Real Hyperelliptic Curves

Renate Scheidler

rscheidl@math.ucalgary.ca

Centre for Information Security and Cryptography

University of Calgary

Fields Institute Workshop on New Directions in Cryptography, June 25-27, 2008, University of Ottawa

Joint work with

Mike Jacobson (University of Calgary) and **Andreas Stein** (University of Oldenburg)

Research supported in part by NSERC of Canada

Hyperelliptic Curves over $\mathbb{F}_q$

Hyperelliptic Curves over $\mathbb{F}_q$

$$\boxed{C \;:\quad y^2 + h(x)y \;=\; f(x)}$$

Hyperelliptic Curves over $\mathbb{F}_q$

$$\boxed{C \ : \quad y^2 + h(x)y \ = \ f(x)}$$

$f, h \in \mathbb{F}_q[x]$; $h = 0$ if q odd;
absolutely irreducible; non-singular; of *genus* g

Hyperelliptic Curves over $\mathbb{F}_q$

$$\boxed{C \; : \quad y^2 + h(x)y \; = \; f(x)}$$

$f, h \in \mathbb{F}_q[x]$; $h = 0$ if q odd;
absolutely irreducible; non-singular; of *genus* g

- **Imaginary Model**
  - f monic and $\deg(f) = 2g + 1$
  - $\deg(h) \leq g$ if q even

Hyperelliptic Curves over $\mathbb{F}_q$

$$\boxed{C \ : \quad y^2 + h(x)y \ = \ f(x)}$$

$f, h \in \mathbb{F}_q[x]$; $h = 0$ if q odd;
absolutely irreducible; non-singular; of *genus* g

- **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
    - $\deg(f) = 2g + 2$, sgn$(f) = e^2 + e$ ($e \in \mathbb{F}_q^*$)

Degree 0 Divisors (C Imaginary)

Degree 0 Divisors (C Imaginary)

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

Degree 0 Divisors (C Imaginary)

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

Representation of degree zero divisors: $D = (s; a, b)$:

- $s, a, b \in \mathbb{F}_q[x]$, s and a monic
- s and a unique, b (mod a) unique
- $a \mid f + hb - b^2$

Degree 0 Divisors (C Imaginary)

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

Representation of degree zero divisors: $D = (s; a, b)$:

- $s, a, b \in \mathbb{F}_q[x]$, s and a monic
- s and a unique, b (mod a) unique
- $a \mid f + hb - b^2$

D *semi-reduced*: $s = 1$
D *reduced*: $s = 1$ and $\deg(a) \leq g$.

Degree 0 Divisors (C Imaginary)

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

Representation of degree zero divisors: $D = (s; a, b)$:

- $s, a, b \in \mathbb{F}_q[x]$, s and a monic
- s and a unique, b (mod a) unique
- $a \mid f + hb - b^2$

D *semi-reduced*: $s = 1$
D *reduced*: $s = 1$ and $\deg(a) \leq g$.

Theorem: Every class $[D] \in \mathcal{J}$ has a unique reduced representative $\text{Red}(D)$

Degree 0 Divisors (C Imaginary)

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

Representation of degree zero divisors: $D = (s; a, b)$:

- $s, a, b \in \mathbb{F}_q[x]$, s and a monic
- s and a unique, b (mod a) unique
- $a \mid f + hb - b^2$

D *semi-reduced*: $s = 1$
D *reduced*: $s = 1$ and $\deg(a) \leq g$.

Theorem: Every class $[D] \in \mathcal{J}$ has a unique reduced representative $\text{Red}(D)$

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

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

Giant Step Arithmetic

Giant Step Arithmetic

- Cantor's algorithm (Cantor 1987)
  divisor addition with subsequent reduction steps
  $17g^2 + O(g)$ operations in $\mathbb{F}_q$ (Stein 2001)

Giant Step Arithmetic

- Cantor's algorithm (Cantor 1987)
  divisor addition with subsequent reduction steps
  $17g^2 + O(g)$ operations in $\mathbb{F}_q$ (Stein 2001)

- NUCOMP (Shanks-Atkin 1989, van der Poorten 2003)
  Cg^2 operation in $\mathbb{F}_q$, $C < 17$

Giant Step Arithmetic

- Cantor's algorithm (Cantor 1987)
  divisor addition with subsequent reduction steps
  $17g^2 + O(g)$ operations in $\mathbb{F}_q$ (Stein 2001)

- NUCOMP (Shanks-Atkin 1989, van der Poorten 2003)
  Cg^2 operation in $\mathbb{F}_q$, $C < 17$

- Explicit formulas for low genus curves:

Giant Step Arithmetic

- Cantor's algorithm (Cantor 1987)
  divisor addition with subsequent reduction steps
  $17g^2 + O(g)$ operations in $\mathbb{F}_q$ (Stein 2001)

- NUCOMP (Shanks-Atkin 1989, van der Poorten 2003)
  Cg^2 operation in $\mathbb{F}_q$, $C < 17$

- Explicit formulas for low genus curves:
  - Imaginary: $g = 2, 3, 4$
    `www.hyperelliptic.org/EFD/`

Giant Step Arithmetic

- Cantor's algorithm (Cantor 1987)
  divisor addition with subsequent reduction steps
  $17g^2 + O(g)$ operations in $\mathbb{F}_q$ (Stein 2001)

- NUCOMP (Shanks-Atkin 1989, van der Poorten 2003)
  Cg^2 operation in $\mathbb{F}_q$, $C < 17$

