The discrete logarithm problem and its application in Cryptography

Roger Oyono

University of French Polynesia, Tahiti

Lecture in Cryptography for Master class
Madrid, April 2008

Discrete logarithm problem (DLP) (1)

Two main problems on which public key cryptography is based:

- integer factorisation (in RSA).
- DLP (ElGamal Cryptosystem, Diffie-Hellman key exchange):

  Let G be a cyclic finite abelian group and $g \in G$ be a generator of G. The discrete logarithm problem (DLP) in G is the following:

  Given an element $h \in G$, find the smallest positive integer x such that

  $$h = [x]g \ \text{(additive group)} \quad / \quad h = g^x \ \text{(multiplicative group)} .$$

  We will denote such an x with $DL_g(h)$.

Discrete logarithm problem (DLP) (2)

As we will see later, a cryptographically suitable group G must satisfy the following conditions:

- representation is easy and compact.
- fast arithmetic.
- DLP is computationally hard.
- group order can be computed efficiently.

CDHP and DDHP

- The computational Diffie-Hellman Problem (CDHP) is the problem:

$$\text{Given } g, h_x = [x]g \text{ and } h_y = [y]g\,, \text{ compute } [xy]g\,.$$

- The resolution of the DLP implies the resolution of the CDHP.
- The decisional Diffie-Hellman Problem (CDHP) is the problem:

$$\text{Given } g,\, h_x = [x]g,\, h_y = [y]g \text{ and } h_z = [z]g\,, \text{ decide if } h_z = [xy]g.$$

- There are groups G for which DDH is easier than CDLP or DLP, but we do not know how to answer this question in general.

Overview (1)

- Efficient scalar multiplication
- Solving the DLP in generic groups
  - Pohlig-Hellman
  - Shanks' Baby step - Giant step
  - Pollard rho
- Cryptographic protocols based on the DLP
  - Key exchange
  - Encryption
  - Signature
  - Security: what is a cryptographically secure group?

Overview (2)

- Subexponential algorithms for the DLP in finite (prime) fields
  - Generalities
  - Smooth numbers, factor base and subexponentiality
  - Adleman's algorithm
- Elliptic curves
  - Generalities
    - Why interesting?
    - Group Law
  - DLP on "special elliptic curves"
- Hyper- and Non-hyperelliptic curves
  - Generalization: Abelian varieties and Jacobian varieties
    - Generalities
    - Why interesting?
  - Group law on Hyperelliptic Jacobians (of small genus)
  - Group law on non-hyperelliptic Jacobians (of small genus)
  - Index calculus

Scalar multiplication using binary left to right (1)

Algorithm (binary left to right (1))

IN: $P \in G$ et $n \in \mathbb{N}$

$n = (n_{l-1} \dots n_0), n_{l-1} = 1.$

OUT: $[n]P \in G$.

1. $R \leftarrow P$
2. $\texttt{for } i = l - 2 \texttt{ to } 0 \texttt{ do}$
    1. $R \leftarrow [2]R$
    2. $\texttt{if } n_i = 1 \texttt{ then } R \leftarrow R \oplus P$
    3. $i \leftarrow i - 1$
3. $\texttt{return } R$

cost: $O(\log n)$ doublings /additions in the group G.

Example: binary left to right (2)

The above algorithm is based on the binary expansion of the scalar n:

$$[(n_{l-1} \dots n_0)_2]P = [2]([(n_{l-1} \dots n_1)_2]P) \oplus [n_0]P$$

Example: $45 = (101101)_2$

P
$2P$
$2(2P) \oplus P$
$2(2(2P) \oplus P) \oplus P$
$2(2(2(2P) \oplus P) \oplus P)$
$2(2(2(2(2P) \oplus P) \oplus P)) \oplus P \quad = [45]P$

Generic groups (1)

A generic group is a group where we can only:

- Represent group elements (uniquely)
- Apply the group operation to a pair of elements to obtain a new element

The representation of the group elements gives us no information on the structure of the group.

The group operation may be done using an oracle.

Most groups are not generic groups, but we can look at them as generic groups if we ”forget” the extra information...

Algorithms for solving the DLP for generic groups give us an upper bound on how hard things are!

Generic groups (2)

In generic groups, we have three methods to compute $DL_g(h)$:

- Baby step - Giant step (Shanks)
- Pollard ρ
- Pollard kangaroo

and one more method to take advantage of the decomposition of the group order

- Pohlig-Hellman

Idea: Non trivial subgroups can make the DLP easier!

Suppose the additive cyclic group $G = \langle g \rangle$ has order

$$N = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$$

If we know $DL_g(h)$ modulo $p_i^{\alpha_i}$ for every i, then we can compute $DL_g(h)$ via the Chinese remainder theorem.

From the group order, we have:

$$G \simeq G_1 \times G_2 \times \cdots \times G_k$$

with

$$G_i \simeq \mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}$$

Subgroups

We can restrict the DLP from G to G_i:

Define $g_i = \left[\frac{N}{p_i^{\alpha_i}}\right] g$ and $h_i = \left[\frac{N}{p_i^{\alpha_i}}\right] h$.
We can compute $DL_{g_i}(h_i)$ in a group of order $p_i^{\alpha_i}$ (instead of N).

We have

$$DL_{g_i}(h_i) \equiv \frac{DL_g(h_i)}{DL_g(g_i)} \equiv \frac{DL_g(\left[\frac{N}{p_i^{\alpha_i}}\right] h)}{DL_g(\left[\frac{N}{p_i^{\alpha_i}}\right] g)} \equiv \frac{\left[\frac{N}{p_i^{\alpha_i}}\right] DL_g(h)}{\left[\frac{N}{p_i^{\alpha_i}}\right] DL_g(g)} \equiv DL_g(h),$$

and g_i has order $p_i^{\alpha_i}$, so

$$DL_g(h) \equiv DL_{g_i}(h_i) \bmod p_i^{\alpha_i}.$$

Prime Powers

Assume now that $G = \langle g \rangle \simeq \mathbb{Z}/p^\alpha\mathbb{Z}$ and $h \in G$. For $DL_g(h) = x$, write

$$x = x_0 + x_1 p + x_2 p^2 + \ldots + x_{\alpha-1} p^{\alpha-1} \pmod{p^\alpha}$$

with $x_i \in [0, p-1]_\mathbb{Z}$.

Let $g' = [p^{\alpha-1}]g$, then g' has order p and the equality $[x]g = h$ becomes:

$$[x_0]g' = [x]g' = [p^{\alpha-1}]h$$

x_0 can be find by computing $DL_{g'}([p^{\alpha-1}]h)$ in $\langle g' \rangle$ (a subgroup of order p). We also compute x_1 via a DLP in $\langle g' \rangle$:

$$[x_1]g' = p^{\alpha-2}([-x_0]g + h)$$

We iterate this approach to compute $x_2, x_3, \ldots, x_{\alpha-1}$ and thus x.

Pohlig-Hellman: An example

Consider a finite abelian group G of order

$$\#G = 2^{29}3^{21}5^{14}7^{5}11^{9}101^{3}$$

$\#G$ is a 160 bits number ...

Using Pohlig-Hellman with a exhaustive search for the discrete log on the (sub)groups of prime order, we can solve the DLP in less than 3000 group operations.

That's less than the cost of 12.5 scalar multiplications!

Shanks' Baby step - Giant step

Let $G = \langle g \rangle$, and n a good upper bound of $\#G$. Let $u \approx \sqrt{n}$.

Considering the u-adic expansion of $x = DL_g(h)$

$$x = x_0 + ux_1, \text{ with } x_i \in [0, u-1],$$

we get

$$[x]g = h \Longleftrightarrow [x_1]([u]g) = h - [x_0]g\,.$$

To solve the DLP in G:

- We construct the list

$$S = \{h, h-[g], h-[2]g, \ldots, h-[u-1]g\} \quad \text{(Baby step)}$$

- We compute succesively the values $[x_1]([u]g)$ for $x_1 = 0, 1, \ldots$ and stop when such an element belongs to S (Giant step).

Cost

- We have u Baby steps, each taking 1 group operation.
- Computing $[u]g$ takes $O(\log u)$ group operations.
- We have u Giant steps, each taking 1 group operation.
- The total cost is $u + u + O(\log u)$, which is $O(\sqrt{n})$.
- The memory requirements is also $O(\sqrt{n})$.

Let G be a finite group of order N (in practice $G = \langle g \rangle$).

- A random map is a function $F : G \longrightarrow G$ such that the image of $x \in G$ is choosen (uniformly) at random in G.
- A random walk in G is a sequence of elements of G, starting at x_0, such that $x_{i+1} = F(x_i)$. The sequence $x_0, x_1, x_2, \ldots$ is eventually periodic (G is finite). We are interested in the value of i for which the first repetition occurs.
- Claim: The average time for the first repetition is $\sqrt{\pi/2}\sqrt{N}$.
- Proof: Starting from x_0, choose the image of x_i at random the first time you see x_i. The first repetition occurs at the first time when your random choice is an element that was chosen at a previous step. Use the Birthday Paradox.

Pollard ρ

Once again, we want to compute $DL_g(h)$ for $h \in G = \langle g \rangle$, a group of prime order N.

If we define

$$F(x) = [\alpha_x]g + [\beta_x]h \,,$$

and $x_0 = [\alpha_0]g + [\beta_0]h$ for randomly choosen $\alpha_x, \beta_x, \alpha_0$ and β_0, then the the first repetition (the point where we close the loop) gives us a relation of the form

$$[\alpha_i]g + [\beta_i]h = [\alpha_j]g + [\beta_j]h$$

Pollard ρ

We group the $g's$ and $h's$ together, and we get:

$$[\beta_i - \beta_j]h = [\alpha_j - \alpha_i]g \, .$$

With a little bit of luck, $\gcd(N, \beta_i - \beta_j) = 1$, and we have

$$DL_g(h) \equiv (\alpha_j - \alpha_i)/(\beta_i - \beta_j) \pmod{N} \, .$$

The expected time for the algorithm is $O(\sqrt{N})$.

But in this form, the algorithm has memory $O(\sqrt{N})$...

Although, it is possible to reduce the memory complexity to $O(1)$ using distinguished points and pseudo-Random walks (Floyd's method for cycles detection).

Principal goals of the Cryptography

- Historically, the most important goal of the cryptography was to secure private communication (Encryption).
- Nowadays, there are other goals
  - authentification
  - non-repudiation
  - integrity

The discover of public key cryptography provides methods to realize the above goals:

- asymmetric encryption
- Signature
- Key exchange (for session key in symmetric encryptions)
- electronic voting, etc ...

Diffie-Hellman Key exchange

Let $G = \langle g \rangle$ be a finite abelian cyclic group of order N.

Alice	unsecure channel	Bob
choose $x_A \in_R [1, N]$		
compute $k_A := [x_A]g$	$\longrightarrow k_A$	
		choose $x_B \in_R [1, N]$
	$k_B \longleftarrow$	compute $k_B := [x_B]g$
compute $k_{AB} := [x_A]k_B$		compute $k_{AB} := [x_B]k_A$

Massey-Omura encryption

Let G be a finite cyclic group of prime order N. We consider message (to encrypt) as elements m of G.

Alice	unsecure channel	Bob
choose $x_A \in_R [1, N]$		
compute $a := [x_A]m$	$\longrightarrow a$	
		choose $x_B \in_R [1, N]$
		compute
	$b \longleftarrow$	$b := [x_B]a = [x_A x_B]m$
compute		compute
$a' := [x_A^{-1}]b = [x_B]m$	$\longrightarrow a'$	$b' := [x_B^{-1}]a' = m$

- This encryption scheme is purely from theoretical interest (pedagogic).

  It is more convenient to generate a session key (via Diffie-Hellman) for a use in a symmetric encryption (hybrid encryption).
- Principle: Both users are concerned to encrypt a message m.
- Crucial point: the encryption in probabilistic.

