Calcul d'indice et courbes algébriques : de meilleures récoltes

Alexandre Wallet

ENS de Lyon, Laboratoire LIP, Equipe AriC

Today:

Discrete Logarithm Problem
over curves

$\downarrow$

Index-calculus

New results on **harvesting**

$\swarrow$ $\searrow$

A sieving approach
for "smooth harvesting"

- for **all curves**
- improve a general method

Complexity improvements
for "decomposition harvesting"

- hyperelliptic curves over $\mathbb{F}_{q^n}, q = 2^k$
- new practical computations

Discrete Logarithm Problem (DLP)

Let $g, h = [x] \cdot g \in (G, +)$, with $x \in \mathbb{Z}$. Compute x.

Is this a hard problem ?

Classic

- Generic group: **yes**
- For some groups: **no**
- Cryptography: **“yes”**

Quantum

“NO”

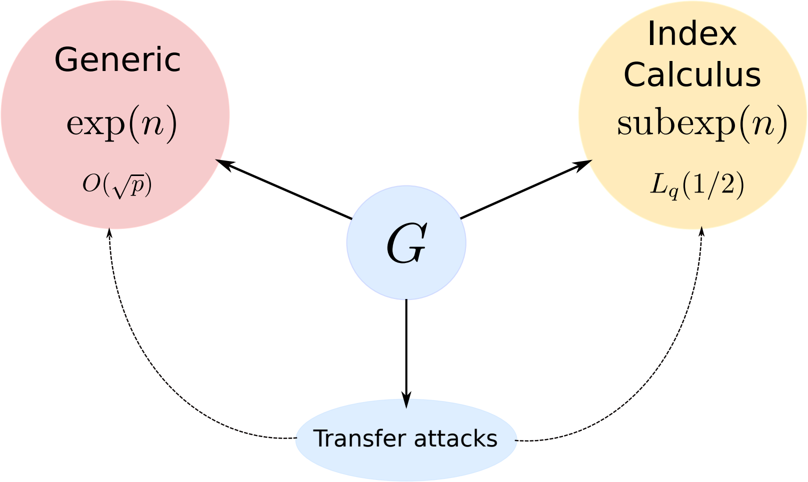
Today’s groups:

Class groups of algebraic curves $\mathcal{J}_{\mathbb{F}_q}(\mathcal{C})$

(and elliptic curves)

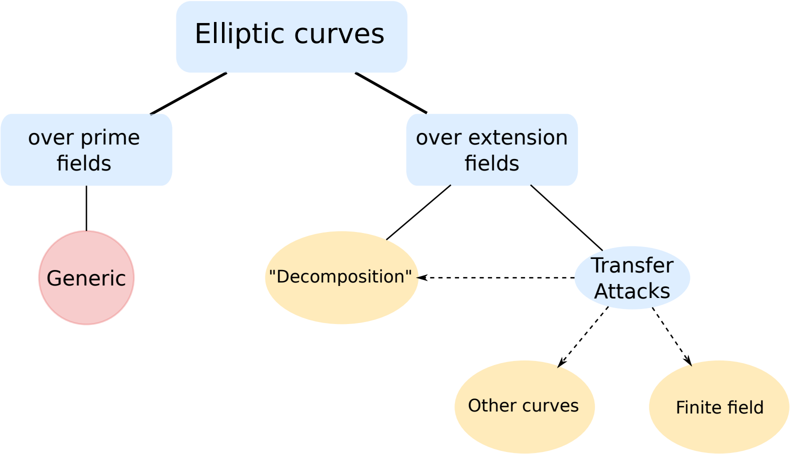
Hardness of Curve-DLP

$\#G \sim p$ prime; $n = \log_2 p$



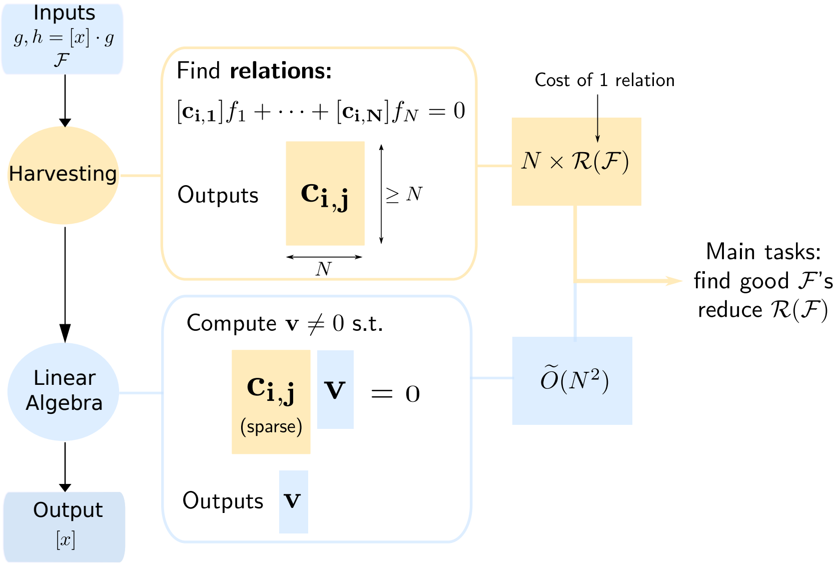
Situation for elliptic curves

For cryptography: **elliptic curves (genus $g = 1$)**



Index-Calculus

Preprocessing: select **factor base** $\mathcal{F} = \{f_1, \ldots, f_N\} \subset G$.