- Explicit formulas for low genus curves:
  - Imaginary: $g = 2, 3, 4$
    `www.hyperelliptic.org/EFD/`
  - Real: $g = 2$, affine coordinates
    Erickson-Jacobson-Shang-Shen-Stein, WAIFI 2007

Key Agreement in $\mathcal{J}$ (Koblitz 1989)

Key Agreement in $\mathcal{J}$ (Koblitz 1989)

Alice and Bob agree on q, C imaginary, a reduced divisor D

Key Agreement in $\mathcal{J}$ (Koblitz 1989)

Alice and Bob agree on q, C imaginary, a reduced divisor D

	Alice	Bob
1.	Generates $m \in_R]1, \text{ord}(D)[$	Generates $n \in_R]1, \text{ord}(D)[$
	// Fixed base scenario //	
2.	Sends $D_m = \text{Red}(mD)$ to Bob	Sends $D_n = \text{Red}(nD)$ to Alice
	// Variable base scenario //	
3.	Computes $K = \text{Red}(mD_n)$	Computes $K = \text{Red}(nD_m)$

Key Agreement in $\mathcal{J}$ (Koblitz 1989)

Alice and Bob agree on q, C imaginary, a reduced divisor D

	Alice	**Bob**
1.	Generates $m \in_R]1, \text{ord}(D)[$	Generates $n \in_R]1, \text{ord}(D)[$
	// Fixed base scenario //	
2.	Sends $D_m = \text{Red}(mD)$ to Bob	Sends $D_n = \text{Red}(nD)$ to Alice
	// Variable base scenario //	
3.	Computes $K = \text{Red}(mD_n)$	Computes $K = \text{Red}(nD_m)$

The secret is the reduced divisor $K = \text{Red}(mnD)$

Key Agreement in $\mathcal{J}$ (Koblitz 1989)

Alice and Bob agree on q, C imaginary, a reduced divisor D

	Alice	Bob
1.	Generates $m \in_R]1, \text{ord}(D)[$	Generates $n \in_R]1, \text{ord}(D)[$
	// Fixed base scenario //	
2.	Sends $D_m = \text{Red}(mD)$ to Bob	Sends $D_n = \text{Red}(nD)$ to Alice
	// Variable base scenario //	
3.	Computes $K = \text{Red}(mD_n)$	Computes $K = \text{Red}(nD_m)$

The secret is the reduced divisor $K = \text{Red}(mnD)$

$\langle D \rangle \approx |\mathcal{J}| \approx q^g$ (exponentially large in the size of C)

Key Agreement in $\mathcal{J}$ (Koblitz 1989)

Alice and Bob agree on q, C imaginary, a reduced divisor D

	Alice	Bob
1.	Generates $m \in_R \,]1, \text{ord}(D)[$	Generates $n \in_R \,]1, \text{ord}(D)[$
	// Fixed base scenario //	
2.	Sends $D_m = \text{Red}(mD)$ to Bob	Sends $D_n = \text{Red}(nD)$ to Alice
	// Variable base scenario //	
3.	Computes $K = \text{Red}(mD_n)$	Computes $K = \text{Red}(nD_m)$

The secret is the reduced divisor $K = \text{Red}(mnD)$

$\langle D \rangle \approx |\mathcal{J}| \approx q^g$ (exponentially large in the size of C)

DLP is exponential for small g

($g = 2$ is best; DLP complexity $O(q) = O(\sqrt{|\mathcal{J}|})$)

Degree 0 Divisors (C Real)

Degree 0 Divisors (C Real)

Representation of degree zero divisors: $D = (s; a, b; v)$:

- s, a, b as before
- $v \in \mathbb{Z}$

Degree 0 Divisors (C Real)

Representation of degree zero divisors: $D = (s; a, b; v)$:

- s, a, b as before
- $v \in \mathbb{Z}$

Semi-reduced and reduced defined as before – no restrictions on v

Degree 0 Divisors (C Real)

Representation of degree zero divisors: $D = (s; a, b; v)$:

- s, a, b as before
- $v \in \mathbb{Z}$

Semi-reduced and reduced defined as before – no restrictions on v

Fact: Reduced representatives of divisor classes are no longer unique

Degree 0 Divisors (C Real)

Representation of degree zero divisors: $D = (s; a, b; v)$:

- s, a, b as before
- $v \in \mathbb{Z}$

Semi-reduced and reduced defined as before – no restrictions on v

Fact: Reduced representatives of divisor classes are no longer unique

Theorem: Every class $[D] \in \mathcal{J}$ has a unique reduced representative Red'$(D) = (a, b, v)$ where v is restricted to a suitable interval of length $g - \deg(a) + 1$

(Paulus-Rück 1999; Galbraith-Harrison-Mireless 2008)

Degree 0 Divisors (C Real)

Representation of degree zero divisors: $D = (s; a, b; v)$:

- s, a, b as before
- $v \in \mathbb{Z}$

Semi-reduced and reduced defined as before – no restrictions on v

Fact: Reduced representatives of divisor classes are no longer unique

Theorem: Every class $[D] \in \mathcal{J}$ has a unique reduced representative Red'$(D) = (a, b, v)$ where v is restricted to a suitable interval of length $g - \deg(a) + 1$

(Paulus-Rück 1999; Galbraith-Harrison-Mireless 2008)

Could use this again for arithmetic in $\mathcal{J}$

Computing Red'(D)

Computing Red'(D)

- Compute $D' \oplus D'' = \text{Red}(D' + D'')$ as in the imaginary case using a giant step — Cg^2 operations in $\mathbb{F}_q$

