Elliptic Curve Discrete Logarithm Problem

Vanessa VITSE

Université de Versailles Saint-Quentin, Laboratoire PRISM

October 19, 2009

Motivations

Discrete logarithm problem (DLP)

Given G group and $g, h \in G$, find – when it exists – an integer x s.t.

$$h = g^x$$

Many cryptosystems rely on the hardness of this problem:

- Diffie-Hellman key exchange protocol
- Elgamal encryption and signature scheme, DSA
- pairing-based cryptography : IBE, BLS short signature scheme...

Hardness of DLP

It depends of the choice of G:

1. G subgroup of $(\mathbb{Z}/n\mathbb{Z}, +)$:
   polynomial complexity with extended Euclid algorithm

2. G subgroup of $(\mathbb{F}_q^*, \times)$ $(q = p^n)$:
   subexponential complexity with index calculus
   - $O(L_q(1/3))$ complexity with FFS (resp. NFS) for small (resp. larger) characteristic, where $L_q(\nu, c) = e^{c(\log q)^{\nu}(\log\log q)^{1-\nu}}$
   - key sizes needed: $\simeq 1900$ bits

3. G subgroup of $(E(\mathbb{F}_{p^n}), +)$:
   exponential complexity (in most cases) for known algorithms
   - $E(\mathbb{F}_p)$ (p prime) or $E(\mathbb{F}_{2^n})$ are now standards (FIPS 186-3), and $E(\mathbb{F}_{p^n})$ used in many protocols
   - key sizes needed: $\simeq 160$ bits

Generic attacks

Definition

Generic algorithm: only makes use of the group law but not the specific description of G
$\hookrightarrow$ formal definition: oracle calls

Generic attacks

Definition

Generic algorithm: only makes use of the group law but not the specific description of G
$\hookrightarrow$ formal definition: oracle calls

Lower bound (Shoup)

A generic algorithm that solves the DLP has a complexity of at least

$$\Omega(\max(\alpha_i \sqrt{p_i}))$$

where $\#G = \prod_i p_i^{\alpha_i}$, p_i primes.

How to achieve the lower bound ?

1. Pohlig-Hellman reduction
2. Shanks's "Baby Step Giant Step" or "Pollard-ρ" algorithm

DLP on elliptic curves over finite fields (ECDLP)

Question

Are there known algorithms faster than generic methods for solving the ECDLP ?

DLP on elliptic curves over finite fields (ECDLP)

Question

Are there known algorithms faster than generic methods for solving the ECDLP ?

Some answers...

- No in general
- Specific methods work in some cases:
  - supersingular curves: transfer to $\mathbb{F}_{q^k}^*$ via pairings
  - anomalous curves: lift to $E(\mathbb{Q}_p)$
  - some curves over $\mathbb{F}_{q^n}$: transfer to $J_{\mathcal{C}}(\mathbb{F}_q)$ where $\mathcal{C}$ is a genus $g > 1$ curve via Weil descent

An index calculus method over $E(\mathbb{F}_{q^n})$

Original algorithm (Gaudry, Diem)

Complexity of DLP over $E(\mathbb{F}_{q^n})$ in $\tilde{O}(q^{2-\frac{2}{n}})$ but with hidden constant exponential in n^2

- faster than generic methods when $n \geq 3$ and $\log q > C.n$
- subexponential complexity when $n = \Theta(\sqrt{\log q})$

An index calculus method over $E(\mathbb{F}_{q^n})$

Original algorithm (Gaudry, Diem)

Complexity of DLP over $E(\mathbb{F}_{q^n})$ in $\tilde{O}(q^{2-\frac{2}{n}})$ but with hidden constant exponential in n^2

- faster than generic methods when $n \geq 3$ and $\log q > C.n$
- subexponential complexity when $n = \Theta(\sqrt{\log q})$

Our variant

Complexity in $\tilde{O}(q^2)$ but with a better dependency in n

- better than generic methods when $n \geq 5$ and $\log q > c.n$
- better than Gaudry and Diem's method when $\log q < c'.n^2 \log n$

An index calculus method over $E(\mathbb{F}_{q^n})$

Original algorithm (Gaudry, Diem)

Complexity of DLP over $E(\mathbb{F}_{q^n})$ in $\tilde{O}(q^{2-\frac{2}{n}})$ but with hidden constant exponential in n^2

- faster than generic methods when $n \geq 3$ and $\log q > C.n$
- subexponential complexity when $n = \Theta(\sqrt{\log q})$

Our variant

Complexity in $\tilde{O}(q^2)$ but with a better dependency in n

- better than generic methods when $n \geq 5$ and $\log q > c.n$
- better than Gaudry and Diem's method when $\log q < c'.n^2 \log n$

In practice...