A good $\mathcal{F}$ must be:

- easy to enumerate
- not too big, not too small
- a set of **"small"** elements

$\rightsquigarrow$

There are standard choices.

New choices: **open problem**

Today's target: harvesting in Index-Calculus for curves

Motivations:

Algorithmic Number Theory	Computational Algebraic Geometry	Cryptography

Compute discrete logs in abelian varieties.
How efficient can we be?

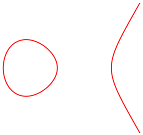
Transfer attacks

Algebraic curves

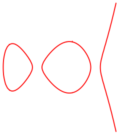
Algebraic curve of **genus** g over a field K:

$\mathcal{C} : P(x, y) = 0$, for some $P \in K[X, Y]$.

$g = 1$: elliptic: $Y^2 = X^3 + AX + B$,
$A, B \in K$



$g = 2$: hyperelliptic: $Y^2 + h_1(X)Y = X^5 + \dots$
$h_1 \in K[X], \deg h_1 \leq 2$



$g \geq 3$: hyperelliptic: $Y^2 + h_1(X)Y = X^{2g+1} + \dots$
$h_1 \in K[X], \deg h_1 \leq \mathbf{g}$

Non-hyperelliptic (all the rest).

Class group and its arithmetic

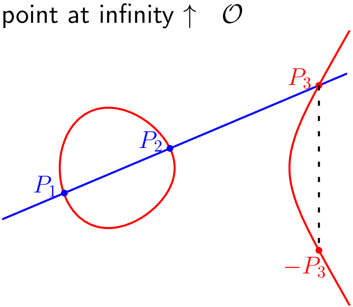
Example: $\mathbf{g} = 1$, $\mathcal{C}$ **elliptic** curve

Line through $P_1, P_2 : f(x,y) = 0$

"Line has 3 zeros and a triple pole at $\mathcal{O}$."

$$\rightsquigarrow \; P_1 + P_2 + P_3 - 3\mathcal{O} \sim 0$$

Addition:
$(P_1 - \mathcal{O}) + (P_2 - \mathcal{O}) \sim ([-P_3] - \mathcal{O})$



Class group and its arithmetic

Example: $\mathbf{g} = 1$, $\mathcal{C}$ **elliptic** curve

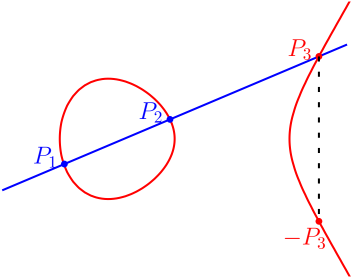
Line through $P_1, P_2 : f(x, y) = 0$

"Line has 3 zeros and a triple pole at $\mathcal{O}$."

$$\rightsquigarrow \; P_1 + P_2 + P_3 - 3\mathcal{O} \sim 0$$

Addition:
$(P_1 - \mathcal{O}) + (P_2 - \mathcal{O}) \sim ([-P_3] - \mathcal{O})$

point at infinity ↑ $\mathcal{O}$



Curve $\mathcal{C}$, **"class group"** $\mathcal{J}(\mathcal{C})$.

It is a quotient group.

Its elements are **"reduced divisors"**.

A reduced divisor is a **formal sum:**

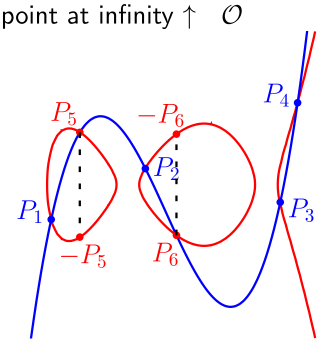
$$D = \sum_{i=1}^{k} P_i - k\mathcal{O},$$

for some $P_1, \ldots, P_k \in \mathcal{C}$, $\mathbf{k} \leq \mathbf{g}$.

Another example: $\mathbf{g} = 2$, $\mathcal{C}$ **hyperelliptic**

Cubic through $P_1, \ldots, P_4 : f(x,y) = 0$

In $\mathcal{J}(\mathcal{H}) : P_1 + \cdots + P_6 - 6\mathcal{O} = 0$



Addition:

$$\underbrace{(P_1 + P_2 - 2\mathcal{O})}_{D_1} + \underbrace{(P_3 + P_4 - 2\mathcal{O})}_{D_2} \sim \underbrace{[-P_5] + [-P_6] - 2\mathcal{O}}_{D_3}$$

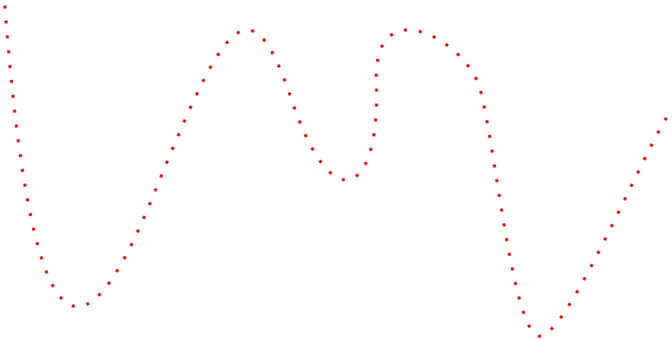
"Smooth harvesting"

Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$) $\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

$K = \mathbb{F}_q$, for $q = p^d$, p prime In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)

Preprocessing: enumerate $\mathcal{F}$.