ElGamal Encryption

- public parameters: A finite cyclic group $G = \langle g \rangle$.
- Bob's public key: $h = [x]g$
- Bob's private key: x
- To encrypt a message $m \in G$ that Alice want to send to Bob, Alice use the public key h of Bob and choose $k \in_R [1, N-1]$ to compute
$$a = [k]g\,, \text{ and } b = [k]h + m\,.$$
- Alice send (a, b) to Bob.
- Bob can recover the message by computing
$$b - [x]a = [k]h + m - [kx]g = [kx]g - [kx]g + m = m\,.$$

- public parameters: A finite cyclic group $G = \langle g \rangle$.
- Bob's public key: $h = [x]g$
- Bob's private key: x
- Hypothesis: There is a (public fonction) $f : G \longrightarrow \mathbb{Z}/N\mathbb{Z}$.
- To sign a message $m \in [1, N-1]$, Bob choose $k \in_R [1, N-1]$ to compute $a = [k]g$.
- Bob compute $b \in \mathbb{Z}/N\mathbb{Z}$ with

$$m \equiv xf(a) + bk \pmod{N}.$$

- Bob send the message m and its signature $s = (a, b)$ to Alice.
- Alice accepts the signature if

$$[f(a)]h + [b]a = [xf(a) + kb]g = [m]g\,.$$

The security of those protocols depends on

- The choice of the (pseudo-) random generators
- The problem of distribution of public key's (PKI)
- The choice of hash fonction
- Hardware attacks, etc ...

Furthermore, for those simple protocols, we do not know if their security is equivalent to the DLP (but for CDHP).

Suitable groups

A cryptographically suitable group G must satisfy:

- Representation of its elements in an easy and compact way.
- Fast arithmetic, i.e. fast scalar multiplication.
- DLP is computationally hard, in best case only the generic methods works.

Consequence of Pohlig-Hellman reduction: It is important to know the group order, or better to compute it efficiently. Furthermore, the value or this order is used in some protocols.

The minimal amount of computations that we suppose infeasible is $\approx 2^{80}$.

$\Longrightarrow$ The cardinality of the group order should have at least a 160-prime factor to avoid the generic attacks.

- Prime fields: $q = p$
  - Multiplication: product of two integers, and reduction modulo p.
  - Inverse: extended euclidian algorithm.
- Finite fields of characteristic 2 ;

$$\mathbb{F}_2[x]/(f(x)) = \left\{ \sum_{i=0}^{n-1} c_i x^i \ : \ c_i \in \mathbb{F}_2, 0 \leq i < n \right\}.$$

  - Multiplication : product of polynomials with coefficients in $\mathbb{F}_2$, and reduction modulo the defining polynomial $f(x)$.
  - Inverse: extended euclidian algorithm for polynomials.

$\Longrightarrow$ Extremly efficient arithmetic on those finite fields.

Index calculus attacks in prime fields

- Index calculus is a method to compute discrete logarithms, also called indices.
- p prime, elements of $\mathbb{F}_p$ represented by numbers in $\{0, 1, \ldots, p-1\}$; g generator of multiplicative group.
- If $h \in \mathbb{F}_p$ factors as $h = h_1 \cdot h_2 \cdots h_n$ then

$$h = g^{a_1} \cdot g^{a_2} \cdots g^{a_n} = g^{a_1 + a_2 + \cdots + a_n}$$

  with $h_i = g^{a_i}$.
- Knowledge of the a_i, i.e. the discrete logarithms of h_i to base g gives knowledge of the discrete logarithm of h to base g.
- If h factors appropriately ...

An integer is said to be B-smooth if its decomposition in prime factors only contains primes $p \leq B$.

To evaluate the proportion of smooth numbers, we introduce the function

$$\phi(x,y) = \#\{1 \leq n \leq x \,;\, n \text{ is } y-\text{smooth}\}\,.$$

For $y = 23$ we obtain the following proportions:

x	100	1000	10000	100000
$\frac{\phi(x,y)}{x}$	76 %	37 %	14 %	4 %

Definition: subexponential functions

- Let $N > 0, 0 \leq \alpha \leq 1, c > 0$.

$$L_N(\alpha, c) := \exp\left(c(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right)$$

- If $\alpha = 0$, then $L_N(\alpha, c) = (\log N)^c$: polynomial in the length of N.
- If $\alpha = 1$, then $L_N(\alpha, c) = \exp c(\log N) = N^c$: exponential in the length of N.
- We say that $L_N(\alpha, c)$ is subexponential if $0 < \alpha < 1$.

N.B.: There exists algorithms for the "special" integer factorization ($n = p \cdot q$) with a subexponential running time: the fastest known method is the Number field sieve with time complexity

$$O\left(\exp\left((1.923 + o(1))(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}\right)\right)$$

where $o(1) = \theta(n) \longrightarrow 0$ for $n \longrightarrow +\infty$.

Theorem fundamental

For any $c > 0$, when $x \longrightarrow +\infty$, then

$$\frac{\phi(x, L_x(\frac{1}{2}, c))}{x} \sim \frac{1}{\sqrt{L_x(\frac{1}{2}, \frac{1}{c})}} \sim \frac{1}{L_x(\frac{1}{2}, \frac{1}{2c})}$$

Adleman's algorithm in prime fields

Let p a prime number, g a generator of $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*, h \in \langle g \rangle$.

- Choice of the "factors base":
  - Bound of smoothness B,
  - $\mathcal{F}_B = \{p_i, p_i \text{ prime }, p_i < B\}$.
  - How to compute the $DL_g(p_i)$ for the $p_i \in \mathcal{F}_B$? ($p_i = g^{DL_g(p_i)}$)
- Find "some relations":
  - For a random $r \in_R [0, p-2]$, compute $g^r \pmod p$.
  - If the obtained number is B-smooth, it gives "a relation"

$$g^r = \prod_{p_i \in \mathcal{F}_B} p_i^{\alpha_i} = \prod_{p_i \in \mathcal{F}_B} g^{DL_g(p_i)\alpha_i} = g^{\sum_{p_i \in \mathcal{F}_B} DL_g(p_i)\alpha_i}$$

    such that $r \equiv \sum_{p_i \in \mathcal{F}_B} DL_g(p_i)\alpha_i \pmod{p-1}$.
  - Iterate the last step to get at least $\#\mathcal{F}_B$ relations.

- Lineare algebra:
  - We have a linear system (in the unknown $DL_g(p_i)$) with more equations than unknown. We solve it to obtain $DL_g(p_i)$ for all p_i.
  - This step needs to be done only once per field and generator, it does not depend on the target DLP $h = g^x$.
- Solving the original DLP:

  How now to solve the DLP for $h \in \langle g \rangle$, i.e. how to compute $DL_g(h)$?
  Choose randomly $r \in [1, p-2]$ until $g^r \cdot h \pmod p$ is B-smooth.
  Then,

$$g^r \cdot h = \prod_{p_i \in \mathcal{F}_B} p_i^{\beta_i} \text{ and thus } DL_g(h) = \sum_{p_i \in \mathcal{F}_B} DL_g(p_i)\beta_i - r\,.$$

Analysis of Adleman's algorithm

Principle

It is much easier to find some relation if B is large, however we then need much more relation (since $\mathcal{F}_B$ will be large too)!

We will choose B to be of the form

$$B = L_p\left(\frac{1}{2}, \rho\right).$$

From the smoothness theorem, the probability that a random element in $\mathbb{F}_p^*$ is B-smooth is

$$\mathbb{P} = \frac{1}{L_p\left(\frac{1}{2}, \frac{1}{2\rho}\right)}.$$

- The average time we will need to find the $\#\mathcal{F}_B$ relation is:

$$L_p\left(\frac{1}{2},\frac{1}{2\rho}\right)\cdot L_p\left(\frac{1}{2},\rho\right)=L_p\left(\frac{1}{2},\rho+\frac{1}{2\rho}\right).$$

- Linear algebra: The matrix representing the linear sytem is sparse ($O(\log p)$ non zero terms in each row). We can then use adequate algorithms with quadratic (in the length of the matrix) running time.

  The cost of the linear algebra is:

$$L_p\left(\frac{1}{2},\rho\right)^2=L_p\left(\frac{1}{2},2\rho\right).$$

Analysis of Adleman's algorithm

- The cost of the final step (the smoothness relation of $g^r \cdots$) is equivalent to the cost of one smoothness relation.
- The total cost of the algorithm is

$$L_p\left(\frac{1}{2}, 2\rho\right) + L_p\left(\frac{1}{2}, \rho + \frac{1}{2\rho}\right) = L_p\left(\frac{1}{2}, \max\left(2\rho, \rho + \frac{1}{2\rho}\right)\right).$$

- The optimal value is obtained when $\rho = \frac{1}{\sqrt{2}}$, which gives the complexity

$$L_p(\frac{1}{2}, \sqrt{2}).$$

- Running time with much more clever way of finding relations is

$$O\left(\exp\left((1.923 + o(1))(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}}\right)\right)$$

$\cdots$ in characetrsitic 2

Let $q = 2^n$. The field with q elements $\mathbb{F}_q$ is isomorphic to

$$\mathbb{F}_2[x]/(f(x)) = \left\{ \sum_{i=0}^{n-1} c_i x^i \; : \; c_i \in \mathbb{F}_2, 0 \leq i < n \right\}.$$

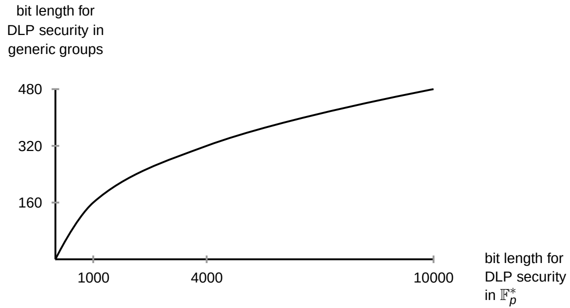
where $f \in \mathbb{F}_2[x]$ is an irreducible polynomial of degree n.
Adleman's algorithm can be trivially extended to such fields :

- Factoring into powers of small primes is replaced by factoring into irreducible polynomials of small degree.
- Same approach works, same problem of balancing size of factorbase (and thus complexity of the matrix step) and the likelihood of splitting completely over the factors base.

Cryptographic interests

Best known attack for $G = \mathbb{F}_q^* : L_q(\frac{1}{3}, c)$
Best known attack for generic groups: $2^{n/2}$

For the same security level, the bit length of the group order of generic groups beahves like the cubic root of the bit length of $\#\mathbb{F}_q^*$



Let $K = \mathbb{F}_q$ be the finite field with q elements. An elliptic curve over K is given by a non-singular equation

$$(1) \quad E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

where $a_i \in K$. For a field extension L of K, the set of rational points of E is

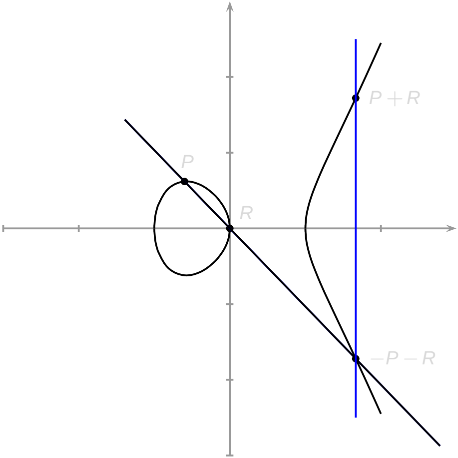
$$E(L) := \left\{(x,y) \in L^2 : (x,y) \text{ satisfy } (1)\right\} \cup \{\mathcal{O}\},$$

where $\mathcal{O}$ denotes the point at infinity.

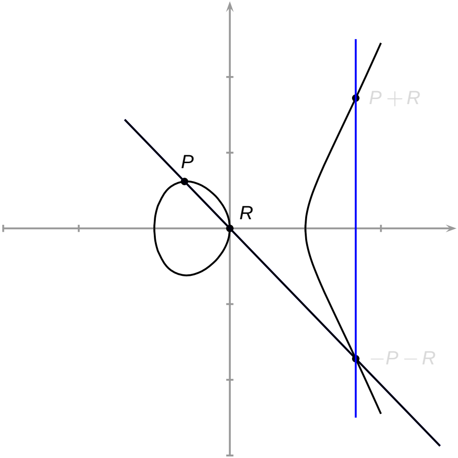
A point of E is an element of $E(\bar{K})$ where $\bar{K}$ is the algebraic closure of K.

