaginary model
- not as well studied, arithmetic considered not as efficient
- often arise naturally from constructive methods (eg. CM, pairings)
- how competitive and secure are they in practice?

Hyperelliptic Curves over \mathbb{F}_q

$C : y^2 + h(x)y = f(x); h, f \in \mathbb{F}_q[x];$ absolutely irreducible, non-singular

C is *imaginary* (one point at ∞) of *genus* g if

- q is odd, $h = 0$, f monic and square-free with $\deg(f) = 2g + 1$
- q is even, $h \neq 0$ with $\deg(h) \leq g$ and f monic with $\deg(f) = 2g + 1$

C is *real* (two points at ∞) of *genus* g if

- q is odd, $h = 0$, f square-free with $\deg(f) = 2g + 2$ and $\text{sgn}(f) = e^2$
- q is even, $h \neq 0$ monic with $\deg(h) = g + 1$ and
 - $\deg(f) \leq 2g + 1$ or
 - $\deg(f) = 2g + 2$ and $\text{sgn}(f) = e^2 + e$

Fact: every imaginary curve is birationally equivalent to one in real model

The Divisor Class Group (Imaginary)

$\mathcal{J} = \text{Jac}_{\mathbb{F}_q}(C)$: degree zero divisor class group of C over \mathbb{F}_q

- $|\mathcal{J}| \approx q^g$ (divisor class number)

Representation of degree zero divisors (Mumford): $D = (s; Q, P)$

- $s, Q, P \in \mathbb{F}_q[x]$, s and Q monic and unique, P unique modulo Q
- $Q \mid f + hP - P^2$

D is *reduced* if $s = 1$ and $\deg(Q) \leq g$

- every class $[D] \in \mathcal{J}$ has a unique reduced representative, $\text{Red}(D)$

Arithmetic in \mathcal{J} via reduced representatives (**giant steps**):

$$D' \oplus D'' \stackrel{\text{def}}{=} \text{Red}(D' + D'')$$

The Divisor Class Group (Real)

Representation of degree zero divisors: $D = (s; Q, P; v)$

- s, Q, P as before, $v \in \mathbb{Z}$
- reduced defined as before – no restrictions on v

Problem: reduced representatives of divisor classes are no longer unique

(Paulus-Rück 1999): Every $[D] \in \mathcal{J}$ has a unique reduced representative

$$\text{Red}'(D) = D_S - \deg(D_S)\infty_2 + v(\infty_1 - \infty_2)$$

with $0 \leq v \leq g - \deg(Q)$

Arithmetic:

- giant step, plus additional reduction steps until bound on v is satisfied

Infrastructure

Consider reduced divisors with $v = 0$

- correspond to reduced ideals $\mathfrak{a} = Q\mathbb{F}_q[x] + (P + y)\mathbb{F}_q[x] \in \mathbb{F}_q[C]$

Infrastructure: $\mathcal{R} = \{D_1, D_2, \dots, D_{r_C}\}$, D_i corresponds to a reduced, principal ideal \mathfrak{a}_i

- divisors are pair-wise inequivalent, ideals are equivalent
- ordered by *distance* $\delta(D_i) = \deg \alpha$, where $\mathfrak{a}_i = (\alpha)$, $\alpha \in \mathbb{F}_q(C)$
- **baby step**: given D_i , compute D_{i+1} (fast)

$|\mathcal{R}| \approx R$, where R is the *regulator* of C (order of divisor class $\infty_1 - \infty_2$)

- $|\mathcal{J}| = HR \approx q^g$, where H is the ideal class number (usually small)

Infra-STRUCTURE

\mathcal{R} is “almost” an Abelian group under giant steps

- Closure: $D', D'' \in \mathcal{R} \Rightarrow D' \oplus D'' \in \mathcal{R}$
- Identity: $D_1 = \mathbf{0} = (1, 0; 0)$
- Inverses: The inverse of $D = (a, b; v)$ is

$$-D = (a, -h - b; -\deg(a) - v)$$

- Commutativity: $D' \oplus D'' = D'' \oplus D'$
- “Almost” associative:

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

So $D \oplus (D' \oplus D'')$ is “close to” $(D \oplus D') \oplus D''$ (within $4g$ in distance)