"Smooth harvesting"

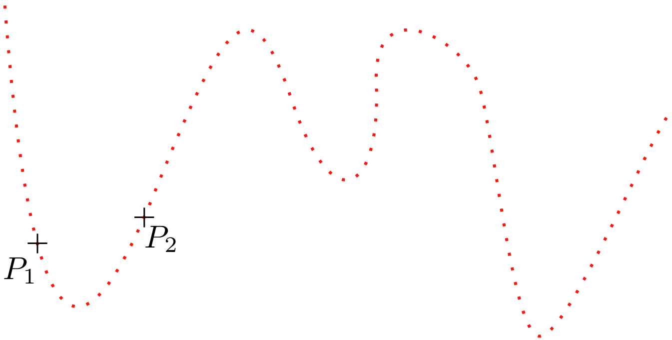
Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$)

$\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

$K = \mathbb{F}_q$, for $q = p^d$, p prime

In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)

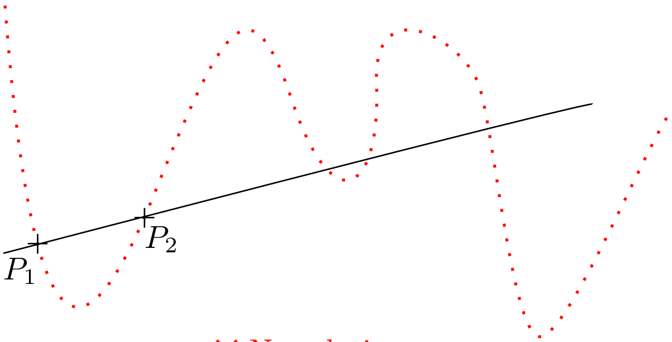


"Smooth harvesting"

Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$) $\qquad$ $\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

$K = \mathbb{F}_q$, for $q = p^d$, p prime $\qquad$ In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)



✕ No relation

"Smooth harvesting"

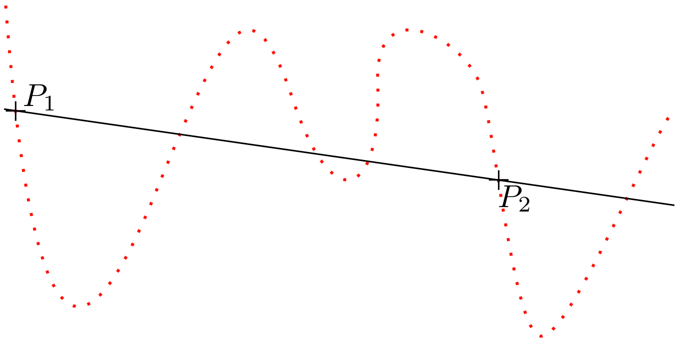
Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$)

$K = \mathbb{F}_q$, for $q = p^d$, p prime

$\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)



"Smooth harvesting"

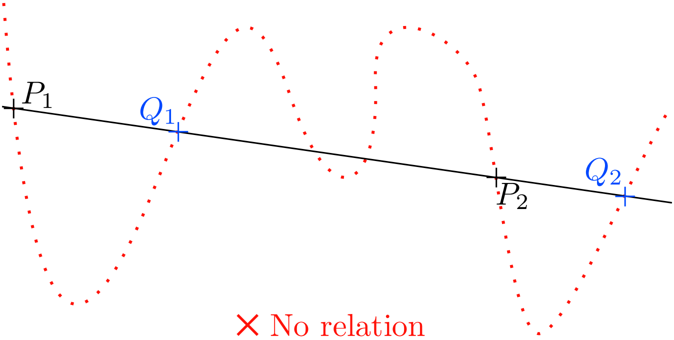
Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$)

$\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

$K = \mathbb{F}_q$, for $q = p^d$, p prime

In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)



"Smooth harvesting"

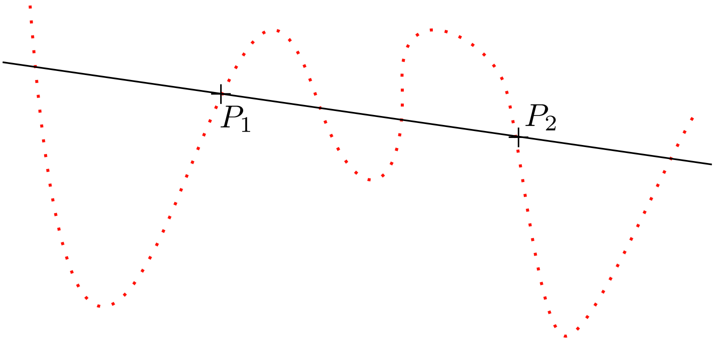
Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$)

$K = \mathbb{F}_q$, for $q = p^d$, p prime

$\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)



"Smooth harvesting"