Computing Red'(D)

- Compute $D' \oplus D'' =$ Red$(D' + D'')$ as in the imaginary case using a giant step — Cg^2 operations in $\mathbb{F}_q$
- Apply further reduction steps to until Red'(D) is reached — up to $C'g^2$ operations in $\mathbb{F}_q$

Computing Red'(D)

- Compute $D' \oplus D'' =$ Red$(D' + D'')$ as in the imaginary case using a giant step — Cg^2 operations in $\mathbb{F}_q$
- Apply further reduction steps to until Red'(D) is reached — up to $C'g^2$ operations in $\mathbb{F}_q$

$(C + C')g^2$ field operations — slower than imaginary Jacobian arithmetic

Computing Red'(D)

- Compute $D' \oplus D'' = \text{Red}(D' + D'')$ as in the imaginary case using a giant step — Cg^2 operations in $\mathbb{F}_q$
- Apply further reduction steps to until Red'(D) is reached — up to $C'g^2$ operations in $\mathbb{F}_q$

$(C + C')g^2$ field operations — slower than imaginary Jacobian arithmetic

Henceforth, let C be real. Then $|\mathcal{J}| = HR$ where

Computing Red'(D)

- Compute $D' \oplus D'' = \text{Red}(D' + D'')$ as in the imaginary case using a giant step — Cg^2 operations in $\mathbb{F}_q$
- Apply further reduction steps to until Red'(D) is reached — up to $C'g^2$ operations in $\mathbb{F}_q$

$(C + C')g^2$ field operations — slower than imaginary Jacobian arithmetic

Henceforth, let C be real. Then $|\mathcal{J}| = HR$ where

- H is the order of the ideal class group of the coordinate ring of C; usually very small

Computing Red'(D)

- Compute $D' \oplus D'' = \text{Red}(D' + D'')$ as in the imaginary case using a giant step — Cg^2 operations in $\mathbb{F}_q$
- Apply further reduction steps to until Red'(D) is reached — up to $C'g^2$ operations in $\mathbb{F}_q$

$(C + C')g^2$ field operations — slower than imaginary Jacobian arithmetic

Henceforth, let C be real. Then $|\mathcal{J}| = HR$ where

- H is the order of the ideal class group of the coordinate ring of C; usually very small
- R is the *regulator* of C, i.e. the order of the divisor class of $\infty_1 - \infty_2$ where ∞_1 and ∞_2 are the two points at infinity; usually $R \approx |\mathcal{J}| \approx q^g$

Infrastructure

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid\ D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid\ D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid\ D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

- $\mathcal{R}$ is finite and of cardinality $\approx R \approx q^g$

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid \ D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

- $\mathcal{R}$ is finite and of cardinality $\approx R \approx q^g$
- $\mathcal{R}$ is ordered by *distance*: $\delta(D) = -v$, so

$$\mathcal{R} \ = \ \{D_1 = \mathbf{0}, D_2, \ldots, D_r\}, \qquad D_{i+1} = D_i - \text{div}\left(\frac{a_i + y}{b_i}\right)$$

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

- $\mathcal{R}$ is finite and of cardinality $\approx R \approx q^g$
- $\mathcal{R}$ is ordered by *distance*: $\delta(D) = -v$, so

  $\mathcal{R} = \{D_1 = \mathbf{0}, D_2, \ldots, D_r\}$, $\quad D_{i+1} = D_i - \text{div}\left(\frac{a_i + y}{b_i}\right)$

  - $0 = \delta_1 < \delta_2 < \cdots < \delta_r < R$

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid \ D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

- $\mathcal{R}$ is finite and of cardinality $\approx R \approx q^g$
- $\mathcal{R}$ is ordered by *distance*: $\delta(D) = -v$, so

  $\mathcal{R} \ = \ \{D_1 = \mathbf{0}, D_2, \ldots, D_r\}$, $\quad D_{i+1} = D_i -$ div $\left(\frac{a_i + y}{b_i}\right)$

  - $0 = \delta_1 < \delta_2 < \cdots < \delta_r < R$
  - $\delta_1 = 0$, $\ \delta_2 = g + 1$, $\ 1 \leq \delta_{i+1} - \delta_i \leq g$ for $2 \leq i \leq r - 1$

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid \; D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

- $\mathcal{R}$ is finite and of cardinality $\approx R \approx q^g$
- $\mathcal{R}$ is ordered by *distance*: $\delta(D) = -v$, so

  $$\mathcal{R} \;=\; \{D_1 = \mathbf{0}, D_2, \ldots, D_r\}, \qquad D_{i+1} = D_i - \text{div}\left(\frac{a_i + y}{b_i}\right)$$

  - $0 = \delta_1 < \delta_2 < \cdots < \delta_r < R$
  - $\delta_1 = 0, \;\; \delta_2 = g + 1, \;\; 1 \leq \delta_{i+1} - \delta_i \leq g$ for $2 \leq i \leq r - 1$
- A reduction step moves from D_i to D_{i+1} and is also known as a **baby step** — $O(g)$ operations in $\mathbb{F}_q$

Infrastructure

Infrastructure:

$\mathcal{R} = \{D \mid D$ is a reduced *principal* divisor with $0 \leq -v < R\}$

Properties

- $\mathcal{R}$ is finite and of cardinality $\approx R \approx q^g$
- $\mathcal{R}$ is ordered by *distance*: $\delta(D) = -v$, so

  $$\mathcal{R} \;=\; \{D_1 = \mathbf{0}, D_2, \ldots, D_r\}, \qquad D_{i+1} = D_i - \text{div}\left(\frac{a_i + y}{b_i}\right)$$