The original algorithm can realistically be implemented only for $n \leq 4$, whereas our variant is working for $n = 5$.

Basic form of the index calculus method

Discrete logarithm problem (DLP)

G finite group, given $h, g \in G$ such that $h = [x]g$, recover the secret x.

Basic outline

1. choice of a factor base: $\mathcal{F} = \{g_1, \ldots, g_N\} \subset G$
2. relation search: decompose $[a_i]g + [b_i]h$ (a_i, b_i random) into $\mathcal{F}$

$$[a_i]g + [b_i]h = \sum_{j=1}^{N} [c_{i,j}]g_j$$

3. linear algebra: once k relations found ($k > N$)
   - construct the matrices $A = \begin{pmatrix} a_i & b_i \end{pmatrix}_{1 \leq i \leq k}$ and $M = (c_{i,j})_{\substack{1 \leq i \leq k \\ 1 \leq j \leq N}}$
   - find $v = (v_1, \ldots, v_k) \in ker({}^t M)$ s.t. $vA \neq 0 \bmod r$.
4. solution of DLP : $x = -\left(\sum_i a_i v_i\right) / \left(\sum_i b_i v_i\right) \bmod r$.

Basic example in $\mathbb{F}_p^*$ (p prime)

Discrete logarithm over $\mathbb{F}_{101}^*$

Let $h \in \mathbb{F}_{101}^* = \langle g \rangle$ where $g = 11$ and $h = 82$
Find $x \in [0; 100]$ such that $h = g^x \bmod 101$

Basic example in $\mathbb{F}_p^*$ (p prime)

Discrete logarithm over $\mathbb{F}_{101}^*$

Let $h \in \mathbb{F}_{101}^* = \langle g \rangle$ where $g = 11$ and $h = 82$
Find $x \in [0; 100]$ such that $h = g^x \bmod 101$

1. Factor base :
   $\mathcal{F} = \{2; 3\}$

Basic example in $\mathbb{F}_p^*$ (p prime)

Discrete logarithm over $\mathbb{F}_{101}^*$

Let $h \in \mathbb{F}_{101}^* = \langle g \rangle$ where $g = 11$ and $h = 82$
Find $x \in [0; 100]$ such that $h = g^x \bmod 101$

1. Factor base :
   $\mathcal{F} = \{2; 3\}$
2. Relation search :
   $hg^2 = 24 = 2^3 \times 3$
   $h^2 g = 32 = 2^5$
   $h^3 = 9 = 3^2$

Basic example in $\mathbb{F}_p^*$ (p prime)

Discrete logarithm over $\mathbb{F}_{101}^*$

Let $h \in \mathbb{F}_{101}^* = \langle g \rangle$ where $g = 11$ and $h = 82$
Find $x \in [0; 100]$ such that $h = g^x \bmod 101$

1. Factor base :
   $\mathcal{F} = \{2; 3\}$
2. Relation search :
   $hg^2 = 24 = 2^3 \times 3$
   $h^2 g = 32 = 2^5$
   $h^3 = 9 = 3^2$
3. Linear algebra :

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \overbrace{\begin{pmatrix} 3 & 1 \\ 5 & 0 \\ 0 & 2 \end{pmatrix}}^{M} \begin{pmatrix} \log_g 2 \\ \log_g 3 \end{pmatrix} \text{ and } \begin{pmatrix} 10 & -6 & -5 \end{pmatrix} \in \ker {}^t M$$

Basic example in $\mathbb{F}_p^*$ (p prime)

Discrete logarithm over $\mathbb{F}_{101}^*$

Let $h \in \mathbb{F}_{101}^* = \langle g \rangle$ where $g = 11$ and $h = 82$
Find $x \in [0;100]$ such that $h = g^x \bmod 101$