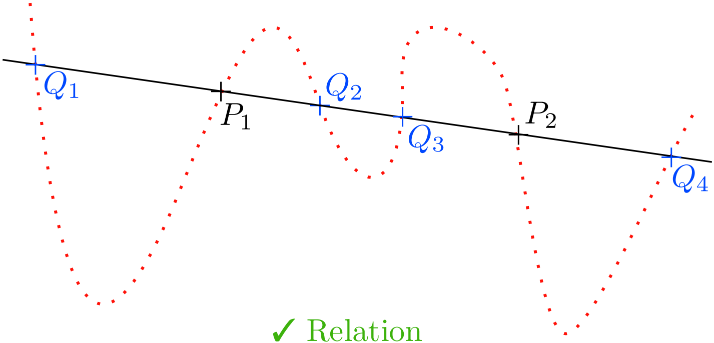
Assume $\mathcal{C}$ is non-hyperelliptic ($\Rightarrow \mathbf{g} \geq 3$)

$\mathcal{C} : C(x, y) = 0$, [Diem'08] $\deg C \leq \mathbf{g} + 1$

$K = \mathbb{F}_q$, for $q = p^d$, p prime

In example: $\deg C = 6$

Factor base $\mathcal{F} = \{ P = (x, y) \in \mathcal{C}(\mathbb{F}_q) \}$ (rational points)



✓ Relation

Cost of smooth harvesting

Input: $C(X,Y)$ in $\mathbb{F}_q[X,Y]$, $\mathcal{F}$ rational points
Output: A relation.

A) Do:

1- Select $P_1, P_2 \in \mathcal{F}$ at random.
2- Compute their line $Y = \lambda X + \mu$.
3- Compute $F(X) = \dfrac{C(X, \lambda X + \mu)}{(X - x_1)(X - x_2)}$.
4- Compute roots x_i's of F in $\mathbb{F}_q$.

While $\#\{\text{roots}\} < \mathbf{g} - 1$.

B) $y_i \leftarrow \lambda x_i + \mu$ for $1 \leq i \leq \mathbf{g} - 1$.

C) Output $\{(x_1, y_1), \ldots, (x_{g-1}, y_{g-1})\}$.

Cost analysis:
$\deg C = \mathbf{g} + 1$

—

1 inversion, 3 multiplications

evaluation

$\sim \mathbf{g}^2 \log q$

Success probability: $\frac{1}{(\mathbf{g}-1)!}$

$\sim 2\mathbf{g}$ multiplications

$\mathcal{R}(\mathcal{F}) \sim (\mathbf{g} - 1)!\mathbf{g}^2 \log q$

A sieving approach to harvesting

	No sieve		Sieve
Theory	Hope to find good lines	VS	Parametrize lines, keep only the good ones
Practice	Root finding at random		**Store** results of cheap computations

Existing approach [SS'14]: restricted to hyperelliptic, rely on sort, backtracking

Our result:

V.Vitse, A.W., *Improved sieving on algebraic curves*, LatinCrypt 2015

- all curve types, adaptable to "balanced" variants
- VS [SS'14]: skip computations, better memory efficiency, no sorting.

Illustration for non-hyperelliptic curves

$$\mathcal{C} : C(x,y) = 0, \text{ genus } \mathbf{g} \geq 3. \text{ [Diem'08]: } \deg C \leq \mathbf{g}+1$$

Factor base $\mathcal{F} = \{P, P_1, P_2, \dots\}$. **First round of sieving:** fix $P = (x_P, y_P)$.

Slope of a line through P: $\lambda_P(P_i) = \dfrac{y_i - y_P}{x_i - x_P}$ (cheap)

Loop over $\mathcal{F}$, compute $\lambda_P(P_i)$'s:

$$\begin{array}{cccccc} & \lambda_P(P_1) & \lambda_P(P_2) & \lambda_P(P_3) & \dots \\ T = & \left[\begin{array}{c} 0 \end{array}\right. & 0 & 0 & \left. \dots \right] \end{array}$$

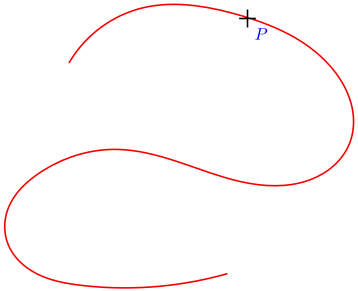


Illustration for non-hyperelliptic curves

$$\mathcal{C} : C(x, y) = 0, \text{ genus } \mathbf{g} \geq 3. \text{ [Diem'08]: } \deg C \leq \mathbf{g} + 1$$

Factor base $\mathcal{F} = \{P, P_1, P_2, \ldots\}$. **First round of sieving:** fix $P = (x_P, y_P)$.

Slope of a line through P: $\lambda_P(P_i) = \dfrac{y_i - y_P}{x_i - x_P}$ (cheap)

Loop over $\mathcal{F}$, compute $\lambda_P(P_i)$'s:

$$\begin{array}{ccccc} & \lambda_P(P_1) & \lambda_P(P_2) & \lambda_P(P_3) & \ldots \\ T = & \left[\; 1 \right. & 0 & 0 & \left. \ldots \;\right] \end{array}$$

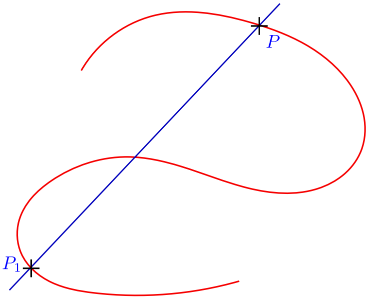


Illustration for non-hyperelliptic curves

$$\mathcal{C} : C(x,y) = 0, \text{ genus } \mathbf{g} \geq 3. \text{ [Diem'08]: } \deg C \leq \mathbf{g} + 1$$