  - $0 = \delta_1 < \delta_2 < \cdots < \delta_r < R$
  - $\delta_1 = 0,\ \delta_2 = g + 1,\ 1 \leq \delta_{i+1} - \delta_i \leq g$ for $2 \leq i \leq r - 1$
- A reduction step moves from D_i to D_{i+1} and is also known as a **baby step** — $O(g)$ operations in $\mathbb{F}_q$
- $D_{mr+i} = D_i + mR(\infty_1 - \infty_2)$ for $m \in \mathbb{N}$ and $1 \leq i \leq r$

Infra-STRUCTURE

Infra-STRUCTURE

$\mathcal{R}$ is "almost" an Abelian group under giant steps:

Infra-STRUCTURE

$\mathcal{R}$ is "almost" an Abelian group under giant steps:

- Closure: $D', D'' \in \mathcal{R} \;\Rightarrow\; D' \oplus D'' \in \mathcal{R}$

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: $D_1 = \mathbf{0} = (1, 0; 0)$

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

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: $D' \oplus D'' = D'' \oplus D'$

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:

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

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:

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

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

More on Infrastructure

More on Infrastructure

- If $D = (a, b; v) \in \mathcal{R}$, then (a, b) determines $v = -\delta(D)$ uniquely and vice versa. So we can write $D = (a, b)$

More on Infrastructure

- If $D = (a, b; v) \in \mathcal{R}$, then (a, b) determines $v = -\delta(D)$ uniquely and vice versa. So we can write $D = (a, b)$
- Given $D = (a, b) \in \mathcal{R}$, it is computationally infeasible to find $\delta(D)$ — infrastructure discrete log problem

More on Infrastructure

- If $D = (a, b; v) \in \mathcal{R}$, then (a, b) determines $v = -\delta(D)$ uniquely and vice versa. So we can write $D = (a, b)$
- Given $D = (a, b) \in \mathcal{R}$, it is computationally infeasible to find $\delta(D)$ — infrastructure discrete log problem

Divisors of Fixed Distance:

More on Infrastructure

- If $D = (a, b; v) \in \mathcal{R}$, then (a, b) determines $v = -\delta(D)$ uniquely and vice versa. So we can write $D = (a, b)$
- Given $D = (a, b) \in \mathcal{R}$, it is computationally infeasible to find $\delta(D)$ — infrastructure discrete log problem

Divisors of Fixed Distance: For $n \in [0, R)$, the divisor $D(n) \in \mathcal{R}$ *below* n is the divisor $D_i \in \mathcal{R}$ such that

$$\delta_i \le n < \delta_{i+1}$$

More on Infrastructure

- If $D = (a, b; v) \in \mathcal{R}$, then (a, b) determines $v = -\delta(D)$ uniquely and vice versa. So we can write $D = (a, b)$
- Given $D = (a, b) \in \mathcal{R}$, it is computationally infeasible to find $\delta(D)$ — infrastructure discrete log problem

Divisors of Fixed Distance: For $n \in [0, R)$, the divisor $D(n) \in \mathcal{R}$ *below* n is the divisor $D_i \in \mathcal{R}$ such that

$$\delta_i \ \leq \ n \ < \ \delta_{i+1}$$

Key Point: $n \rightsquigarrow D(n)$ easy, $D \rightsquigarrow \delta(D)$ hard

Key Agreement in $\mathcal{R}$

Key Agreement in $\mathcal{R}$

(S.-Stein-Williams 1996) Alice and Bob agree on a real hyperelliptic curve C over a finite field $\mathbb{F}_q$ with regulator R

Key Agreement in $\mathcal{R}$

(S.-Stein-Williams 1996) Alice and Bob agree on a real hyperelliptic curve C over a finite field $\mathbb{F}_q$ with regulator R

	Alice	**Bob**
1.	Generates $m \in_R \,]1, R[$	Generates $n \in_R \,]1, R[$
	// Fixed base scenario //	
2.	Sends $D(m)$ to Bob	Sends $D(n)$ to Alice
	// Variable base scenario //	
3.	Computes $D(nm)$ from $D(n)$ and m	Computes $D(mn)$ from $D(m)$ and n

Key Agreement in $\mathcal{R}$

(S.-Stein-Williams 1996) Alice and Bob agree on a real hyperelliptic curve C over a finite field $\mathbb{F}_q$ with regulator R

	Alice	Bob
1.	Generates $m \in_R]1, R[$	Generates $n \in_R]1, R[$
2.	*// Fixed base scenario //*	
	Sends $D(m)$ to Bob	Sends $D(n)$ to Alice
3.	*// Variable base scenario //*	
	Computes $D(nm)$ from $D(n)$ and m	Computes $D(mn)$ from $D(m)$ and n

The secret is $K = D(mn)$

Key Agreement in $\mathcal{R}$

(S.-Stein-Williams 1996) Alice and Bob agree on a real hyperelliptic curve C over a finite field $\mathbb{F}_q$ with regulator R

	Alice	**Bob**
1.	Generates $m \in_R]1, R[$	Generates $n \in_R]1, R[$
	// Fixed base scenario //	
2.	Sends $D(m)$ to Bob	Sends $D(n)$ to Alice
	// Variable base scenario //	
3.	Computes $D(nm)$ from $D(n)$ and m	Computes $D(mn)$ from $D(m)$ and n