For any extension L of K, the set $E(L)$ forms an abelian group with identity element $\mathcal{O}$.

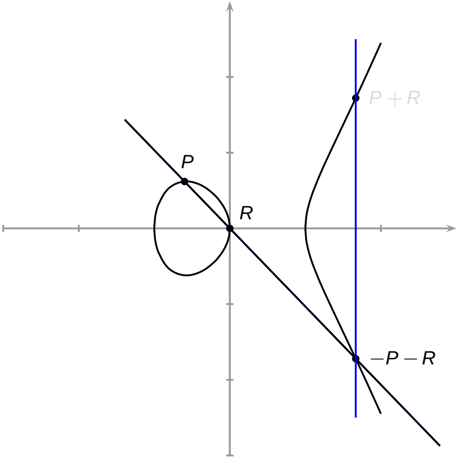
$E : y^2 = x^3 - x$



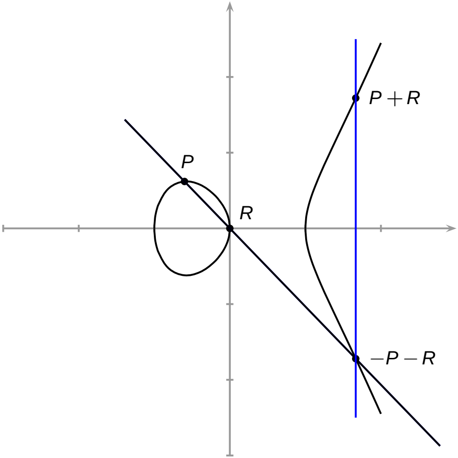
$E : y^2 = x^3 - x$



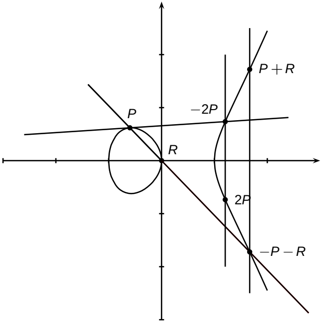
$$E : y^2 = x^3 - x$$



$E : y^2 = x^3 - x$



$$E : y^2 = x^3 + a_4 x + a_6,\ a_i \in \mathbb{F}_q$$



Pour $(x_1, y_1) \neq (x_2, -y_2)$:

$$(x_1, y_1) \oplus (x_2, y_2) = (x_3, y_3)$$
$$= (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1),$$

avec

$$\lambda = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{si } x_1 \neq x_2, \\ (3x_1^2 + a_4)/(2y_1) & \text{si } x_1 = x_2 \end{cases}$$

$\Rightarrow$ Addition and Doubling differ considerably.:

1 I, 2M, 1S vs. 1 I, 2M, 2S

Projective Coordinates

$P = (X_1 : Y_1 : Z_1)$, $Q = (X_2 : Y_2 : Z_2)$, $P \oplus Q = (X_3 : Y_3 : Z_3)$ on
$E : Y^2Z = X^3 + a_4XZ^2 + a_6$

Addition: $P \neq \pm Q$ $A = Y_2Z_1 - Y_1Z_2, B = X_2Z_1 - X_1Z_2$
$C = A^2Z_1Z_2 - B^3 - 2B^2X_1Z_2$
$X_3 = BC, Z_3 = B^3Z_1Z_2$
$Y_3 = A(B^2X_1Z_2 - C) - B^3Y_1Z_2$

Doubling: $P = Q \neq -P$
$A = a_4Z_1^2 + 3X_1^2, B = Y_1Z_1,$
$C = X_1Y_1B, D = A^2 - 8C$
$X_3 = 2BD, Z_3 = 8B^3.$
$Y_3 = A(4C - D) - 8Y_1^2B^2$

No inversion is needed and the computation times are $12M + 2S$ for a general addition and $7M + 5S$ for a doubling.

système	points	correspondence
affine ($\mathcal{A}$)	(x, y)	
projective ($\mathcal{P}$)	(X, Y, Z)	$(X/Z, Y/Z)$
jacobi ($\mathcal{J}$)	(X, Y, Z)	$(X/Z^2, Y/Z^3)$
Chudnovsky jacobi ($\mathcal{J}^C$)	(X, Y, Z, Z^2, Z^3)	$(X/Z^2, Y/Z^3)$
jacobi modifié ($\mathcal{J}^m$)	(X, Y, Z, aZ^4)	$(X/Z^2, Y/Z^3)$

système	addition			doublements		
affine ($\mathcal{A}$)	2M	1S	1I	2M	2S	1I
projective ($\mathcal{P}$)	12M	2S	–	7M	5S	–
jacobi ($\mathcal{J}$)	12M	4S	–	4M	6S	–
Chudnovsky jacobi ($\mathcal{J}^C$)	11M	3S	–	5M	6S	–
jacobi modifié ($\mathcal{J}^m$)	13M	6S	–	4M	4S	–

New efficient and "complete" formulae using Edward's model for elliptic curves: $\Longrightarrow$ Lange & Berstein's talks in two weeks

Number of points

Hasse's theorem

In cryptograhy, we usually consider elliptic curves over finite fields $\mathbb{F}_q$.

The number of $\mathbb{F}_q$-rational points of E is also finite, a bound is given by Hasse's theorem:

$$\#E(\mathbb{F}_q) = q + 1 - t,$$

with $|t| \leq 2\sqrt{q}$. The integer t is called the trace of E.

DLP on special elliptic curves

For a "generic" elliptic curve, the best known attack is Pollard ρ (combined with Pohlig-Hellman).

$\Longrightarrow$ Elliptic curves behave like generic groups.

Although, there are some classes of specific curves with much faster attack :

- MOV Reduction
- Anomalous curves
- Curves with non-trivial automorphisms group
- Weil descent

Definition

Let G a subgroup of $E(\mathbb{F}_q)$ of prime order $N | \# E(\mathbb{F}_q)$. The MOV degree is the smallest integer k such that $N | q^k - 1$.

Theorem (Menezes-Okamoto-Vanstone, Frey-Rück)

The DLP in G can be reduced to the DLP in $\mathbb{F}_{q^k}^*$.

Idea of the proof: Use the Weil pairing to embedd G in $\mathbb{F}_{q^k}$. ($\Longrightarrow$ Galbraith's lectures on pairing in June).

Remark: The DLP can be solved in a subexponential running time in $\mathbb{F}_{q^k}$. However, for a random elliptic curve E, k is very large!

For elliptic curves with trace $t = 0$, we then have $\# E(\mathbb{F}_p) = p + 1 | p^2 - 1$ and thus $k = 2$. Supersingular elliptic curves over prime fields are thus less suitable for DLP based cryptography .

Weil descent and anomalous curves

Weil descent

In some case, the DLP in $E(\mathbb{F}_{2^n})$ can be reduced in a DLP of an hyperelliptic curve of large genus over a smaller field.

We will see that there exists subexponential attacks for large genus curves (last lecture ”maybe”).

The curves defined over $E(\mathbb{F}_{2^n})$ where n is composite are in danger regarding this attack.

An anomalous elliptic curve is a curve over $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = p$, such that $\#E(\mathbb{F}_p) \simeq (\mathbb{F}_p, +)$.

Theorem (Smart, Satoh-Araki, Semaev)

The above isomorphism can be given explicitly.

The DLP on such groups can be computed very efficiently.

Standards

- ANSI Public Key Cryptography for the Financial Services Industry
  - X9.62-1998 – The Elliptic Curve Digital Signature Algorithm (ECDSA)
  - X9.63-1999 – Key Agrrement and Key Transport Using Elliptic Curve Cryptography (ECIES etc.)
- NIST – FDigital Signature Standard FIPS 186-2 (revision 2000)
- IEEE P1363a – Standart Specifications for Public Key Cryptography
- Standarts for Efficient Cryptography Group (Certicom)
- ISO 15946

Abelian varieties

The natural generalization of elliptic curves to higher dimension are abelian varieties.

DLP on an abelian variety over a finite field seems to be hard in general.

Problem: difficult to obtain explicit examples.

$\Rightarrow$ *Jacobian varieties of algebraic curves*.

The natural generalization of elliptic curves to higher dimension are abelian varieties.

DLP on an abelian variety over a finite field seems to be hard in general.

Problem: difficult to obtain explicit examples.

$\Rightarrow$ *Jacobian varieties of algebraic curves*.

Let $C : f(x,y) = 0$ be an algebraic curve defined over a field K, and let L be an extension of K

- Rational points of C :

$$C(L) := \{(x,y) \in L^2 : f(x,y) = 0\}$$

- The points of C are the elements of $C(\bar{K})$
- Let $K = \mathbb{F}_q$ a finite field. The Frobenuis of $\mathbb{F}_q$: $x \longmapsto x^q$ induces a morphism of C via

$$P = (x,y) \longmapsto P^q := (x^q, y^q)$$

Then

$$C(\mathbb{F}_{q^n}) = \{P \in C(\overline{F_q}) | P^{q^n} = P\}$$

Let k be a field and C an algebraic complete curve defined over k, $g := g(C)$ its genus.

Definition

C is said to be *hyperelliptic* if there exists a morphism $\varphi : C \longrightarrow \mathbb{P}^1$ of degree 2.

Explicit model

Every hyperelliptic curve C/k admits a non-singular affine model

$$y^2 + h(x)y = f(x)$$

with $\deg(f) \in \{2g+2, 2g+1\}$ and $\deg(h) \leq g$.

Let k be a field and C an algebraic complete curve defined over k, $g := g(C)$ its genus.

Definition

C is said to be hyperelliptic if there exists a morphism $\varphi : C \longrightarrow \mathbb{P}^1$ of degree 2.

Explicit model

Every hyperelliptic curve C/k admits a non-singular affine model

$$y^2 + h(x)y = f(x)$$

with $\deg(f) \in \{2g+2, 2g+1\}$ and $\deg(h) \leq g$.

Let k be a field and C an algebraic complete curve defined over k, $g := g(C)$ its genus.

Definition

C is said to be hyperelliptic if there exists a morphism $\varphi : C \longrightarrow \mathbb{P}^1$ of degree 2.

Explicit model

Every hyperelliptic curve C/k admits a non-singular affine model

$$y^2 + h(x)y = f(x)$$

with $\deg(f) \in \{2g+2, 2g+1\}$ and $\deg(h) \leq g$.

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Example of hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 12:12:12 [Seed =1234567890]
   Type ? for help.  Type <Ctrl>-D to quit.

> A<x,y> := AffineSpace(GF(7),2);
> C1 := Curve(A, y^2-(x^7+x-1));
> Genus(C1);
3
> Points(C1);
{@ (1, 1), (1, 6), (4, 0), (5, 3), (5, 4), (6, 2), (6, 5) @}
> P<X,Y,Z> := ProjectiveSpace(GF(7),2);
> C2:=Curve(P, Z^5*Y^2-(X^7+X*Z^6-Z^7));
> Genus(C2);
3
> Points(C2);
@ (1 : 1 : 1), (1 : 6 : 1), (4 : 0 : 1), (5 : 3 : 1), (5 : 4 : 1), (6 : 2 : 1),
(6 : 5 : 1), (0 : 1 : 0) @}
> SingularPoints(C2);
@ (0 : 1 : 0) @}
```

Definition

A non-hyperelliptic curve C is a curve for which there exists no morphism $C \longrightarrow \mathbb{P}^1$ of degree 2.

Canonical embedding

Let $\{\omega_1, \cdots, \omega_g\}$ a basis of $\Omega^1(C)$. The curve C is non-hyperelliptic iff the canonical morphism

$$\begin{array}{rccl} \varphi : & C & \longrightarrow & \mathbb{P}^{g-1} \\ & P & \longmapsto & \varphi(P) := (\omega_1(P), \ldots, \omega_g(P)), \end{array}$$

is an embedding.

In this case, $\varphi(C)$ is a degree $2g-2$ curve of genus g.

Definition

A non-hyperelliptic curve C is a curve for which there exists no morphism $C \longrightarrow \mathbb{P}^1$ of degree 2.

Canonical embedding

Let $\{\omega_1, \cdots, \omega_g\}$ a basis of $\Omega^1(C)$. The curve C is non-hyperelliptic iff the canonical morphism

$$\begin{array}{rccl} \varphi : & C & \longrightarrow & \mathbb{P}^{g-1} \\ & P & \longmapsto & \varphi(P) := (\omega_1(P), \ldots, \omega_g(P)), \end{array}$$

is an embedding.

In this case, $\varphi(C)$ is a degree $2g-2$ curve of genus g.

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Example of non-hyperelliptic curves

```
Magma V2.14-1   Mon Feb 19 2007 15:15:22 [Seed =3629778794]
   Type ? for help.  Type <Ctrl>-D to quit.