Factor base $\mathcal{F} = \{P, P_1, P_2, \dots\}$. **First round of sieving:** fix $P = (x_P, y_P)$.

$$\text{Slope of a line through } P\text{: } \lambda_P(P_i) = \frac{y_i - y_P}{x_i - x_P} \text{ (cheap)}$$

Loop over $\mathcal{F}$, compute $\lambda_P(P_i)$'s:

$$\begin{array}{cc} & \begin{array}{cccc} \lambda_P(P_1) & \lambda_P(P_2) & \lambda_P(P_3) & \dots \end{array} \\ T = & \left[\begin{array}{cccc} 1 & 1 & 0 & \dots \end{array} \right] \end{array}$$

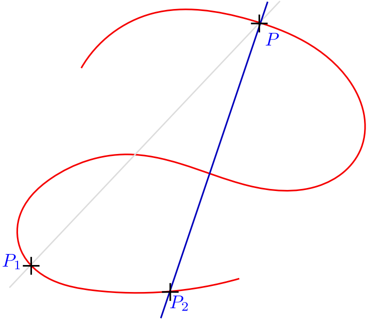


Illustration for non-hyperelliptic curves

$\mathcal{C} : C(x, y) = 0$, genus $\mathbf{g} \geq 3$. [Diem'08]: $\deg C \leq \mathbf{g} + 1$

Factor base $\mathcal{F} = \{P, P_1, P_2, \dots\}$.

First round of sieving: fix $P = (x_P, y_P)$.

Slope of a line through P: $\lambda_P(P_i) = \dfrac{y_i - y_P}{x_i - x_P}$ (cheap)

Loop over $\mathcal{F}$, compute $\lambda_P(P_i)$'s:

$$\begin{array}{cc} & \begin{array}{cccc} \lambda_P(P_1) & \lambda_P(P_2) & \lambda_P(P_3) & \dots \end{array} \\ T = & \left[\begin{array}{cccc} 1 & 1 & 1 & \dots \end{array} \right] \end{array}$$

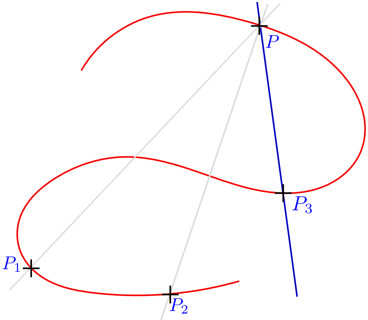


Illustration for non-hyperelliptic curves

$\mathcal{C} : C(x, y) = 0$, genus $\mathbf{g} \geq 3$. [Diem'08]: $\deg C \leq \mathbf{g} + 1$

Factor base $\mathcal{F} = \{P, P_1, P_2, \dots\}$. **First round of sieving:** fix $P = (x_P, y_P)$.

Slope of a line through P: $\lambda_P(P_i) = \dfrac{y_i - y_P}{x_i - x_P}$ (cheap)

Loop over $\mathcal{F}$, compute $\lambda_P(P_i)$'s:

$$\begin{array}{cc} & \begin{array}{cccc} \lambda_P(P_1) & \lambda_P(P_2) & \lambda_P(P_3) & \dots \end{array} \\ T = & \left[\begin{array}{cccc} \mathbf{2} & 1 & 1 & \dots \end{array} \right] \end{array}$$

$\lambda_P(P_i) = \lambda_P(P_j) \Leftrightarrow P, P_i, P_j$ lined up.

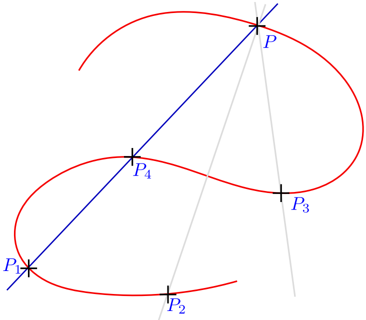


Illustration for non-hyperelliptic curves

$\mathcal{C} : C(x, y) = 0$, genus $\mathbf{g} \geq 3$. [Diem'08]: $\deg C \leq \mathbf{g} + 1$

Factor base $\mathcal{F} = \{P, P_1, P_2, \dots\}$. **First round of sieving:** fix $P = (x_P, y_P)$.

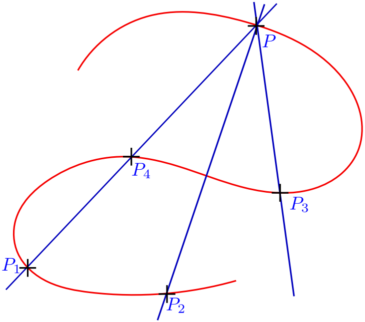
Slope of a line through P: $\lambda_P(P_i) = \dfrac{y_i - y_P}{x_i - x_P}$ (cheap)

Loop over $\mathcal{F}$, compute $\lambda_P(P_i)$'s:

$$\begin{array}{cccccc} & \lambda_P(P_1) & \lambda_P(P_2) & \lambda_P(P_3) & \dots & \\ T = [& \mathbf{2} & 1 & 1 & \dots &] \end{array}$$

$\lambda_P(P_i) = \lambda_P(P_j) \Leftrightarrow P, P_i, P_j$ lined up.