Key Agreement in the Infrastructure

For $n \in [0, R)$, the divisor $D(n) \in \mathcal{R}$ below n is $D_i \in \mathcal{R}$ such that

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

(Scheidler, Stein, Williams 1996) Diffie-Hellman based key agreement

- Round 1 (fixed base): Alice computes $D(n)$; Bob computes $D(m)$
- Round 2 (variable base): Alice computes $D(nm)$ from $D(m)$ and n ; Bob computes $D(nm)$ from $D(n)$ and m

Can compute $D(n)$ efficiently (binary or NAF-based scalar multiplication)

- problem: extra baby steps required after each giant step to recover shortfall in distance

Finding n given $D(n)$ believed hard (infrastructure DLP)

Improvements to Scalar Multiplication in Infrastructure

(J. Scheidler, Stein 2007)

Fixed base: use baby steps as much as possible

- use NAF-based scalar multiplication, but apply baby step (or inverse) when NAF digit is not 0
- requires precomputed divisor D^* with $\delta(D^*) = 2^l(g + 1) + g$ (assuming l -term NAF expansions)

Variable base: eliminate all baby-step adjustments

- apply $d = \lceil g/2 \rceil$ baby steps to base divisor D' , apply scalar multiplication
- result is $D(n\delta(D') + d)$

Divisor Class Group Revisited

(Galbraith, Harrison, Mireles-Morales 2008) unique representative of $[D] \in \mathcal{J}$ using *balanced divisors*

- use $D_S - g/2(\infty_1 + \infty_2)$ or $D_S - (g + 1)/2\infty_1 - (g - 1)/2\infty_2$
- roughly equivalent to taking $v \approx g/2$ in Paulus-Rück representation
- at most one adjustment step required per giant step (none if g even)

Scalar multiplication should be same as or better than infrastructure variable base

Explicit Formulas

Divisor/ideal arithmetic described generically via polynomial arithmetic

- much faster in practice to describe explicitly by operations in \mathbb{F}_q
- (Erickson, J., Shang, Shen, Stein 2007) explicit formulas for genus 2, q odd
- (Erickson, J., Stein 2009) extended to q even, all special cases

Operation counts (I — inversion, S — square, M — multiplication):

Model	Baby	Add	Double
Imaginary	1I, 1S, 10M	1I, 3S, 22M	1I, 5S, 22M
Real (q odd)	1I, 2S, 4M	1I, 2S, 26M	1I, 4S, 28M
Real (q even)	1I, 1S, 5M	1I, 1S, 27M	1I, 2S, 29M

Erickson 2009: inversion-free versions

Key Exchange Timings, q odd

Intel Core Duo 2.66 Ghz, Linux, g++ 4.1.2, NTL, times in milliseconds

Security (in bits)	Imaginary			Real		
	Fixed	Var	DH Total	Fixed	Var	DH Total
80	2.137	2.304	4.440	2.307	2.618	4.925
112	3.545	3.942	7.487	3.809	4.469	8.278
128	4.702	5.149	9.851	5.003	5.869	10.872
192	10.526	11.562	22.088	11.192	13.048	24.240
256	15.560	17.077	32.636	16.492	19.168	35.660

Key Exchange Timings, q even

Security (in bits)	Imaginary			Real		
	Fixed	Var	DH Total	Fixed	Var	DH Total
80	4.721	5.331	10.052	5.112	6.139	11.250
112	4.096	4.475	8.571	4.425	5.076	9.500
128	4.814	5.304	10.118	5.138	5.920	11.057
192	11.700	12.942	24.641	12.715	14.721	27.436
256	22.255	24.572	46.827	24.525	28.326	52.851

Infrastructure Problems in a Group?

(Fontein 2008) infrastructure can be embedded in a cyclic group of order R

Uses technique called “ f -representations”

- fill in gaps between consecutive ideals with distance $\text{gap} > 1$
- can be generalized to function fields with higher unit rank

Implication: generic group algorithms can be applied to infrastructure problems

- Eg. Pohlig-Hellman techniques can be applied if R is smooth

Infrastructure in the Jacobian?