> P<X,Y,Z> := ProjectiveSpace(Rationals(),2);
> C1 := Curve(P,X^7 + X^3*Y^2*Z^2 + Z^7);
> Genus(C1);
3
> phi1 := CanonicalMap(C1,P);
> phi(C1);
Curve over Rational Field defined by -X*Z + Y^2
> C2 := Curve(P,Y^3*Z^2-X*(X-Z)*(X-2*Z)*(X-3*Z)^2);
> Genus(C2);
3
> phi2 := CanonicalMap(C2,P);
> phi2(C2);
Curve over Rational Field defined by
X^3*Y - 6*X^3*Z - Y^3*Z + 12*Y^2*Z^2 - 47*Y*Z^3 + 60*Z^4
```

Equations for non-hyperelliptic curves

Petri's theorem (1923) gives an explicit description of the image of a non-hyperelliptic curve under the canonical embedding.

Theorem

Let C be a curve of genus $g \geq 1$ defined over a field k s.t. $C(k) \neq \emptyset$. Then, there exists a g dimensional abelian variety $\mathrm{Jac}(C)$ (the jacobian of C) and a morphism (both defined over k)

$$\Phi : C \longrightarrow \mathrm{Jac}(C)$$

with the universal property:

Let $h : C \longrightarrow A$ a morphism of C in an abelian variety A. Then there exist an homomorphism $\alpha : \mathrm{Jac}(C) \longrightarrow A$ and an element $a \in A$, s.t. $h(x) = \alpha(\Phi(x)) + a$ for all $x \in C$.

Theorem

Let C be a curve of genus $g \geq 1$ defined over a field k s.t. $C(k) \neq \emptyset$. Then, there exists a g dimensional abelian variety $\mathrm{Jac}(C)$ (the jacobian of C) and a morphism (both defined over k)

$$\Phi : C \longrightarrow \mathrm{Jac}(C)$$

with the universal property:

Let $h : C \longrightarrow A$ a morphism of C in an abelian variety A. Then there exist an homomorphism $\alpha : \mathrm{Jac}(C) \longrightarrow A$ and an element $a \in A$, s.t. $h(x) = \alpha(\Phi(x)) + a$ for all $x \in C$.

- A divisor on C is a formal sum $D = \sum_P n_P P$ (almost all $n_P = 0$) where $P \in C(\overline{\mathbb{F}}_q)$.

  Examples:

$$\begin{array}{lcl} D_1 & = & P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 301P_5 \\ D_2 & = & -7P_1 - 301P_5 + 101Q_1 + Q_2 - 3Q_3 \\ D_1 + D_2 & = & -6P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 101Q_1 + Q_2 - 3Q_3, \end{array}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(C)$.
- A divisor D is effectif ($D \geq 0$) if $n_P \geq 0$ for all P.
- $\mathrm{Supp}(D) := \{P \in C(\overline{\mathbb{F}}_q) : n_P \neq 0\}$.
- Degree of a divisor: $\deg(D) := \sum_P n_P$.

Divisors

- A divisor on C is a formal sum $D = \sum_P n_P P$ (almost all $n_P = 0$) where $P \in C(\overline{\mathbb{F}}_q)$.
  Examples:

$$
\begin{aligned}
D_1 &= P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 301P_5 \\
D_2 &= -7P_1 - 301P_5 + 101Q_1 + Q_2 - 3Q_3 \\
D_1 + D_2 &= -6P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 101Q_1 + Q_2 - 3Q_3,
\end{aligned}
$$

- The set of all divisors forms an abelian group $\mathrm{Div}(C)$.
- A divisor D is effectif ($D \geq 0$) if $n_P \geq 0$ for all P.
- $\mathrm{Supp}(D) := \{P \in C(\overline{\mathbb{F}}_q) : n_P \neq 0\}$.
- Degree of a divisor: $\deg(D) := \sum_P n_P$.

- A divisor on C is a formal sum $D = \sum_P n_P P$ (almost all $n_P = 0$) where $P \in C(\overline{\mathbb{F}}_q)$.
  Examples:

$$
\begin{array}{lcl}
D_1 & = & P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 301P_5 \\
D_2 & = & -7P_1 - 301P_5 + 101Q_1 + Q_2 - 3Q_3 \\
D_1 + D_2 & = & -6P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 101Q_1 + Q_2 - 3Q_3,
\end{array}
$$

- The set of all divisors forms an abelian group $\mathrm{Div}(C)$.
- A divisor D is effectif ($D \geq 0$) if $n_P \geq 0$ for all P.
- $\mathrm{Supp}(D) := \{P \in C(\overline{\mathbb{F}}_q) : n_P \neq 0\}$.
- Degree of a divisor: $\deg(D) := \sum_P n_P$.

- A divisor on C is a formal sum $D = \sum_P n_P P$ (almost all $n_P = 0$) where $P \in C(\overline{\mathbb{F}}_q)$.
  Examples:

$$\begin{array}{lcl}
D_1 & = & P_1 + 2P_2 + 3P_3 - 10^{121} P_4 + 301 P_5 \\
D_2 & = & -7P_1 - 301 P_5 + 101 Q_1 + Q_2 - 3Q_3 \\
D_1 + D_2 & = & -6P_1 + 2P_2 + 3P_3 - 10^{121} P_4 + 101 Q_1 + Q_2 - 3Q_3,
\end{array}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(C)$.
- A divisor D is effectif ($D \geq 0$) if $n_P \geq 0$ for all P.
- $\mathrm{Supp}(D) := \{P \in C(\overline{\mathbb{F}}_q) : n_P \neq 0\}$.
- Degree of a divisor: $\deg(D) := \sum_P n_P$.

- A divisor on C is a formal sum $D = \sum_P n_P P$ (almost all $n_P = 0$) where $P \in C(\overline{\mathbb{F}}_q)$.
  Examples:

$$\begin{array}{lcl}
D_1 & = & P_1 + 2P_2 + 3P_3 - 10^{121}P_4 + 301P_5 \\
D_2 & = & -7P_1 \qquad\qquad\qquad\qquad\qquad - 301P_5 + 101Q_1 + Q_2 - 3Q_3 \\
D_1 + D_2 & = & -6P_1 + 2P_2 + 3P_3 - 10^{121}P_4 \qquad\qquad + 101Q_1 + Q_2 - 3Q_3,
\end{array}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(C)$.
- A divisor D is effectif ($D \geq 0$) if $n_P \geq 0$ for all P.
- $\mathrm{Supp}(D) := \{P \in C(\overline{\mathbb{F}}_q) : n_P \neq 0\}$.
- Degree of a divisor: $\deg(D) := \sum_P n_P$.

- A divisor D is defined over $\mathbb{F}_q$ if $D = D^{\sigma}$ for any $\sigma \in \mathrm{Gal}(\overline{\mathbb{F}}_q, \mathbb{F}_q)$.

  Examples:

  Let $P_1, \cdots, P_6$ points of the curve $C/\mathbb{F}_q$ s.t.

  - $P_1, P_2, P_3 \in C(\mathbb{F}_q)$
  - $P_4 \in C(\mathbb{F}_{q^2}) - C(\mathbb{F}_q)$
  - $P_5 \in C(\mathbb{F}_{q^3}) - C(\mathbb{F}_q)$

  Then, the following divisors are $\mathbb{F}_q$-rational

$$D_1 := P_1, D_2 := P_1 + P_2, D_3 := P_4 + P_4^q, D_4 := P_5 + P_5^q + P_5^{q^2}.$$

  The divisors $D_6 := P_4$, $D_7 := P_5 + P_5^q$ are not $\mathbb{F}_q$-rational.

- $\mathrm{Div}_{\mathbb{F}_q}(C)$ is a subgroup of $\mathrm{Div}(C)$.

Divisors

- A divisor D is defined over $\mathbb{F}_q$ if $D = D^\sigma$ for any $\sigma \in \mathrm{Gal}(\overline{\mathbb{F}}_q, \mathbb{F}_q)$.

  Examples:

  Let $P_1, \cdots, P_6$ points of the curve $C/\mathbb{F}_q$ s.t.

  - $P_1, P_2, P_3 \in C(\mathbb{F}_q)$
  - $P_4 \in C(\mathbb{F}_{q^2}) - C(\mathbb{F}_q)$
  - $P_5 \in C(\mathbb{F}_{q^3}) - C(\mathbb{F}_q)$

  Then, the following divisors are $\mathbb{F}_q$-rational

  $$D_1 := P_1, D_2 := P_1 + P_2, D_3 := P_4 + P_4^q, D_4 := P_5 + P_5^q + P_5^{q^2}.$$

  The divisors $D_6 := P_4$, $D_7 := P_5 + P_5^q$ are not $\mathbb{F}_q$-rational.

- $\mathrm{Div}_{\mathbb{F}_q}(C)$ is a subgroup of $\mathrm{Div}(C)$.

Divisors

- A divisor D is defined over $\mathbb{F}_q$ if $D = D^\sigma$ for any $\sigma \in \mathrm{Gal}(\overline{\mathbb{F}}_q, \mathbb{F}_q)$.

  Examples:

  Let $P_1, \cdots, P_6$ points of the curve $C/\mathbb{F}_q$ s.t.

  - $P_1, P_2, P_3 \in C(\mathbb{F}_q)$
  - $P_4 \in C(\mathbb{F}_{q^2}) - C(\mathbb{F}_q)$
  - $P_5 \in C(\mathbb{F}_{q^3}) - C(\mathbb{F}_q)$

  Then, the following divisors are $\mathbb{F}_q$-rational

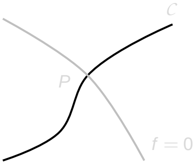
  $$D_1 := P_1, D_2 := P_1 + P_2, D_3 := P_4 + P_4^q, D_4 := P_5 + P_5^q + P_5^{q^2}.$$

  The divisors $D_6 := P_4$, $D_7 := P_5 + P_5^q$ are not $\mathbb{F}_q$-rational.

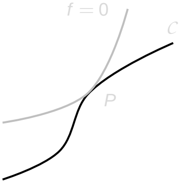
- $\mathrm{Div}_{\mathbb{F}_q}(C)$ is a subgroup of $\mathrm{Div}(C)$.

For a function $f \in \overline{\mathbb{F}}_q(C)^*$ we associate the principal divisor (f) defined by

$$(f) = \sum_P v_P(f)P$$



$v_P(f) = 1$

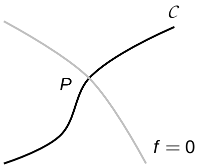


$v_P(f) \geq 1$

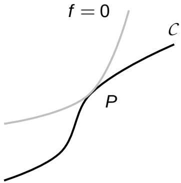
Principal divisors

For a function $f \in \overline{\mathbb{F}}_q(C)^*$ we associate the principal divisor (f) defined by

$$(f) = \sum_P v_P(f)P$$



$\nu_P(f) = 1$

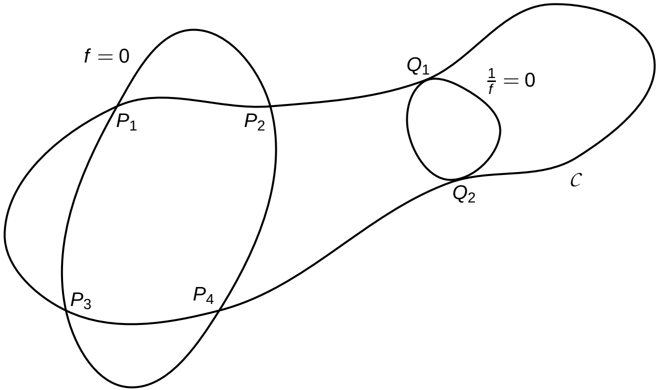


$\nu_P(f) \geq 1$

Principal divisors: an example

The following divisor is principal:

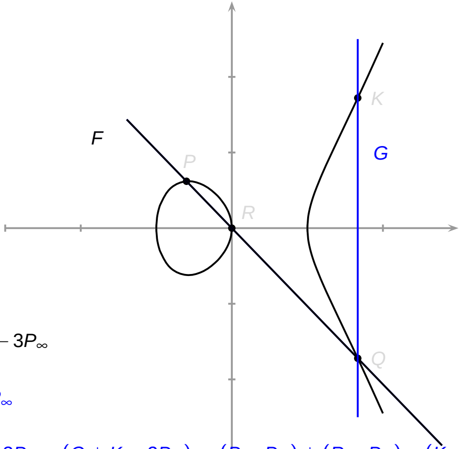
$(f) = P_1 + P_2 + P_3 + P_4 - (2Q_1 + 2Q_2)$



Linear equivalence

- The principal divisor (f) describe the zeros and poles (with multiplicities) of f.
- (f) is defined over $\mathbb{F}_q$ iff f is defined over $\mathbb{F}_q$.
- Any principal divisor is of degree 0.
- The set $\mathrm{Princ}(C)$ of principal divisors forms a subgroup of the set $\mathrm{Div}^0(C)$ of all divisors of degree zero.
- Two divisors D_1 and D_2 are said to be equivalent if they differ from a principal divisor. Write $D_1 \sim D_2$.

$$E\ : y^2 = x^3 - x$$



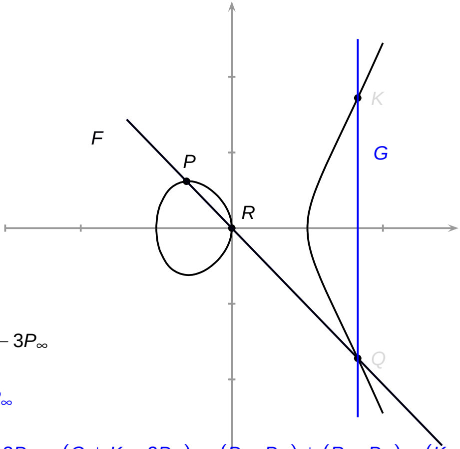
$(F(x,y)) = P + Q + R - 3P_\infty$

$(G(x,y)) = Q + K - 2P_\infty$

$\left(\frac{F(x,y)}{G(x,y)}\right) = P + Q + R - 3P_\infty - (Q + K - 2P_\infty) = (P - P_\infty) + (R - P_\infty) - (K - P_\infty)$

and thus $(P - P_\infty) + (R - P_\infty) \sim (K - P_\infty)$

$$E : y^2 = x^3 - x$$



$(F(x,y)) = P + Q + R - 3P_\infty$

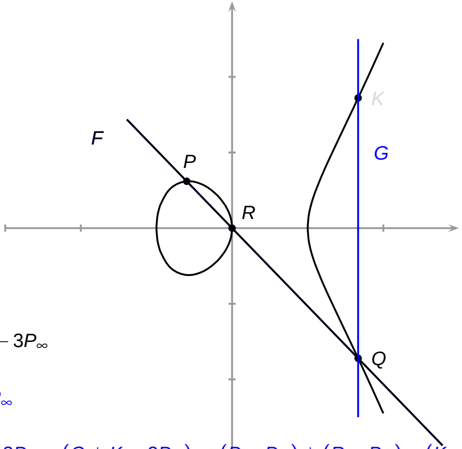
$(G(x,y)) = Q + K - 2P_\infty$

$(\frac{F(x,y)}{G(x,y)}) = P + Q + R - 3P_\infty - (Q + K - 2P_\infty) = (P - P_\infty) + (R - P_\infty) - (K - P_\infty)$

and thus $(P - P_\infty) + (R - P_\infty) \sim (K - P_\infty)$

Jacobian of elliptic curves and group law

$$E\;:\; y^2 = x^3 - x$$



$(F(x,y)) = P + Q + R - 3P_\infty$

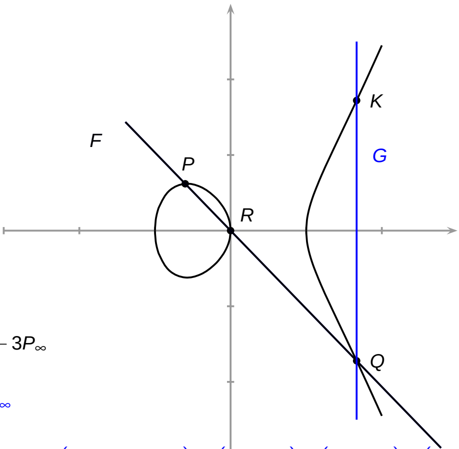
$(G(x,y)) = Q + K - 2P_\infty$

$(\frac{F(x,y)}{G(x,y)}) = P + Q + R - 3P_\infty - (Q + K - 2P_\infty) = (P - P_\infty) + (R - P_\infty) - (K - P_\infty)$

and thus $(P - P_\infty) + (R - P_\infty) \sim (K - P_\infty)$

Jacobian of elliptic curves and group law

$$E : y^2 = x^3 - x$$



$(F(x,y)) = P + Q + R - 3P_\infty$

$(G(x,y)) = Q + K - 2P_\infty$

$\left(\frac{F(x,y)}{G(x,y)}\right) = P + Q + R - 3P_\infty - (Q + K - 2P_\infty) = (P - P_\infty) + (R - P_\infty) - (K - P_\infty)$

and thus $(P - P_\infty) + (R - P_\infty) \sim (K - P_\infty)$

Picard group

- $\mathrm{Pic}^0_{\mathbb{F}_q}(C)$ is the quotient group of $\mathrm{Div}^0_{\mathbb{F}_q}(C)$ by the subgroup of principal divisors.
- Call this the divisor class group (or Picard group).
- $\mathrm{Pic}^0_{\mathbb{F}_{q^l}}(C)$ is isomorphic to the group of $\mathbb{F}_{q^l}$-valued points of the Jacobian $\mathrm{Jac}(C)$ of C.
- $dim(\mathrm{Jac}(C)) = g_C$.
- Weil's theorem implies

$$|\#C(\mathbb{F}_q) - (q+1)| \leq 2g\sqrt{q},$$

$$(\sqrt{q}-1)^{2g} \leq \#\mathrm{Jac}(C)(\mathbb{F}_q) \leq (\sqrt{q}+1)^{2g},$$

in particular $\#\mathrm{Jac}(C)(\mathbb{F}_q) \approx q^g$.

Suitable representations for divisor classes (?)

Let $P_1, \cdots, P_{100}$ points of the curve $C/_{\mathbb{F}_q}$. Let

$$D := P_1 + P_2 + \cdots + P_{99} - 99P_{100}$$

a degree zero divisor with support

$$\mathrm{Supp}(D) = \{P_1, \cdots, P_{100}\}$$

Is it possible to find a divisor $D' \sim D$ with less points on its support?

Answer: YES, it is even possible to have $\#\mathrm{Supp}(D') = \mathrm{MIN}$.

Idea: **Use the Riemann Roch theorem.**

Let $P_1, \cdots, P_{100}$ points of the curve $C/_{\mathbb{F}_q}$. Let

$$D := P_1 + P_2 + \cdots + P_{99} - 99P_{100}$$

a degree zero divisor with support

$$\mathrm{Supp}(D) = \{P_1, \cdots, P_{100}\}$$

Is it possible to find a divisor $D' \sim D$ with less points on its support?

Answer: YES, it is even possible to have $\#\mathrm{Supp}(D') = \mathrm{MIN}$.

Idea: **Use the Riemann Roch theorem.**

Reduced divisors

Theorem

Let C/k be a hyperelliptic curve of genus g, P_∞ a fixed k-rational point of C. For a k-rational divisor D of degree 0, there exists a unique positive divisor E of minimal degree $m \leq g$, with $P_\infty \notin \mathrm{Supp}(E)$, such that

$$D \sim E - mP_\infty$$

The divisor E is called a (reduced divisor), and m its weight.

Goal

Given two reduced divisors $D_1 - n_1P_\infty$ et $D_2 - n_2P_\infty$, compute the reduced representative $D^+ - n_3P_\infty$ of the formal sum $(D_1 - n_1P_\infty) + (D_2 - n_2P_\infty)$.

Reduced divisors

Theorem

Let C/k be a hyperelliptic curve of genus g, P_∞ a fixed k-rational point of C. For a k-rational divisor D of degree 0, there exists a unique positive divisor E of minimal degree $m \leq g$, with $P_\infty \notin \mathrm{Supp}(E)$, such that

$$D \sim E - mP_\infty$$

The divisor E is called a (reduced divisor), and m its weight.

Goal

Given two reduced divisors $D_1 - n_1P_\infty$ et $D_2 - n_2P_\infty$, compute the reduced representative $D^+ - n_3P_\infty$ of the formal sum $(D_1 - n_1P_\infty) + (D_2 - n_2P_\infty)$.

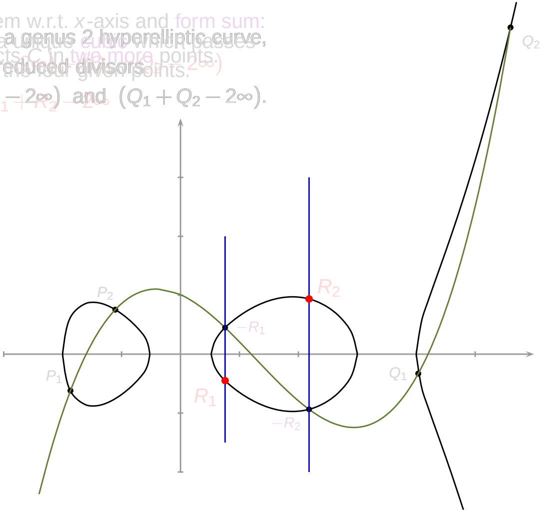
How to do HECC arithmetic?

Consider a genus 2 hyperelliptic curve, with two reduced divisors
$(P_1 + P_2 - 2\infty)$ and $(Q_1 + Q_2 - 2\infty)$.

There is a unique cubic which passes through the four given points.

It intersects C in two more points.

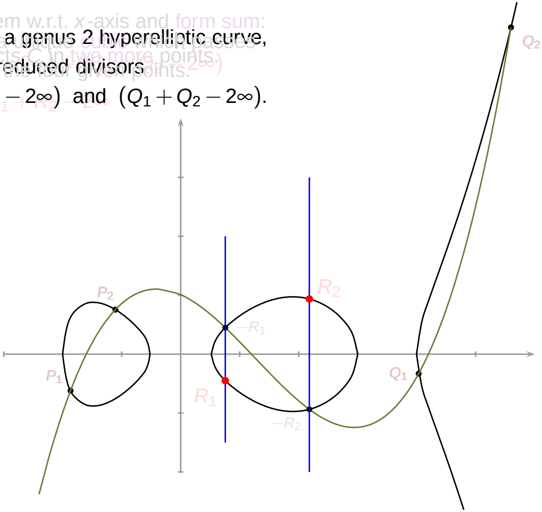
Mirror them w.r.t. x-axis and form sum:
$(P_1 + P_2 - 2\infty) + (Q_1 + Q_2 - 2\infty)$
$\sim R_1 + R_2 - 2\infty$



How to do HECC arithmetic?

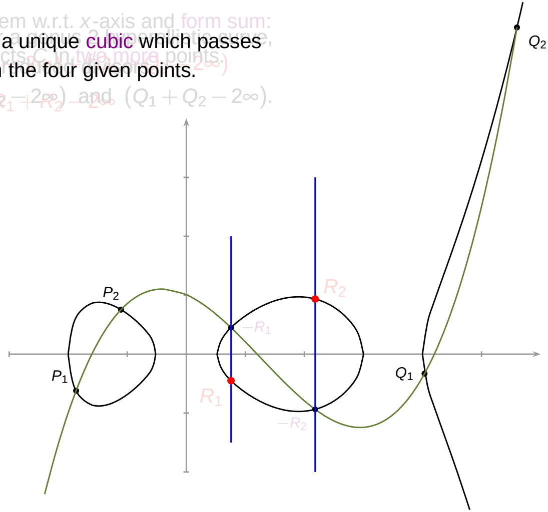
Consider a genus 2 hyperelliptic curve, with two reduced divisors

$(P_1 + P_2 - 2\infty)$ and $(Q_1 + Q_2 - 2\infty)$.