When $\mathbf{T}[\lambda_\mathbf{i}] = \mathbf{g}$: **relation**



Analysis

For one loop:

- $O(q)$ multiplications + $O(q)$ storage.
- Expect $\approx \dfrac{q}{\mathbf{g}!}$ relations.

$\Rightarrow$ Harvesting in $\approx \mathbf{g}!q$.

Previous approach: $\approx (\mathbf{g}-1)!q(\mathbf{g}^2 \log q)$ $\Rightarrow$ Speed-up $\approx \mathbf{g} \log q$.

Relations management:

Loop on P uses all lines through P. $\Rightarrow$ No duplicate relations.

Sieving = time/memory trade-off.

Timings

q		78137	177167	823547	1594331
Genus 3, degree 4	Diem	11.5	27.5	135.1	266.1
	Sieving	3.6	9.3	46.9	94.6
	Ratio	**3.1**	**2.9**	**2.8**	**2.8**
Genus 4, degree 5	Diem	51.8	122.4	595.8	1174
	Sieving	15.5	40.1	195.1	387.6
	Ratio	**3.3**	**3.1**	**3.1**	**3**
Genus 5, degree 6	Diem	229.4	535.8	2581	5062
	Sieving	75.6	199	969.3	1909
	Ratio	**3**	**2.6**	**2.6**	**2.6**
Genus 7, degree 7	Diem	1382	3173	14990	29280
	Sieving	458.5	1199	5859	11510
	Ratio	**3**	**2.6**	**2.5**	**2.5**

Implementation in Magma; CPU Intel© Core i5@2.00Ghz processor.
Time to collect 10000 relations, expressed in seconds.

Factor bases over an extension field

Let $\mathcal{C}$ be a curve of genus g, with defining equation in $\mathbb{F}_{q^n}[X, Y]$.

$$\mathbb{F}_{q^n} = \mathbb{F}_q + \mathbb{F}_q \cdot \mathbf{t} + \cdots + \mathbb{F}_q \cdot \mathbf{t}^{n-1}$$

"Small elements": points with coordinates in a subspace of $\mathbb{F}_{q^n}$.

A candidate: $\mathcal{F} = \{P = (x, y) \in \mathcal{C} \,:\, x \in \mathbb{F}_q, y \in \mathbb{F}_{q^n}\}$.

[Gaudry'09, Nagao'10, Diem'11]

Factor bases over an extension field

Let $\mathcal{C}$ be a curve of genus $\mathbf{g}$, with defining equation in $\mathbb{F}_{q^n}[X, Y]$.

$$\mathbb{F}_{q^n} = \mathbb{F}_q + \mathbb{F}_q \cdot \mathbf{t} + \cdots + \mathbb{F}_q \cdot \mathbf{t}^{n-1}$$

"Small elements": points with coordinates in a subspace of $\mathbb{F}_{q^n}$.

A candidate: $\mathcal{F} = \{P = (x, y) \in \mathcal{C} \; : \; x \in \mathbb{F}_q, y \in \mathbb{F}_{q^n}\}$.
[Gaudry'09, Nagao'10, Diem'11]

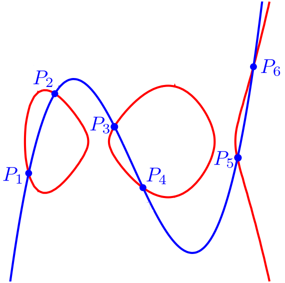
Want: $P_1 + \cdots + P_m = 0$, with $D_i \in \mathcal{F}$

meaning: find curve f s.t. $f(x_i, y_i) = 0$

Fix m: good f's $\in$ space of $\dim = m - \mathbf{g} = d$.

$a_1, \ldots, a_d$: symbolic coordinates.

Goal: find values for a_i's s.t. f is good.

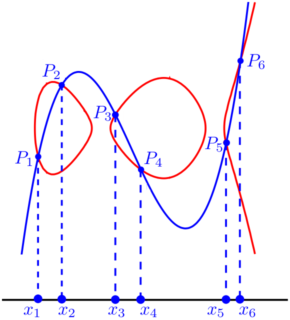


“Projection” of a relation

$(x, y) \in \mathcal{F}$ iff $x \in \mathbb{F}_q \Rightarrow$ “project” on x-line to restrict coordinate space.

$R(x) :=$ Symbolic resultant (in y) of $f(x, y)$ and $\mathcal{C}$'s equation.

- $R(x) = x^m + \sum_{j<m} \underbrace{R_j(a_1, \ldots, a_d)}_{\in \mathbb{F}_{q^n}[a_1, \ldots, a_d]} \cdot x^j$,
- Property: $R(x_i) = 0$.
- All x_i's $\in \mathbb{F}_q$ implies all $R_j(a_1, \ldots, a_d)$'s $\in \mathbb{F}_q$.