The secret is $K = D(mn)$

Same size key space and security as imaginary scenario, but slower – as this simply mimics real Jacobian arithmetic

Non-Adjacent Form

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $n = \sum_{i=0}^{l} b_i 2^{l-i}$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $n = \sum_{i=0}^{l} b_i 2^{l-i}$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Idea: $2^{i+1} + 2^i = 2^{i+2} - 2^i$

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $$n = \sum_{i=0}^{l} b_i 2^{l-i}$$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Idea: $2^{i+1} + 2^i = 2^{i+2} - 2^i$

Properties

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $$n = \sum_{i=0}^{l} b_i 2^{l-i}$$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Idea: $2^{i+1} + 2^i = 2^{i+2} - 2^i$

Properties

- For any $n \in \mathbb{N}$, NAF exists and is unique

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $$n = \sum_{i=0}^{l} b_i 2^{l-i}$$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Idea: $2^{i+1} + 2^i = 2^{i+2} - 2^i$

Properties

- For any $n \in \mathbb{N}$, NAF exists and is unique
- $2^{l+1} < 3n < 2^{l+2}$, so NAF length is at most one more than the binary length of n

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $\quad n = \sum_{i=0}^{l} b_i 2^{l-i}$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Idea: $2^{i+1} + 2^i = 2^{i+2} - 2^i$

Properties

- For any $n \in \mathbb{N}$, NAF exists and is unique
- $2^{l+1} < 3n < 2^{l+2}$, so NAF length is at most one more than the binary length of n
- Only 1/3 of all the digits is expected to be non-zero (as opposed to 1/2 of the ordinary bits of n)

Non-Adjacent Form

Useful for scalar multiplication in groups where computing inverses is cheap (note: $D = (a, b) \ \Rightarrow -D = (a, -h - b)$)

The **non-adjacent form** of $n \in \mathbb{N}$ is $\quad n = \sum_{i=0}^{l} b_i 2^{l-i}$

$b_0 = 1$, $b_i \in \{\pm 1, 0\}$, no two consecutive b_i are non-zero

Idea: $2^{i+1} + 2^i = 2^{i+2} - 2^i$

Properties

- For any $n \in \mathbb{N}$, NAF exists and is unique
- $2^{l+1} < 3n < 2^{l+2}$, so NAF length is at most one more than the binary length of n
- Only 1/3 of all the digits is expected to be non-zero (as opposed to 1/2 of the ordinary bits of n)
- NAF is easily computable (almost for free)

Scalar Multiplication in $\mathcal{J}$

Scalar Multiplication in $\mathcal{J}$

Input: a reduced divisor D and a scalar $n = \sum_{i=0}^{l} b_i 2^{l-i}$ in NAF

Scalar Multiplication in $\mathcal{J}$

Input: a reduced divisor D and a scalar $n = \sum_{i=0}^{l} b_i 2^{l-i}$ in NAF

Output: the reduced divisor Red(nD)

Scalar Multiplication in $\mathcal{J}$

Input: a reduced divisor D and a scalar $n = \sum_{i=0}^{l} b_i 2^{l-i}$ in NAF

Output: the reduced divisor Red(nD)

Algorithm:

1. Set $E = D$
2. For $i = 1$ to l do

// Double	Replace E by $E \oplus E$
// Add	If $b_i = 1$, replace E by $E \oplus D$
	If $b_i = -1$, replace E by $E \oplus (-D)$

3. Output E

Scalar Multiplication in $\mathcal{J}$

Input: a reduced divisor D and a scalar $n = \sum_{i=0}^{l} b_i 2^{l-i}$ in NAF

Output: the reduced divisor $\mathrm{Red}(nD)$

Algorithm:

1. Set $E = D$
2. For $i = 1$ to l do

   // Double Replace E by $E \oplus E$

   // Add If $b_i = 1$, replace E by $E \oplus D$

   If $b_i = -1$, replace E by $E \oplus (-D)$
3. Output E

l doubles, $l/3$ adds

Variable Base Scalar Mult. in $\mathcal{R}$

Variable Base Scalar Mult. in $\mathcal{R}$

Input: $D \in \mathcal{R}$ and $n = \sum b_i 2^{l-i}$ in NAF

Variable Base Scalar Mult. in $\mathcal{R}$

Input: $D \in \mathcal{R}$ and $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor E below $n\delta(D)$

Variable Base Scalar Mult. in $\mathcal{R}$

Input: $D \in \mathcal{R}$ and $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor E below $n\delta(D)$
Algorithm:

1. Set $E = D$
2. For $i = 1$ to l do
   // Double Replace E by $E \oplus E$
   // Adjust Replace E by $D(2\delta(E))$
   If $b_i \neq 0$ then
      If $b_i = 1$, set $D' = D$
      If $b_i = -1$, set $D' = -D$
      // Add replace E by $E \oplus D'$
      // Adjust Replace E by $D(\delta(E) + \delta(D'))$
3. Output E

Variable Base Scalar Mult. in $\mathcal{R}$

Input: $D \in \mathcal{R}$ and $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor E below $n\delta(D)$
Algorithm:

1. Set $E = D$
2. For $i = 1$ to l do
   - *// Double* Replace E by $E \oplus E$
   - *// Adjust* Replace E by $D(2\delta(E))$
   - If $b_i \neq 0$ then
     - If $b_i = 1$, set $D' = D$
     - If $b_i = -1$, set $D' = -D$
     - *// Add* replace E by $E \oplus D'$
     - *// Adjust* Replace E by $D(\delta(E) + \delta(D'))$
3. Output E

l doubles, $l/3$ adds, up to $(l + l/3) \cdot 2g = (8g/3)l$ baby steps

Fixed Base Scalar Mult. in $\mathcal{R}$

Fixed Base Scalar Mult. in $\mathcal{R}$

Input: $n \in \mathbb{N}$, $s = \lfloor n/(g+1) \rfloor$ in NAF

Fixed Base Scalar Mult. in $\mathcal{R}$

Input: $n \in \mathbb{N}$, $s = \lfloor n/(g+1) \rfloor$ in NAF

Output: The divisor $D(n) \in \mathcal{R}$ below n

Fixed Base Scalar Mult. in $\mathcal{R}$

Input: $n \in \mathbb{N}$, $s = \lfloor n/(g+1) \rfloor$ in NAF

Output: The divisor $D(n) \in \mathcal{R}$ below n

Algorithm:

1. Compute $E' = D(s(g+1)$ by calling the previous algorithm on inputs D_2 and s
2. Apply at most $n - s(g+1)$ baby steps to E' to compute $D(n)$
3. Output E

Fixed Base Scalar Mult. in $\mathcal{R}$

Input: $n \in \mathbb{N}$, $s = \lfloor n/(g+1) \rfloor$ in NAF

Output: The divisor $D(n) \in \mathcal{R}$ below n

Algorithm:

1. Compute $E' = D(s(g+1)$ by calling the previous algorithm on inputs D_2 and s
2. Apply at most $n - s(g+1)$ baby steps to E' to compute $D(n)$
3. Output E

- one integer division with remainder
- all the operations from previous algorithm
- at most g baby steps

Heuristics, Real Model

Heuristics, Real Model

Heuristics: with probability $1 - O(q^{-1})$:

Heuristics, Real Model

Heuristics: with probability $1 - O(q^{-1})$:

- $\delta_{i+1} - \delta_i \ = \ 1$ for $2 \leq i \leq r$

Heuristics, Real Model

Heuristics: with probability $1 - O(q^{-1})$:

- $\delta_{i+1} - \delta_i \;=\; 1$ for $2 \leq i \leq r$
- $\delta(D' \oplus D'') \;=\; \delta(D') + \delta(D'') - \lceil g/2 \rceil$

Heuristics, Real Model

Heuristics: with probability $1 - O(q^{-1})$:

- $\delta_{i+1} - \delta_i \; = \; 1$ for $2 \leq i \leq r$
- $\delta(D' \oplus D'') \; = \; \delta(D') + \delta(D'') - \lceil g/2 \rceil$

Consequences: with probability $1 - O(q^{-1})$:

Heuristics, Real Model

Heuristics: with probability $1 - O(q^{-1})$:

- $\delta_{i+1} - \delta_i \; = \; 1$ for $2 \leq i \leq r$
- $\delta(D' \oplus D'') \; = \; \delta(D') + \delta(D'') - \lceil g/2 \rceil$

Consequences: with probability $1 - O(q^{-1})$:

- If $D_i = (a_i, b_i)$ then

$$\deg(a_i) = \deg(b_{i+1} + b_i - h) - \deg(c_i) = (g+1) - 1 = g$$

Heuristics, Real Model

Heuristics: with probability $1 - O(q^{-1})$:

- $\delta_{i+1} - \delta_i \;=\; 1$ for $2 \le i \le r$
- $\delta(D' \oplus D'') \;=\; \delta(D') + \delta(D'') - \lceil g/2 \rceil$

Consequences: with probability $1 - O(q^{-1})$:

- If $D_i = (a_i, b_i)$ then

$$\deg(a_i) = \deg(b_{i+1} + b_i - h) - \deg(c_i) = (g+1) - 1 = g$$

- Relative distances (distance advancements) for both baby steps and giant steps are known and need no longer be kept track of

Improvements, Infrastructure

Improvements, Infrastructure

Variable Base Scenario

Improvements, Infrastructure

Variable Base Scenario

- Eliminate all adjustment steps, at the expense of $d = \lceil g/2 \rceil$ baby steps at the beginning (independent of the NAF length l of the scalars m, n)

Improvements, Infrastructure

Variable Base Scenario

- Eliminate all adjustment steps, at the expense of $d = \lceil g/2 \rceil$ baby steps at the beginning (independent of the NAF length l of the scalars m, n)

Fixed Base Scenario

Improvements, Infrastructure

Variable Base Scenario

- Eliminate all adjustment steps, at the expense of $d = \lceil g/2 \rceil$ baby steps at the beginning (independent of the NAF length l of the scalars m, n)

Fixed Base Scenario

- Replace all adds by baby steps

Improvements, Infrastructure

Variable Base Scenario

- Eliminate all adjustment steps, at the expense of $d = \lceil g/2 \rceil$ baby steps at the beginning (independent of the NAF length l of the scalars m, n)

Fixed Base Scenario

- Replace all adds by baby steps
- Eliminate all adjustment steps, at the expense of the following pre-computation ($g + 2$ baby steps, l doubles):
  - D^* with $\delta(D^*) = 2^l(g + 1) + g$
    - $d + 1$ baby steps applied to D_1 to obtain D_{d+2}
    - l doubles, starting with D_{d+2}: gets to distance $2^l(g + 1) + d$
    - $g - d$ baby steps
  - D_{d+3} with $\delta_{d+3} = d + g + 2$: one baby step from D_{d+2}