1. Factor base :
   $\mathcal{F} = \{2;3\}$
2. Relation search :
   $hg^2 = 24 = 2^3 \times 3$
   $h^2g = 32 = 2^5$
   $h^3 = 9 = 3^2$
3. Linear algebra :

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \overbrace{\begin{pmatrix} 3 & 1 \\ 5 & 0 \\ 0 & 2 \end{pmatrix}}^{M} \begin{pmatrix} \log_g 2 \\ \log_g 3 \end{pmatrix} \text{ and } \begin{pmatrix} 10 & -6 & -5 \end{pmatrix} \in \ker {}^t M$$

4. Solution :

$$17x = 14 \bmod 100 \Rightarrow x = 42$$

How to find relations ?

1. $G \subset \mathbb{F}_p^*$, p prime: use the prime factor decomposition of a representant in $]-p/2; p/2[$

$$\mathcal{F} = \{\text{prime numbers smaller than } B\}$$

2. $G \subset \mathbb{F}_{p^n}^*$: consider $\mathbb{F}_{p^n}$ as $\mathbb{F}_p[X]/(f(X))$ and use the irreducible factor decomposition of a representant in $\mathbb{F}_p[X]$

$$\mathcal{F} = \{\text{irreducible polynomials of degree smaller than } B\}$$

3. $G \subset J_{\mathcal{C}}(\mathbb{F}_q)$, $\mathcal{C}$ hyperelliptic curve of genus $g > 1$

$$\mathcal{F} = \{\text{prime reduced divisors of weight smaller than } B\}$$

4. $G \subset E(\mathbb{F}_q)$??

Remarks on the index calculus

Trade-off for the smoothness bound B

- if B too small, very few elements are decomposable
- if B too large, many relations needed and expensive linear algebra step

Linear algebra

- the matrix M usually has a specific shape (very sparse, coefficients located mainly in some parts of M...)
- use of adequate linear algebra tools: structured Gaussian elimination, Lanczos, Wiedemann...

Complexity

- for an optimal value of B, the outlined techniques yield a $O(L(1/2))$ complexity
- more sophisticated methods (NFS/FFS) use a more elaborate relation search and have a $O(L(1/3))$ complexity

Index calculus on $E(\mathbb{F}_{q^n})$

ECDLP

Given $P \in E(\mathbb{F}_{q^n})$ and $Q \in \langle P \rangle$, find x such that $Q = [x]P$

Looking for specific relations

Check whether a given random combination $R = [a]P + [b]Q$ can be decomposed as $R = P_1 + \ldots + P_m$, for a fixed number m

Index calculus on $E(\mathbb{F}_{q^n})$

ECDLP

Given $P \in E(\mathbb{F}_{q^n})$ and $Q \in \langle P \rangle$, find x such that $Q = [x]P$

Looking for specific relations

Check whether a given random combination $R = [a]P + [b]Q$ can be decomposed as $R = P_1 + \ldots + P_m$, for a fixed number m

Main idea: Weil restriction

- write $\mathbb{F}_{q^n}$ as $\mathbb{F}_q[t]/(f(t))$ where f irreducible of degree n
- convenient choice of $\mathcal{F} = \{P = (x, y) \in E(\mathbb{F}_{q^n}) : x \in \mathbb{F}_q, y \in \mathbb{F}_{q^n}\}$
- want to find m points $P_i = (x_{P_i}, y_{P_i})$ s.t. $x_{P_i} = x_{0,P_i}$, $y_{P_i} = y_{0,P_i} + y_{1,P_i} t + \ldots + y_{n-1,P_i} t^{n-1}$ and $R = P_1 + \ldots + P_m$
  $\rightsquigarrow$ solve a huge system of $2n$ equations in $m(n+1)$ variables over $\mathbb{F}_q$

Index calculus on $E(\mathbb{F}_{q^n})$

Second idea

Get rid of the variables y_{P_i} by using Semaev's summation polynomials
$\rightsquigarrow$ system of n equations in m variables over $\mathbb{F}_q$

Index calculus on $E(\mathbb{F}_{q^n})$

Second idea

Get rid of the variables y_{P_i} by using Semaev's summation polynomials
$\rightsquigarrow$ system of n equations in m variables over $\mathbb{F}_q$