Restriction of scalars[1]

The base field is $\mathbb{F}_{q^n} = \mathbb{F}_q + \mathbb{F}_q \cdot \mathbf{t} + \cdots + \mathbb{F}_q \cdot \mathbf{t}^{n-1}$

$$
\begin{array}{ccc}
 & R_j(a_1, \ldots, a_d) & \\
\swarrow & & \searrow \\
\forall \text{ coeff } \lambda \in \mathbb{F}_{q^n}: & & \text{New variables:} \\
\lambda = \lambda_1 + \lambda_2 \mathbf{t} + \cdots + \lambda_n \mathbf{t}^{n-1} & & a_i = a_{i1} + a_{i2}\mathbf{t} + \cdots + a_{i,n}\mathbf{t}^{n-1} \\
\searrow & & \swarrow \\
 & R_{j1}(\mathbf{a}) + R_{j2}(\mathbf{a}) \cdot \mathbf{t} + \cdots + R_{jn}(\mathbf{a}) \cdot \mathbf{t}^{n-1} &
\end{array}
$$

$$
R_j(a_1, \ldots, a_d) \in \mathbb{F}_q \quad \Leftrightarrow \quad \begin{cases} R_{j2}(\mathbf{a}) = 0 \\ \vdots \\ R_{jn}(\mathbf{a}) = 0 \end{cases}
$$

polynomial system over $\mathbb{F}_q$.

[1] [Gaudry'09, Nagao'10, Diem'11]

Gröbner bases and relation cost

$$\boxed{\text{Original System}} \xrightarrow[\text{algo}]{\text{F4 or F5}} \boxed{\text{Degree order}} \xrightarrow[\text{algo}]{\text{FGLM}} \boxed{\text{Lex order}} \xrightarrow[\text{finding}]{\text{Root}} \boxed{\text{Solutions}}$$

If $\mathcal{C}$ has genus $\mathbf{g}$, $m = n\mathbf{g}$ implies $d = (n-1)\mathbf{g}$.

$\Rightarrow n(n-1)\mathbf{g}$ equations and variables

Gröbner bases and relation cost

$$\boxed{\text{Original System}} \xrightarrow[\text{algo}]{\text{F4 or F5}} \boxed{\text{Degree order}} \xrightarrow[\text{algo}]{\text{FGLM}} \boxed{\text{Lex order}} \xrightarrow[\text{finding}]{\text{Root}} \boxed{\text{Solutions}}$$

If $\mathcal{C}$ has genus $\mathbf{g}$, $m = n\mathbf{g}$ implies $d = (n-1)\mathbf{g}$.

$\Rightarrow n(n-1)\mathbf{g}$ equations and variables

Main parameter: Δ #**solutions** (in $\overline{\mathbb{F}_{q^n}}$)

Above process runs in $\widetilde{O}(\Delta^\omega)$

Relations when solutions are in $\mathbb{F}_q$.

Cost analysis if $\mathcal{C}$ hyperelliptic

$$\Delta = 2^{n(n-1)\mathbf{g}}$$

Success probability: $\dfrac{1}{(n\mathbf{g})!}$

$$\mathcal{R}(\mathcal{F}) \sim (ng)! \cdot 2^{\omega n(n-1)\mathbf{g}}$$

Reducing the number of solutions

$\Delta = 2^{n(n-1)\mathbf{g}}$ is quickly huge.

Can be reduced by **exploiting structural properties** (e.g. symmetries) before running algorithms.

Examples:

- $\begin{cases} x_1 + x_2 + x_3 = a \\ x_1x_2 + x_1x_3 + x_2x_3 = b \\ x_1x_2x_3 = c \end{cases}$ $\xrightarrow[\text{symmetries}]{\text{using}}$ $\begin{cases} e_1 = a \\ e_2 = b \\ e_3 = c \end{cases}$

  6 solutions. 1 solution.

- $\begin{cases} x^2 - 2xy + y^2 = a \\ x^2 + y - x - \sqrt{a} = b \end{cases}$ $\xrightarrow[\text{powers}]{\text{removing}}$ $\begin{cases} y = x + \sqrt{a} \\ x^2 = b \end{cases}$

  $x^2 - 2xy + y^2 = (x-y)^2$

  4 solutions. 2 solutions.

Known reductions for elliptic curves ($\mathrm{g} = 1$):
[FGHR'14, FHJRV'14, GG'14]
"Summation polynomials" and symmetries

Our results:

J-C. Faugère, A.W., *The Point Decomposition Problem in Hyperelliptic Curves*.
Designs, Codes and Cryptography [to be published]

- If $q = 2^n$, reduction of Δ for **hyperelliptic** curves of all **genus** g.
- Practical harvesting on a meaningul curve ($\#\mathcal{J}(\mathcal{H}) \sim 184$ bits prime).

$$\mathcal{H} : y^2 + h_1(x)y = h_0(x) \text{ hyperelliptic of genus } \mathbf{g} \text{ over } \mathbb{F}_{q^n}, \text{ with } q = 2^k$$

Good f's: $$f(x,y) = \underbrace{\sum_{i=0}^{d_1} a_i x^i}_{p(x)} + y \cdot \underbrace{\sum_{i=0}^{d_2} a_{i+d_1+1} x^j}_{q(x)} \quad \text{with } \begin{cases} d_1 = \lfloor \frac{m}{2} \rfloor \\ d_2 = \lfloor \frac{m-2\mathbf{g}-1}{2} \rfloor \end{cases}$$

Then :

$$R(x) \quad = \quad p(x)^2 + q(x)^2 h_0(x) \quad + \quad p(x)q(x)h_1(x)$$

Shape of the resultant

$\mathcal{H} : y^2 + h_1(x)y = h_0(x)$ hyperelliptic of genus $\mathbf{g}$ over $\mathbb{F}_{q^n}$, with $q = 2^k$

Good f's: $$f(x,y) = \underbrace{\sum_{i=0}^{d_1} a_i x^i}_{p(x)} + y \cdot \underbrace{\sum_{i=0}^{d_2} a_{i+d_1+1} x^j}_{q(x)}$$ with $\begin{cases} d_1 = \lfloor \frac{m}{2} \rfloor \\ d_2 = \lfloor \frac{m-2\mathbf{g}-1}{2} \rfloor \end{cases}$

Then :

$R(x)$	$=$	$p(x)^2 + q(x)^2 h_0(x)$	$+$	$p(x)q(x)h_1(x)$
$\deg = m$		$\deg = m$		$\deg \leq \mathbf{m}$
Monomials in a_i's:		a_i^2 **only**		$a_i a_j$, $i \neq j$

In Char $= 2$, equations coming from the "head" of R can be **squares**.