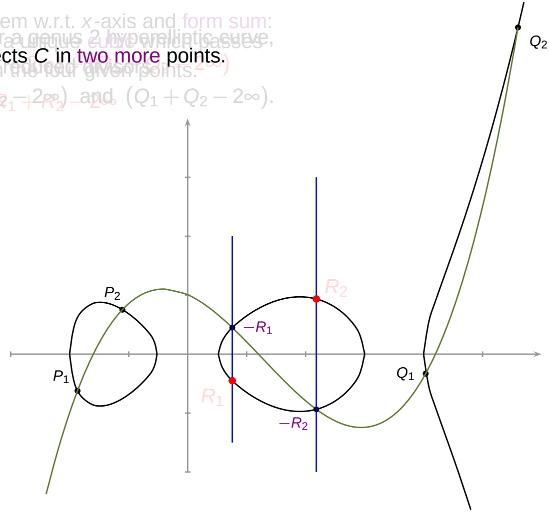
How to do HECC arithmetic?

There is a unique cubic which passes through the four given points.



How to do HECC arithmetic?

It intersects C in two more points.

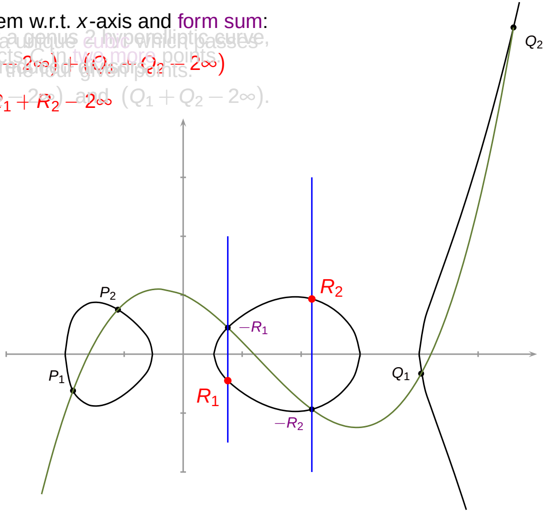


How to do HECC arithmetic?

Mirror them w.r.t. x-axis and form sum:

$(P_1 + P_2 - 2\infty) + (Q_1 + Q_2 - 2\infty)$

$\sim R_1 + R_2 - 2\infty$



Mumford representation

- For hyperelliptic curves $y^2 + h(x)y = f(x)$, Mumford proposed a unique (compact) representation of reduced divisors by a pair of two polynomials u, v s.t.

$$\begin{cases} u, v \in \mathbb{F}_q[x], \\ u \text{ monic}, \\ \deg_x v < \deg_x u \leq g, \\ u(x) \text{ divides } v(x)^2 + h(x)v(x) - f(x). \end{cases}$$

$P_i = (x_i, y_i) \in \mathrm{Supp}_{[u,v]} \Leftrightarrow u(x_i) = 0, v(x_i) = y_i$ with multiplicity .

- Arithmetic uses *"only"* arithmetic on polynomials ... but is far less efficient than on elliptic curves (if applied directly).

Cantor's algorithm

Algorithm Composition & Reduction (Cantor/Koblitz)

INPUT: $D_1 = [u_1, v_1]$, $D_2 = [u_2, v_2]$ and $C : y^2 + h(x)y = f(x)$
OUTPUT: $D_1 + D_2 = [u_{D_1+D_2}, v_{D_1+D_2}]$

1. Compute $d_1 = \gcd(u_1, u_2) = e_1 u_1 + e_2 u_2$
2. Compute $d = \gcd(d_1, v_1 + v_2 + h) = c_1 d_1 + c_2(v_1 + v_2 + h)$
3. Let $s_1 = c_1 e_1, s_2 = c_1 e_2, s_3 = c_2$
4. $u = \frac{u_1 u_2}{d^2}$ $\quad v = \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f)}{d} \bmod u$
5. The result $[u, v]$ corresponds to a semi reduced divisor.
6. Let $u' = \frac{f - vh - v^2}{u}$ $\quad v' = (-h - v) \bmod u'$
7. **if** $\deg u' > g$ alors $u := u', v := v'$ **goto** step 5
8. Make u monic
9. The result $[u, v]$ corresponds to a reduced divisor.

genus 2

2000	Harley (car. impaire)	$\Rightarrow$ 2 inv.
2001	Lange (car. arbitraire)	
2001	Matsuo, Chao, Tsujii (...)	
2002	Miyamoto, Doi, Matsuo, Chao, Tsujii	$\Rightarrow$ 1 inv.
2002	Takahashi	
2002	Lange (car. arbitraire)	
2002	Sugizaki, Matsuo, Chao, Tsujii (car. paire)	

genus 3

2002	Kuroki, Gonda, Matsuo, Chao, Tsujii	$\Rightarrow$ 1 inv.
2002	Pelzl, Guyot & Patankar	

Addition, $g = 2$ (Lange)

Addition, $\deg u_1 = \deg u_2 = 2$		
Input Output	$[u_1, v_1], [u_2, v_2], u_i = x^2 + u_{i1}x + u_{i0}, v_i = v_{i1}x + v_{i0}$ $[u', v'] = [u_1, v_1] + [u_2, v_2]$	
Step	Expression	Operations
1	Computation of the resultant r of u_1, u_2: $z_1 = u_{11} - u_{21}, z_2 = u_{20} - u_{10}, z_3 = u_{11}z_1 + z_2;$ $r = z_2 z_3 + z_1^2 u_{10};$	1S, 3M
2	Compute the "almost inverse" of u_2 modulo u_1 ($inv = r/u_2 \bmod u_1$): $inv_1 = z_1, inv_0 = z_3;$	
3	Compute $s' = rs \equiv (v_1 - v_2)inv \bmod u_1$: $w_0 = v_{10} - v_{20}, w_1 = v_{11} - v_{21}, w_2 = inv_0 w_0, w_3 = inv_1 w_1;$ $s'_1 = (inv_0 + inv_1)(w_0 + w_1) - w_2 - w_3(1 + u_{11}), s'_0 = w_2 - u_{10}w_3;$	5M
4	Compute $s'' = x + s_0/s_1 = x + s'_0/s'_1$ et s_1: $w_1 = (rs'_1)^{-1}(= 1/r^2 s_1), w_2 = rw_1(= 1/s'_1), w_3 = s'^2_1 w_1(= s_1);$ $w_4 = rw_2(= 1/s_1), w_5 = w_4^2, s''_0 = s'_0 w_2;$	I, 2S, 5M
5	Compute $l' = s'' u_2 = x^3 + l'_2 x^2 + l'_1 x + l'_0$: $l'_2 = u_{21} + s''_0, l'_1 = u_{21}s''_0 + u_{20}, l'_0 = u_{20}s''_0$	2M
6	Compute $u' = (s(l + h + 2v_2) - k)/u_1 = x^2 + u'_1 x + u'_0$: $u'_0 = (s''_0 - u_{11})(s''_0 - z_1 + h_2 w_4) - u_{10} + l'_1 + (h_1 + 2v_{21})w_4 + (2u_{21} + z_1 - f_4)w_5;$ $u'_1 = 2s''_0 - z_1 + h_2 w_4 - w_5;$	3M
7	Compute $v' \equiv -h - (l + v_2) \bmod u' = v'_1 x + v'_0$: $w_1 = l'_2 - u'_1, w_2 = u'_1 w_1 + u'_0 - l'_1, v'_1 = w_2 w_3 - v_{21} - h_1 + h_2 u'_1;$ $w_2 = u'_0 w_1 - l'_0, v'_0 = w_2 w_3 - v_{20} - h_0 + h_2 u'_0;$	4M
total		I, 3S, 22M

Equations for genus 3 non-hyperelliptic curves

- $\varphi(C)$ is a smooth plane quartic,
- Conversely, any nonsingular quartic curve C in $\mathbb{P}^2(k)$ is a canonical embedding of a non-hyperelliptique curve of genus 3.

Reduced divisors

Theorem

Let C/k be a non-singular curve of genus g and D^∞ be an effective k-rational divisor of degree g. Then every divisor class has a representative of the form

$$E - D^\infty$$

where E is an effective k-rational divisor of degree g. Generically, the divisor E is unique.

Goal

For two reduced divisors $D_1 - D^\infty$ and $D_2 - D^\infty$, compute the reduced representative $D^+ - D^\infty$ of $(D_1 - D^\infty) + (D_2 - D^\infty)$.

Theorem

Let C/k be a non-singular curve of genus g and D^∞ be an effective k-rational divisor of degree g. Then every divisor class has a representative of the form

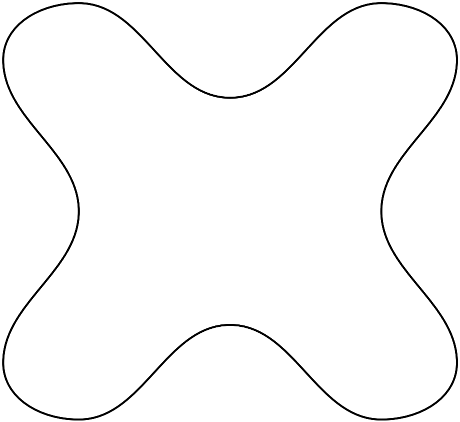
$$E - D^\infty$$

where E is an effective k-rational divisor of degree g. Generically, the divisor E is unique.

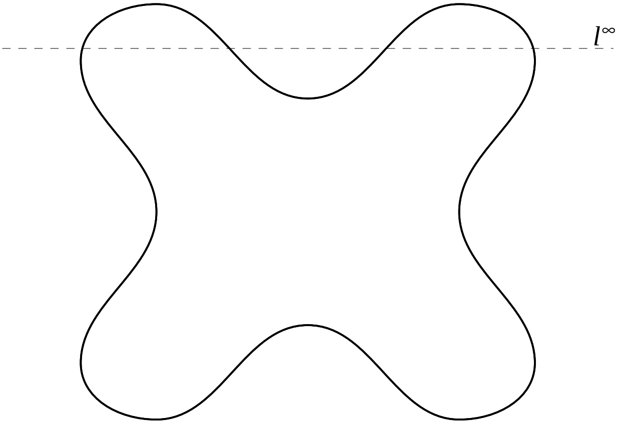
Goal

For two reduced divisors $D_1 - D^\infty$ and $D_2 - D^\infty$, compute the reduced representative $D^+ - D^\infty$ of $(D_1 - D^\infty) + (D_2 - D^\infty)$.

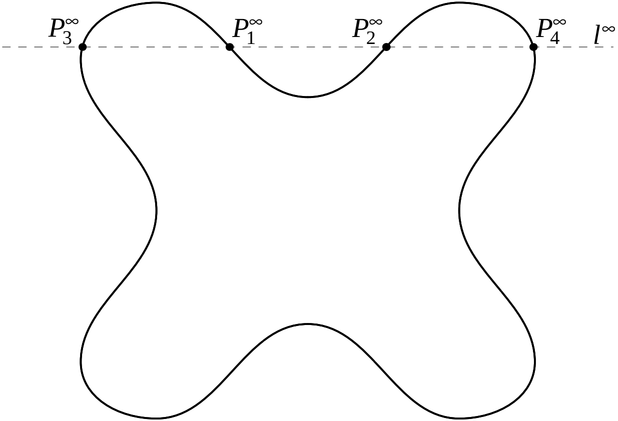
Group law for non-hyperelliptic curves, $g = 3$



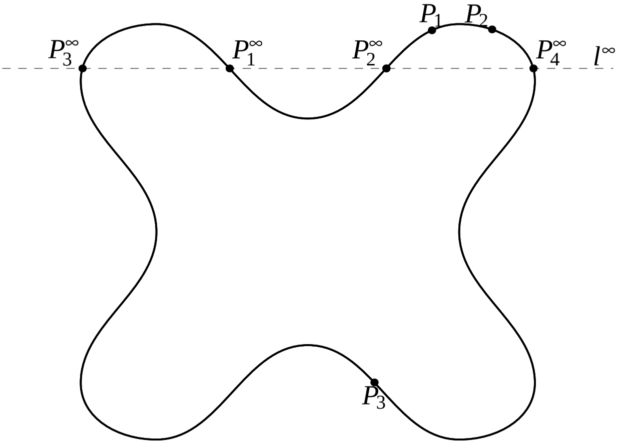
Group law for non-hyperelliptic curves, $g = 3$



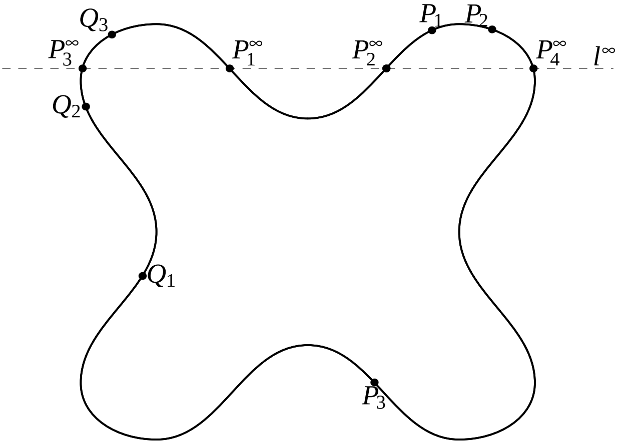
Group law for non-hyperelliptic curves, $g = 3$



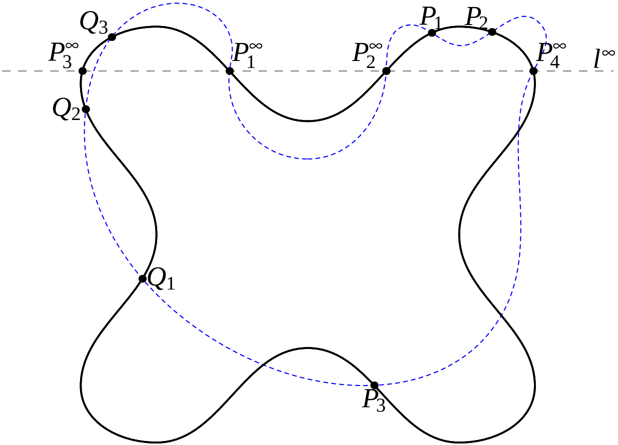
Group law for non-hyperelliptic curves, $g = 3$



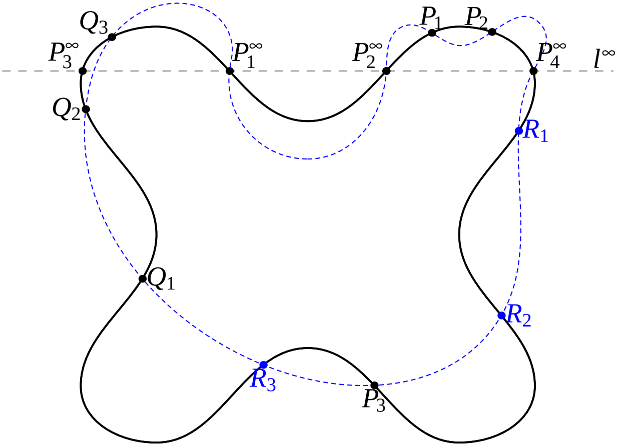
Group law for non-hyperelliptic curves, $g = 3$



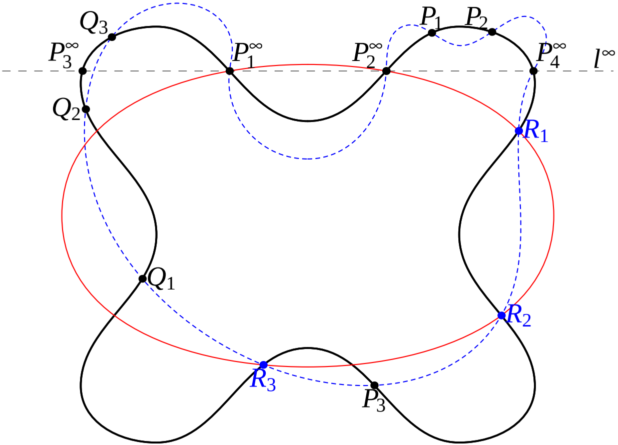
Group law for non-hyperelliptic curves, $g = 3$

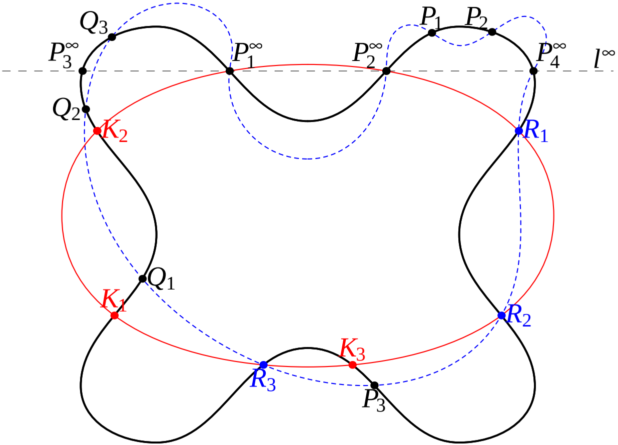


Group law for non-hyperelliptic curves, $g = 3$



Group law for non-hyperelliptic curves, $g = 3$




Q_3
P_3^∞
P_1^∞
P_2^∞
P_1
P_2
P_4^∞
l^∞
Q_2
K_2
R_1
Q_1
K_1
R_2
K_3
R_3
P_3

Group Law for non-hyperelliptic curves, $g = 3$

Let $D^\infty := P_1^\infty + P_2^\infty + P_3^\infty$. For an element D in $\mathrm{Div}^0(C)$, let D^+ be an effective divisor (generically unique) such that $D^+ - D^\infty \sim D$.

Theorem

Let $D_1, D_2 \in Div_k^0(C)$. Then $D_1 + D_2$ is equivalent to a divisor $D = D^+ - D^\infty$, where the points in the support of D^+ are given by the following algorithm:

1. *Take the unique cubic E which goes (with multiplicity) through the support of D_1^+, D_2^+ and $P_1^\infty, P_2^\infty, P_4^\infty$. This cubic also crosses C in the residual effective divisor D_3.*