Improvements, Variable Base

Improvements, Variable Base

Input: $D \in \mathcal{R}$, $n = \sum b_i 2^{l-i}$ in NAF

Improvements, Variable Base

Input: $D \in \mathcal{R}$, $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor $E \in \mathcal{R}$ of distance $n\delta(D) + d$

Improvements, Variable Base

Input: $D \in \mathcal{R}$, $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor $E \in \mathcal{R}$ of distance $n\delta(D) + d$
Algorithm:

1. $E_1 = D$
2. For $i = 1$ to $d-1$ do   *//* $d-1$ *baby steps*

   Replace E_i by E_{i+1}
3. *// Now* $E_i = E_d$ Set $D' = E_i,\ \ D'' = E_{i+1},\ \ E = E_{i+1}$
4. For $i = 1$ to l do

   // Double   Replace E by $E \oplus E$

   // Add   If $b_i = 1$, replace E by $E \oplus D''$

   If $b_i = -1$ and g is even, replace E by $E \oplus \overline{D''}$

   If $b_i = -1$ and g is odd, replace E by $E \oplus \overline{D'}$
5. Output E

Improvements, Variable Base

Input: $D \in \mathcal{R}$, $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor $E \in \mathcal{R}$ of distance $n\delta(D) + d$
Algorithm:

1. $E_1 = D$
2. For $i = 1$ to $d-1$ do *// $d-1$ baby steps*

   Replace E_i by E_{i+1}
3. *// Now $E_i = E_d$* Set $D' = E_i, \; D'' = E_{i+1}, \; E = E_{i+1}$
4. For $i = 1$ to l do

   // Double Replace E by $E \oplus E$

   // Add If $b_i = 1$, replace E by $E \oplus D''$

   If $b_i = -1$ and g is even, replace E by $E \oplus \overline{D''}$

   If $b_i = -1$ and g is odd, replace E by $E \oplus \overline{D'}$
5. Output E

l doubles, $l/3$ adds, d baby steps

Improvements, Fixed Base

Improvements, Fixed Base

Pre-Computation: D^*, D_{d+3}

Improvements, Fixed Base

Pre-Computation: D^*, D_{d+3}

Input: $n = \sum b_i 2^{l-i}$ in NAF

Improvements, Fixed Base

Pre-Computation: D^*, D_{d+3}

Input: $n = \sum b_i 2^{l-i}$ in NAF

Output: The divisor $E = D(n) \in \mathcal{R}$ of distance n

Improvements, Fixed Base

Pre-Computation: D^*, D_{d+3}
Input: $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor $E = D(n) \in \mathcal{R}$ of distance n
Algorithm:

1. Set $E = D_{d+3}$
2. For $i = 1$ to l do

   // Double Replace E by $E \oplus E$

   // Baby Step If $b_i = 1$ then apply a baby step to E
   If $b_i = -1$ then apply a backward baby step to E

3. *// Now at distance* $2^{l+1} + n + d$
   Compute $D = E \oplus (-D^*)$
4. Output D

Improvements, Fixed Base

Pre-Computation: D^*, D_{d+3}
Input: $n = \sum b_i 2^{l-i}$ in NAF
Output: The divisor $E = D(n) \in \mathcal{R}$ of distance n
Algorithm:

1. Set $E = D_{d+3}$
2. For $i = 1$ to l do

   // Double Replace E by $E \oplus E$

   // Baby Step If $b_i = 1$ then apply a baby step to E
   If $b_i = -1$ then apply a backward baby step to E

3. *// Now at distance* $2^{l+1} + n + d$
   Compute $D = E \oplus (-D^*)$
4. Output D

l doubles, one add, $l/3$ baby steps

Analysis

Analysis

Operation Count	Doubles	Adds	Baby Steps
Imaginary	l	$l/3$	$-$
Real, Variable Base	l	$l/3$	d
Real, Fixed Base	l	1	$l/3$

Analysis

Operation Count	Doubles	Adds	Baby Steps
Imaginary	l	$l/3$	$-$
Real, Variable Base	l	$l/3$	d
Real, Fixed Base	l	1	$l/3$

Naive Analysis:

Analysis

Operation Count	Doubles	Adds	Baby Steps
Imaginary	l	$l/3$	$-$
Real, Variable Base	l	$l/3$	d
Real, Fixed Base	l	1	$l/3$

Naive Analysis:

- Real variable base scenario has about the same speed as imaginary model (neglecting the cost of baby steps)

Analysis

Operation Count	Doubles	Adds	Baby Steps
Imaginary	l	$l/3$	$-$
Real, Variable Base	l	$l/3$	d
Real, Fixed Base	l	1	$l/3$

Naive Analysis:

- Real variable base scenario has about the same speed as imaginary model (neglecting the cost of baby steps)
- Real model fixed base scenario is about 25 percent faster than imaginary model (factor 3/4)

Analysis

Operation Count	Doubles	Adds	Baby Steps
Imaginary	l	$l/3$	$-$
Real, Variable Base	l	$l/3$	d
Real, Fixed Base	l	1	$l/3$

Naive Analysis:

- Real variable base scenario has about the same speed as imaginary model (neglecting the cost of baby steps)
- Real model fixed base scenario is about 25 percent faster than imaginary model (factor 3/4)
- Key agreement is about 12.5 percent faster (factor 7/8)

Analysis

Operation Count	Doubles	Adds	Baby Steps
Imaginary	l	$l/3$	$-$
Real, Variable Base	l	$l/3$	d
Real, Fixed Base	l	1	$l/3$

Naive Analysis:

- Real variable base scenario has about the same speed as imaginary model (neglecting the cost of baby steps)
- Real model fixed base scenario is about 25 percent faster than imaginary model (factor 3/4)
- Key agreement is about 12.5 percent faster (factor 7/8)

Supported by numerical data, but not quite fair comparison: imaginary model could use a “point” divisor $D = P - \infty$ as base divisor (Katagi, Kitamura, Akishita & Takagi 2005)

Discrete Logarithm Problem

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Real Model – Infrastructure DLP

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Real Model – Infrastructure DLP

- Given a divisor $D \in \mathcal{R}$ and the divisor $E \in \mathcal{R}$ below $n\delta(D)$, find n

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Real Model – Infrastructure DLP

- Given a divisor $D \in \mathcal{R}$ and the divisor $E \in \mathcal{R}$ below $n\delta(D)$, find n
- Given the divisor $D(n) \in \mathcal{R}$ below n, find suitable n

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Real Model – Infrastructure DLP

- Given a divisor $D \in \mathcal{R}$ and the divisor $E \in \mathcal{R}$ below $n\delta(D)$, find n
- Given the divisor $D(n) \in \mathcal{R}$ below n, find suitable n
- Given a divisor $D \in \mathcal{R}$, find $\delta(D)$

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Real Model – Infrastructure DLP

- Given a divisor $D \in \mathcal{R}$ and the divisor $E \in \mathcal{R}$ below $n\delta(D)$, find n
- Given the divisor $D(n) \in \mathcal{R}$ below n, find suitable n
- Given a divisor $D \in \mathcal{R}$, find $\delta(D)$
- Given a reduced principal ideal in the coordinate ring of C, find a generator (*Principal Ideal Problem*)

Discrete Logarithm Problem

Imaginary Model – Jacobian DLP

- Given a reduced divisor D and the reduced divisor in the divisor class of nD, find n

Real Model – Infrastructure DLP

- Given a divisor $D \in \mathcal{R}$ and the divisor $E \in \mathcal{R}$ below $n\delta(D)$, find n
- Given the divisor $D(n) \in \mathcal{R}$ below n, find suitable n
- Given a divisor $D \in \mathcal{R}$, find $\delta(D)$
- Given a reduced principal ideal in the coordinate ring of C, find a generator (*Principal Ideal Problem*)

Security of both DLPs seems to be the same – exponential

Summary and Further Research

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

- NUCOMP – exact operation count and comparison with Cantor giant steps

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

- NUCOMP – exact operation count and comparison with Cantor giant steps
- Explicit formulas for low genus giant steps, real model

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

- NUCOMP – exact operation count and comparison with Cantor giant steps
- Explicit formulas for low genus giant steps, real model
- Explicit formulas for low genus giant steps, based on NUCOMP, both models

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

- NUCOMP – exact operation count and comparison with Cantor giant steps
- Explicit formulas for low genus giant steps, real model
- Explicit formulas for low genus giant steps, based on NUCOMP, both models
- Use the baby step giant step framework to to speed up the DLP in $\mathcal{R}$ or in $\mathcal{J}$? And to find R or $|\mathcal{J}|$?

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

- NUCOMP – exact operation count and comparison with Cantor giant steps
- Explicit formulas for low genus giant steps, real model
- Explicit formulas for low genus giant steps, based on NUCOMP, both models
- Use the baby step giant step framework to to speed up the DLP in $\mathcal{R}$ or in $\mathcal{J}$? And to find R or $|\mathcal{J}|$?
- Structural relationship between $\mathcal{R}$ and $\mathcal{J}$?

Summary and Further Research

Idea: Replace giant steps by baby steps where possible

Present and Future Work

- NUCOMP – exact operation count and comparison with Cantor giant steps
- Explicit formulas for low genus giant steps, real model
- Explicit formulas for low genus giant steps, based on NUCOMP, both models
- Use the baby step giant step framework to to speed up the DLP in $\mathcal{R}$ or in $\mathcal{J}$? And to find R or $|\mathcal{J}|$?
- Structural relationship between $\mathcal{R}$ and $\mathcal{J}$?
- Special types of curves?

Some General References

1. H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen and F. Vercouteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman & Hall/CRC, Boca Raton (Florida), 2006
2. M. J. Jacobson, Jr., A. J. Menezes and A. Stein, Hyperelliptic curves and cryptography, in *High Primes and Misdemeanors: Lectures in Honour of the 60th Birthday of Hugh Cowie Williams*, *Fields Institute Communications* **41**, American Mathematical Society, Providence (Rhode Island) 2004, 255-282
3. M. J. Jacobson, Jr., R. Scheidler and A. Stein, Cryptographic protocols on real hyperelliptic curves. *Advances in Mathematics of Communications* **1** (2007), 197-221
4. M. J. Jacobson, Jr., R. Scheidler and A. Stein, Fast arithmetic on hyperelliptic curves via continued fraction expansions. *Advances in Coding Theory and Cryptology*, *Series on Coding Theory and Cryptology* **2**, World Scientific Publishing Co. Pte. Ltd., Hackensack (New Jersey) 2007, 201-244
5. A. J. Menezes, Y.-H. Wu and R. J. Zuccherato, An elementary introduction to hyperelliptic curves, in *Algebraic Aspects of Cryptography*, *Algorithms and Computation in Mathematics* **3**, Springer, Berlin (Germany) 1998, 155-178
6. R. Scheidler, A. Stein and H. C. Williams, Key exchange in real quadratic congruence function fields. *Designs, Codes and Cryptography* **7** (1996), 153-174