Results

$$h_1(x) = x^{\deg h_1} + \cdots + a_t x^t, \text{ where } 0 \leq t \leq \deg h_1 \leq \mathbf{g}.$$

Number of squares in R:

For $\mathbf{L} = \deg h_1 - t$, $R - x^m$ has $\mathbf{g} - \mathbf{L}$ square coefficients.

Corollary:

We can find $(n-1)(\mathbf{g} - \mathbf{L})$ square equations in the systems.

Additional results:

- any system contains a subsystem of $n-1$ equations in $n-1$ variables.
- it is determined whp.: solve it before solving the remaining equations.

Analysis of the new number of solutions

Genericity assumption: systems behave like regular systems of dimension 0 over $\mathbb{F}_{q^n}$.

Before:

- $\#$vars $= (n-1)n\mathbf{g}$
- $\#$eqs $= (n-1)n\mathbf{g}$
- Eqs have $\deg = 2$

$\Rightarrow \Delta = 2^{n(n-1)\mathbf{g}}$

Now (after presolving):

- $(n-1)(n\mathbf{g}-1)$ eqs and vars
- $(n-1)(\mathbf{g}-\mathbf{L})$ linear eqs
- remaining have $\deg = 2$

$\Rightarrow \Delta = 2^{(n-1)((n-1)\mathbf{g}+\mathbf{L}-1)}$

$$2^{(n-1)((n-1)\mathbf{g}-1)} \leq \Delta \leq 2^{(n-1)(n\mathbf{g}-1)}$$

$$\textbf{factor} \quad \mathbf{2^{(n-1)(g+1)}} \geq \frac{\Delta}{\Delta} \geq \mathbf{2^{n-1}}$$

Table: Average time[2] to find one relation.
Parameters: $\mathbf{g} = 2$, $q = 2^{15}$, curves with $\mathbf{L} = 0$.

n	Approach	# solutions	Time, one system	Time, one relation
3	classic	4096	~ 1500 sec.	~ 12.5 **days**
	ours	64	~ 0.029 sec.	~ 21 **seconds**
4	classic	2^{24}	—	—
	ours	2^{15}	~ 250 hours	—

NB: Success probability $= \dfrac{1}{(n\mathbf{g})!}$

[2] Computations with Magma 2.19

Expected nops for meaningful parameters

- Parameters: $\mathbf{g} = 2$, $\mathbf{L} = 0$, $q = 2^{31}$, $n = 3$. (NB: base field is $\mathbb{F}_{2^{93}}$).
- $\mathcal{C}$ such that $\#\mathcal{J}(\mathcal{C}) \sim p$ prime, with $\log p = 184$ bits.

Table: Comparisons of possible algorithms

Algorithm	estimated nops	
ρ-Pollard	$\sim 2^{92}$	
Index-calculus	Harvesting	Linear algebra
"Smooth"	$\sim 2^{93}$	$\sim 2^{93}$
Decomposition, old ($\Delta = 2^{12}$)	$\sim 2^{69.5}$	$\sim 2^{63}$
Decomposition, ours ($\Delta = 2^{6}$)	$\sim 2^{55}$	$\sim 2^{63}$

$$\text{NB: Cost of harvesting} \sim \#\mathcal{F} \times \Delta^{\omega} \times (ng)!$$
$$\sim 2^{31} \times \Delta^{2.4} \times 2^{9.5}$$

- With dedicated implementation[3], we find 1 relation in **2.3 sec. in avg.**

Table: Comparison with a recent record computation

	#rels	harvesting	matrix size, density††	#linalg.†††	$\log p$
[KDL+'17]†	$\sim 2^{33}$	6 months	2^{24}, 184	$\sim 2^{56}$	768
our work	$\sim 2^{31}$	7 days	2^{28} , 87	$\sim 2^{63}$	184

†: [KDL+'17] *Computation of a 768 bits prime field discrete logarithm*, EuroCrypt 2017

††: Size after filtering. [KDL+'17] use a dedicated filtering.

† † †: linear algebra is done modulo p

[3] FGb and code gen., Sparse FGLM, NTL

Perspectives and open problems

Decomposition harvesting: (ongoing work)

(A) Reductions for elliptic curves use "summation polynomials".

- analog notion for other curves?
- [FW'17]: a possible approach, but limited efficiency. Improvements?

(B) There are lots of "symmetries" (automorphisms) in higher genus class groups.

- Can we exploit them?

Open problems:

(C) Is there any choice of factor base for elliptic curve over $\mathbb{F}_p$?

(D) New families of curves with subexponential Index-calculus?

(E) Can we find better than Index-calculus?

Thank you :)