2. *Take the unique conic Q which goes through the support of D_3 and P_1^∞, P_2^∞. This conic also crosses C in the residual effective divisor D^+.*

Group Law for non-hyperelliptic curves, $g = 3$

Let $D^\infty := P_1^\infty + P_2^\infty + P_3^\infty$. For an element D in $\mathrm{Div}^0(C)$, let D^+ be an effective divisor (generically unique) such that $D^+ - D^\infty \sim D$.

Theorem

Let $D_1, D_2 \in \mathrm{Div}_k^0(C)$. Then $D_1 + D_2$ is equivalent to a divisor $D = D^+ - D^\infty$, where the points in the support of D^+ are given by the following algorithm:

1. *Take the unique cubic E which goes (with multiplicity) through the support of D_1^+, D_2^+ and $P_1^\infty, P_2^\infty, P_4^\infty$. This cubic also crosses C in the residual effective divisor D_3.*
2. *Take the unique conic Q which goes through the support of D_3 and P_1^∞, P_2^∞. This conic also crosses C in the residual effective divisor D^+.*

- For hyperelliptic curves $y^2 + h(x)y = f(x)$, Mumford proposed a unique (compact) representation of reduced divisors by a pair of two polynomials u, v s.t.

$$\begin{cases} u, v \in \mathbb{F}_q[x], \\ u \text{ monic}, \\ \deg_x v < \deg_x u \leq g, \\ u(x) \text{ divides } v(x)^2 + h(x)v(x) - f(x). \end{cases}$$

$P_i = (x_i, y_i) \in \mathrm{Supp}_{[u,v]} \Leftrightarrow u(x_i) = 0, v(x_i) = y_i$ with multiplicity .

- Not anymore true for non-hyperelliptic curves, only suitable for (typical divisor).

Typical divisors

For non-hyperelliptic curves of genus 3: The Mumford representation in only useful for (typical divisors).

A *typical* divisor $D = D^+ - D_\infty \in \mathrm{Div}_k^0(C)$ is a divisor with the following properties:

- $\deg(D^+) = 3, \quad D^+ \geq 0$,
- the three points in the support of D^+ are non-collinear,
- there is no point at infinity in the support of D^+,
- the $(x_i)_{i=1,2,3}$ are distinct ($P_i = (x_i : y_i : 1)$ be the three points in the support of D^+).

For non-hyperelliptic curves of genus 3 :

Theorem

Let C the plane quartic with affine equation $f(x,y)=0$. A typical reduced divisor over k can be uniquely represented by a pair of two polynomials u,v s.t.

$$\begin{cases} u,v \in \mathbb{F}_q[x], \\ u \textit{ monic}, \quad \deg_x u = 3, \\ \deg_x v = 2, \\ u(x) \textit{ divides } f(x,v(x)). \end{cases}$$

Algebraic interpretation

INPUT: $D_1 = [u_1, v_1], \quad D_2 = [u_2, v_2],$

$$C/k: \quad y^3 + h_1(x)y^2 + h_2(x)y - f_4(x) = 0$$

Three steps: finding the cubic w, reduce $-(D_1 + D_2)$, taking the opposite.

First step: computation of the cubic.

The only step where we distinguish between addition and doubling.

In the most common case $w = y^2 + sy + t$, where $\deg_x(s) = 2$, $\deg_x(t) = 2$.

We use the fact:

- $w \in \langle y - v_1, u_1 \rangle \cap \langle y - v_2, u_2 \rangle$ for addition.
- $w \in \langle y - v_1, u_1 \rangle^2 = \langle (y - v_1)^2, (y - v_1) \cdot u_1, u_1^2 \rangle$ for doubling.

Algebraic interpretation

INPUT: $D_1 = [u_1, v_1], \quad D_2 = [u_2, v_2],$
$C/k : \quad y^3 + h_1(x)y^2 + h_2(x)y - f_4(x) = 0$

Three steps: finding the cubic w, reduce $-(D_1 + D_2)$, taking the opposite.

First step: computation of the cubic.

The only step where we distinguish between addition and doubling.

In the most common case $w = y^2 + sy + t$, where $\deg_x(s) = 2$, $\deg_x(t) = 2$.

We use the fact:

- $w \in \langle y - v_1, u_1 \rangle \cap \langle y - v_2, u_2 \rangle$ for addition.
- $w \in \langle y - v_1, u_1 \rangle^2 = \langle (y - v_1)^2, (y - v_1) \cdot u_1, u_1^2 \rangle$ for doubling.

Algebraic interpretation

Second step: computation of $-(D_1+D_2)$.

- $u_{-(D_1+D_2)} =$ normalized quotient of $\mathrm{Res}(w, C, y)$ by $u_1 \cdot u_2$
- To compute $v_{-(D_1+D_2)}$ use the relation

$$(t - s^2 - h_2 + sh_1) \cdot v_{-(D_1+D_2)} \equiv (st - th_1 - f_4) \mod u_{-(D_1+D_2)},$$

Third step: computation of D_1+D_2.

- $v_{D_1+D_2} = v_{-(D_1+D_2)}$
- $u_{D_1+D_2} =$ normalized quotient of

$$v_{D_1+D_2}^3 + v_{D_1+D_2}^2 h_1 + v_{D_1+D_2} h_2 - f_4$$

by $u_{D_1+D_2}$

Second step: computation of $-(D_1+D_2)$.

- $u_{-(D_1+D_2)} =$ normalized quotient of $\mathrm{Res}(w, C, y)$ by $u_1 \cdot u_2$
- To compute $v_{-(D_1+D_2)}$ use the relation

$$(t - s^2 - h_2 + sh_1) \cdot v_{-(D_1+D_2)} \equiv (st - th_1 - f_4) \mod u_{-(D_1+D_2)},$$

Third step: computation of D_1+D_2.

- $v_{D_1+D_2} = v_{-(D_1+D_2)}$
- $u_{D_1+D_2} =$ normalized quotient of

$$v_{D_1+D_2}^3 + v_{D_1+D_2}^2 h_1 + v_{D_1+D_2} h_2 - f_4$$

by $u_{D_1+D_2}$

Non-hyperelliptic Addition, $g = 3$

Algorithm Addition in $\mathrm{Jac}(C)$ **(Flon-Oyono-Ritzenthaler)**

INPUT: $D_1 = [u_1, v_1]$, $D_2 = [u_2, v_2]$ et $C : y^3 + h_1(x)y^2 + h_2(x)y = f_4(x)$
OUTPUT: $D_1 + D_2 = [u_{D_1+D_2}, v_{D_1+D_2}]$

1. *Computation of the cubic E*

   Compute the inverse t_1 de $v_1 - v_2$ modulo u_2

   Determine the remainder r of $(u_1 - u_2)t_1$ by u_2

   Solve the linear system

   $$\begin{cases} \deg_x(-v_1(v_1+s)+u_1\delta_1) = 2 & \text{(2 eq.)} \\ v_1+v_2+s \equiv r\delta_1 \quad [u_2] & \text{(3 eq.)} \end{cases}$$

   with $s, \delta_1 \in k[x]$, $\deg(s) = 2$ et $\deg(\delta_1) = 1$. Then

   $$E = (y - v_1)(y + v_1 + s) + u_1\delta_1$$

2. *Computation of the conic Q*

   Compute $u' := Res^*(E, C, y)/(u_1u_2)$

   Compute the inverse α_1 of $t - s^2 - h_2 + sh_1$ modulo u'

   Compute the remainder v' of $\alpha_1(st - th_1 - f_4)$ by u'

3. *Compute de $D_1 + D_2$*

   $v_{D_1+D_2} := v'$

   $u_{D_1+D_2} := ((v^3 + v^2h_1 + vh_2 - f_4)/(u'))^*$

   $D_1 + D_2 = [u_{D_1+D_2}, v_{D_1+D_2}]$

Algorithm Doubling in $\mathrm{Jac}(C)$ **(Flon-Oyono-Ritzenthaler)**

INPUT: $D_1 = [u_1, v_1]$ and $C : y^3 + h_1(x)y^2 + h_2(x)y = f_4(x)$
OUTPUT: $2D_1 = [u_{2D_1}, v_{2D_1}]$

1. *Computation of the cubic E*

   Compute $\omega_1 = (v_1^3 + v_1^2 h_1 + v_1 h_2 - f_4)/u_1$

   Compute the inverse t_1 of ω_1 modulo u_1

   Compute the remainder r of $(3v_1^2 + 2v_1 h_1 + h_2)t_1$ by u_1

   Solve the linear system

   $$\begin{cases} \deg_x(-v_1(v_1+s) + u_1\delta_1) = 2 & \text{(2 eq.)} \\ v_1 + v_1 + s \equiv r\delta_1 \quad [u_1] & \text{(3 eq.)} \end{cases}$$

   with $s, \delta_1 \in k[x]$, $\deg(s) = 2$ et $\deg(\delta_1) = 1$. Alors

   $$E = (y - v_1)(y + v_1 + s) + u_1\delta_1$$

2. *Computation of the conic Q*

   Compute $u' := Res^*(E, C, y)/(u_1 u_2)$