(Mireles-Morales 2008) embed \mathcal{R} into subgroup of \mathcal{J} generated by

$$\infty_1 - \infty_2$$

- baby step corresponds to addition of $\infty_1 - \infty_2$

Implication: infrastructure computations can be done in a subgroup of \mathcal{J}

- generic group algorithms can be applied
- use balanced divisor representation of elements of \mathcal{J} for efficiency

Key exchange has similar advantages to infrastructure optimizations

- use $\infty_1 - \infty_2$ as base for round 1 (additions are baby steps)
- at most one adjustment per divisor addition in round 2

The Infrastructure DLP

Known results:

- generic complexity (baby-step giant-step or Pollard-rho): $O(\sqrt{R})$
- (Stein 1994) equivalent to ECDLP for $g = 1$
- (Fontein 2008) can apply Pohlig-Hellman if R is smooth
- (Mireles-Morales 2008) same as discrete logarithm in subgroup of \mathcal{J} generated by $\infty_1 - \infty_2$

Large genus:

- (Müller, Stein, Thiel 1992): if $g > \log q$, $O(L_{q^{2g+2}}[1.44 + o(1)])$
- no implementation provided

Overview of Index Calculus

Factor base: prime ideals \mathfrak{p} with $\deg N(\mathfrak{p}) \leq B$ (smoothness bound)

Step 1: find random *relations*

- (principal) ideals that factor over the factor base (B -smooth)
- requires fast generation of candidates (eg. random walk) and fast smoothness testing

Step 2: linear algebra with relation matrix (columns correspond to ideal factorizations)

- linear system for DLP (over \mathbb{Z} or modulo group order)
- find elements in kernel over \mathbb{Z} for regulator
- determinant / Smith normal form for class number / group structure

New Results (Hammell, J. 2008)

Find relations by iterating through the infrastructure using baby steps

- faster operation $O(g)$ as opposed to random walk $O(g^2)$

New analysis

- incorporates new relation generation, more recent linear algebra
- complexity $O(L_{q^g} [2.45 + o(1)])$ if $g > \log q$, assuming smooth reduced ideals are distributed evenly amongst equivalence classes
- $O(L_{q^g} [2.83 + o(1)])$ to compute regulator, $O(L_{q^g} [3.45 + o(1)])$ to compute class number and group structure

First implementation of index calculus for solving infrastructure DLP

- versions using new relation generation method and sieving

Numerical Results (even q): Computing R

Times given as hh:mm:ss, Pentium 4, 3.0 GHz, 1 GB RAM

q	g	$\log R$	BSGS	Baby Walk	Sieving
2^2	5	9	:00	:00	:00
2^2	10	21	:00	:01	:00
2^2	15	31	:06	:02	:02
2^2	20	39	2:27	:29	:25
2^2	25	50	35:47:10	2:59	1:39
2^2	30	61	—	17:56	6:34
2^2	35	71	—	1:24:53	32:35
2^2	40	80	—	14:29:13	7:11:32
2^6	5	31	:03	:02	:13
2^6	10	61	—	3:39	2:24
2^6	15	91	—	18:39:49	17:11:07

Numerical Results (odd q): Computing R

Times given as hh:mm:ss, Pentium 4, 3.0 GHz, 1 GB RAM

q	g	$\log R$	BSGS	Baby Walk	Sieving
5	5	8	:00	:00	:00
5	10	17	:00	:01	:00
5	15	20	:00	:03	:03
5	20	33	:04	:01	:09
5	25	41	2:41	:07	:27
5	30	51	18:02:51	:31	1:33
5	35	58	—	4:22	12:47
5	40	64	—	22:00	1:43:35
67	5	30	:01	:03	:10
67	10	60	—	2:14	17:07
67	15	85	—	13:31:10	167:34:27

Future Work: Arithmetic

Explicit formulas for low genus:

- further improvements, new techniques (NUCOMP?)?
- use alternative real model (van der Poorten, Scheidler)?
- Edwards model?
- genus 3?

Scalar multiplication:

- fast cubing algorithms and double-base number systems
- find special fast tupling algorithms, combine with fast reduction
- using Jacobian (and balanced divisors) in place of the infrastructure

Future Work: DLP

Extend algorithms and analysis to incorporate latest techniques, including double large primes

Improve sieving, especially for q odd

Apply results to low genus:

- relation generation via sieving?
- infrastructure DLP in low genus (complexity and implementation)

Explore relationship between HCDLP and infrastructure DLP for $g > 1$