Cryptographic Aspects of Real Hyperelliptic Curves

Michael J. Jacobson, Jr.

jacobs@cpsc.ucalgary.ca

Centre for Information Security and Cryptography

Joint work with J. Hammell, R. Scheidler, and A. Stein.

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

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

The Divisor Class Group (Imaginary)

$Cl^0 = \mathrm{Div}^0(C)$: degree zero divisor class group of C over $\mathbb{F}_q$

- $|Cl^0| \approx q^g$ (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 $[D] \in Cl^0$ has a unique reduced representative, $\mathrm{Red}(D)$

Arithmetic in Cl^0 via reduced representatives (**giant steps**):

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

The Divisor Class Group (Real)

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

- s, Q, P as before, $v \in \mathbb{Z}$,

$$D = D' - \deg(Q)\infty_2 + v(\infty_1 - \infty_2)$$

(D' finite, represented by (Q, P))

- reduced defined as before — no restrictions on v
- reduced representatives of divisor classes are no longer unique

(Paulus, Rück 1999): Every $[D] \in Cl^0$ has a unique reduced representative $\mathrm{Red}'(D) = (Q, P; v)$ with $0 \leq v \leq g - \deg(Q)$

- arithmetic (giant steps): as before, 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: $\mathcal{R} = \{D_1, D_2, \ldots, D_{r_C}\}$, D_i corresponds to a reduced, *principal* ideal $\mathfrak{a}_i$

- divisors are pair-wise inequivalent, ideals are equivalent
- ordered by *distance* $\delta(D_i) = \deg \alpha_i$, where $\mathfrak{a}_i = (\alpha_i)$, $\alpha_i \in \mathbb{F}_q(C)$
- for giant steps, have $\delta(D' \oplus D'') = \delta(D') + \delta(D'') - d$, $0 \leq d \leq 2g$
- **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 $\infty_1 - \infty_2$)

- $|Cl^0| = HR \approx q^g$, where H is the ideal class number (usually small)

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) for each non-zero NAF term
- requires precomputed divisor $D^* = D(2^l(g+1)+g)$ (assuming l-term NAF expansions)

(J., Scheidler, Stein 2007) variable base: eliminate all distance adjustments

- apply $d = \lceil g/2 \rceil$ baby steps to base divisor D_0, apply scalar multiplication to the result
- output is $D(n\delta(D_0)+d)$

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) formulas for genus 2, q odd
- (Erickson, J., Stein 2009) extended to q even, all special cases

Operation counts in $\mathbb{F}_q$ (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 (projective) formulas

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

q even similar, but slower

Infrastructure as a Group?

(Mireles-Morales 2008) embed $\mathcal{R}$ into subgroup generated by $\infty_1 - \infty_2$

- baby step corresponds to adding of $\infty_1 - \infty_2$
- implication: can do infrastructure computations in a subgroup of Cl^0

(Galbraith, Harrison, Mireles-Morales 2008) unique representative of $[D] \in Cl^0$ using *balanced divisors*

- use $D' - g/2(\infty_1 + \infty_2)$ or $D' - (g+1)/2\infty_1 - (g-1)/2\infty_2$
- roughly equivalent to taking $v \approx g/2$ in Paulus-Rück representation

Key exchange has similar advantages to infrastructure optimizations

- use $\infty_1 - \infty_2$ as base for round 1, pre-computed D^* not required
- at most one distance adjustment per giant step (none if g even)

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 Cl^0 generated by $\infty_1 - \infty_2$

Large genus:

- (Müller, Stein, Thiel 1992): if $g > \log q$, $O(L_{q^{2g+2}}[1.44 + o(1)])$
- index-calculus: find random *relations* (smooth principal ideals), solve linear algebra problem
- no implementation provided

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 $O(L_{qg}[2.45 + o(1)])$ if $g > \log q$, assuming smooth reduced ideals are distributed evenly amongst equivalence classes
- $O(L_{qg}[2.83 + o(1)])$ to compute regulator, $O(L_{qg}[3.45 + o(1)]))$ 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	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

Odd q similar, but sieving is much worse

Future Work

Arithmetic:

- fast cubing algorithms and double-base number systems
- use fast m-tupling algorithms, combine with fast reduction (next talk)
- use alternative real model for explicit formulas (van der Poorten, Scheidler)?
- better formulas via NUCOMP?
- genus 3?

Infrastructure DLP:

- improve sieving, especially for q odd
- apply results to low genus (complexity and sieving)
- explore relationship between HCDLP and infrastructure DLP for $g > 1$