   Compute the inverse α_1 of $t - s^2 - h_2 + sh_1$ modulo u'

   Compute the remainder v' of $\alpha_1(st - th_1 - f_4)$ by u'

3. *compute de $D_1 + D_2$*

   $v_{D_1+D_2} := v'$

   $u_{D_1+D_2} := ((v^3 + v^2 h_1 + vh_2 - f_4)/(u'))^*$

   $D_1 + D_2 = [u_{D_1+D_2}, v_{D_1+D_2}]$

Explicit formulæ.

- Use Karatsuba or Toom-Cook tricks to speed up the algorithm.
- Cost of the algorithm:

$$C/k: \quad y^3 + h_2(x)y - f_4(x) = 0$$

Operation		hyperelliptic of genus 3	$C_{3,4}$ Picard	$C_{3,4}$ $\deg(h_2) = 1$	$C_{3,4}$ $\deg(h_2) = 2$	generic quartic $\deg(h_2) = 3$
FOR Methods	Add		2I+130M	2I+138M	2I+145M	2I+163M
	Dbl		2I+152M	2I+160M	2I+167M	2I+185M
Previous Work	Add	I+70M	2I+140M	2I+147M	2I+150M	
	Dbl	I+71M	2I+164M	2I+171M	2I+174M	

Comparison and implementation

- Presented algorithm has geometric viewpoint; did not separate composition from reduction like in Cantor algorithm.
- Cantor algorithm and its improvements (Lange) for computing in the Jacobian of hyperelliptic curves of genus 2 coincide with the geometric point of view.
- This geometric approach can be generalized to large genus non-hyperelliptic curves (Oyono-Thériault (work in rogress) : $2I + 272M + 11SQ$ for addition, $2I + 304M + 14SQ$ for doubling and $2I + 41M + 3SQ$ for inverse in the Jacobian of $C_{3,5}$-curves).

Using the 2-adic expansion of 7: $[7]g = [2](([2]g)+g)+g.$

Better, use the -2-adic expansion : $[7]g := -(-[2](-[2](-[2]g))+g).$

This method is useful for groups where computing $-(D_1 + D_2)$ is faster than computing $D_1 + D_2$. This method is in particular interesting for non-hyperelliptic curves.

Algorithm -2-adic Expansion.

INPUT: $m = \sum_{i=0}^{l(m)-1} m_i 2^i \in \mathbb{N}$, $m_i \in \{0, \pm 1\}$, $g \in G$
OUTPUT: $e := mg$

1. Precompute and store $-g$
2. Compute $l(m)$ and $w(m) = \#\{m_i \mid m_i \neq 0\}$
3. Put $e := (-1)^f g$, where $f := l(m) + w(m) \bmod 2$
4. **for** $i = l(m) - 2$ to 0 **do**
   - $e := -2e$
   - $f := 1 - f$
   - **if** $m_i \neq 0$ **then**
     - $e := -(e + (-1)^f m_i g)$
     - $f := 1 - f$
5. **return** e

Scalar multiplication: -2-adic expansion

Using the 2-adic expansion of 7: $[7]g = [2](([2]g)+g)+g$.

Better, use the -2-adic expansion : $[7]g := -(-[2](-[2](-[2]g))+g)$.

This method is useful for groups where computing $-(D_1+D_2)$ is faster than computing D_1+D_2. This method is in particular interesting for non-hyperelliptic curves.

Algorithm -2-adic Expansion.

INPUT: $m = \sum_{i=0}^{l(m)-1} m_i 2^i \in \mathbb{N}$, $m_i \in \{0, \pm 1\}$, $g \in G$
OUTPUT: $e := mg$

1. Precompute and store $-g$
2. Compute $l(m)$ and $w(m) = \#\{m_i \mid m_i \neq 0\}$
3. Put $e := (-1)^f g$, where $f := l(m) + w(m) \bmod 2$
4. **for** $i = l(m) - 2$ to 0 **do**
   $e := -2e$
   $f := 1 - f$
   if $m_i \neq 0$ **then**
      $e := -(e + (-1)^f m_i g)$
      $f := 1 - f$
5. **return** e

Lemma

Let $D = [u(x), v(x)]$ a divisor class in $\mathrm{Jac}(C)(\mathbb{F}_q)$. Let $u(x) = \prod u_i(x)$ the decomposition of u in irreducible factors in $\mathbb{F}_q[x]$. Let $v_i(x) = v(x) \pmod{u_i(x)}$. Then

$$D = \sum [u_i(x), v_i(x)].$$

This "induces" a kind of unique factorisation in $\mathrm{Jac}(C)(\mathbb{F}_q)$:

Definition

- A divisor class $D = [u(x), v(x)]$ is said to be prime if $u(x)$ is irreducible.
- A divisor is said to be B-smooth if the irreducible factors of $u(x)$ have degree $\leq B$.

Index calculus: Gaudry's algorithm

INPUT:

- A genus g hyperelliptic curve C ,
- $D_1 \in \mathrm{Jac}(C)(\mathbb{F}_q)$, and $D_2 \in \langle D_1 \rangle$,
- $n = \mathrm{ord}(D_1)$ (supposed to be prime).

ALGORITHM TO SOLVE THE DLP IN $\langle D_1 \rangle$:

- Choose a good smothness bound $B \leq g$.
- Construct a factor base

$$\mathcal{F}_B = \{D \in \mathrm{Jac}(C)(\mathbb{F}_q) : D \text{ prime, with } \deg u(x) \leq B\} .$$

- Find relations (at least $\#\mathcal{F}_B$)

$$S_i = a_i D_1 + b_i D_2, \quad a_i, b_i \in_R [1, n] ,$$

where S_i factors in $\mathcal{F}_B$.

Index calculus: Gaudry's algorithm

- Linear algebra: Find a linear combination (γ_i)

$$\sum \gamma_i S_i = 0 = (\sum \gamma_i a_i) D_1 + (\sum \gamma_i b_i) D_2 \,.$$

- Deduce the DLP :

$$DL_{D_1}(D_2) = -\frac{\sum \gamma_i a_i}{\sum \gamma_i b_i} \pmod{n} \,.$$

Analysis: Gaudry's algorithm

HEURISTIC: The polynomials $u(x)$ associate to the divisor classes $S_i \in \mathcal{F}_B$ behave like purely random polynomials.

$\Longrightarrow$ The probability of smoothness follows the subexponential law.

Choice of the smoothness bound (Gaudry-Enge):

$$B = \log_q \left(L_{q^g} \left(\frac{1}{2}, \rho \right) \right) .$$

In this case the cost of the algorithm will be:

$$L_{q^g} \left(\frac{1}{2}, \rho \right) .$$

Analysis: Small genus curves

CHEATING: B is an integer (its a degree)!

The above method is subexponential if $g > \log q$.

If g is small (compared to $\log q$), then the optimal value of B tends to 0. However, we must choose $B \geq 1$.
$\Longrightarrow$ Wrong analysis for small g.

Analysis for g fixed: and $q \longrightarrow +\infty$:

Take $B = 1$, then $\#\mathcal{F}_B \approx q$.

The proportion of smooth elements is $\frac{1}{g!}$.

Total costs: $O\left(gq^2 + g!q\right)$, and if $g < \log q$ the time is dominated by $O\left(gq^2\right)$

Costs for Index calculus

Gaudry et al. (2000-2006) provided a modified version of the index calculus (using large / double large primes variation) to get an improvement for the DLP on curves of small genus $g > 2$:

$$O\left((\log q)\, g^2 q^{2-\frac{2}{g}}\right)$$

group operations for solving the DLP.

g	1	2	3	4	5	6
Pollard	$q^{1/2}$	q^1	$q^{3/2}$	q^2	$q^{5/2}$	q^3
Index (original)	q^2	q^2	q^2	q^2	q^2	q^2
index (variation)			$q^{4/3}$	$q^{3/2}$	$q^{8/5}$	$q^{5/3}$

Non-hyperelliptic index calculus

Previous Index calulus attacks carrie over nicely to non-hyperelliptic curves.
Furthermore, Diem (2007) went back to the ideas of Adleman, DeMarrais and Huang: the complexity of its method (for degree d curves) is

$$O\left(q^{2-\frac{2}{d-2}}\right),$$

and is thus $O(q)$ for smooth plane quartics (non-hyperelliptic curves of genus 3).

On the other hand, B. smith (2007) developed a method using isogenies that for $18,57\%$ genus 3 hyperelliptic curves allows one to transfer the DLP to a non-hyperelliptic curve.

Thank you for your attention!