Semaev's summation polynomials

Let E be an elliptic curve over K, with reduced Weierstrass equation
$y^2 = x^3 + ax + b$.
The m**-th summation polynomial** is an irreducible symmetric polynomial
$f_m \in K[X_1, \ldots, X_m]$ such that given
$P_1 = (x_{P_1}, y_{P_1}), \ldots, P_m = (x_{P_m}, y_{P_m}) \in E(\overline{K}) \setminus \{O\}$, we have

$$f_m(x_{P_1}, \ldots, x_{P_m}) = 0 \Leftrightarrow \exists \epsilon_1, \ldots, \epsilon_m \in \{1, -1\}, \epsilon_1 P_1 + \ldots + \epsilon_m P_m = O$$

Computation of Semaev's summation polynomials

1. f_m are uniquely determined by induction:

$$f_2(X_1, X_2) = X_1 - X_2$$

$$f_3(X_1, X_2, X_3) = (X_1 - X_2)^2 X_3^2 - 2\left((X_1 + X_2)(X_1X_2 + a) + 2b\right) X_3 + (X_1X_2 - a)^2 - 4b(X_1 + X_2)$$

and for $m \geq 4$ and $1 \leq j \leq m - 3$ by

$$f_m(X_1, X_2, \ldots, X_m) = \mathrm{Res}_X \left(f_{m-j}(X_1, X_2, \ldots, X_{m-j-1}, X), f_{j+2}(X_{m-j}, \ldots, X_m, X)\right)$$

2. $\deg_{X_i} f_m = 2^{m-2} \Rightarrow$ only computable for small values of m

Index calculus on $E(\mathbb{F}_{q^n})$

Back to decomposition computation

1. **goal:** solve the equation
$f_{m+1}(x_{P_1}, \ldots, x_{P_m}, x_R) = 0$, where unknowns are $x_{P_1}, \ldots, x_{P_m} \in \mathbb{F}_q$
2. express the equation in terms of the elementary symmetric polynomials $e_1, \ldots, e_m$ of the variables $x_{P_1}, \ldots, x_{P_m}$:

$$e_k = \sum_{1 \leq i_1 \leq \ldots \leq i_k \leq m} x_{P_{i_1}} \ldots x_{P_{i_k}}$$

3. Weil restriction: sort according to the powers of t

$$f_{m+1}(x_{P_1}, \ldots, x_{P_m}, x_R) = 0 \Leftrightarrow \sum_{i=0}^{n-1} \varphi_i(e_1, \ldots, e_m) t^i = 0$$

$\rightsquigarrow$ system of n polynomial equations of total degree 2^{m-1} in m unknowns

Gaudry's original algorithm

Choice of m

$m = n$ where n is the degree of the extension field

Gaudry's original algorithm

Choice of m

$m = n$ where n is the degree of the extension field

Relation step

- system of n polynomial equations in n variables, total degree 2^{n-1}
  - generically of dimension 0
  - standard techniques: Gröbner basis for lexicographic order
  - complexity is polynomial in $\log q$ but over-exponential in n
- Probability of decomposition as a sum of n points:

$$\frac{\#(\mathcal{F}^n/\mathfrak{S}_n)}{\#E(\mathbb{F}_{q^n})} \simeq \frac{q^n}{n!}\frac{1}{q^n} = \frac{1}{n!}$$

$\Rightarrow$ expected numbers of trials to get one relation is $n!$

- for a fixed n, complexity of the relation search step: $\tilde{O}(q)$

Gaudry's original algorithm

Linear algebra step

- sparse matrix : n non-zero entries per row
- complexity in $\tilde{O}(q^2)$ using Lanczos algorithm

$\Rightarrow$ total complexity of Gaudry's method in $\tilde{O}(q^2)$

Gaudry's original algorithm

Linear algebra step

- sparse matrix : n non-zero entries per row
- complexity in $\tilde{O}(q^2)$ using Lanczos algorithm

$\Rightarrow$ total complexity of Gaudry's method in $\tilde{O}(q^2)$

Improvement

- Thériault's "double large prime" technique: rebalance the complexity of the two steps
- final complexity in $\tilde{O}(q^{2-2/n})$
  $\longrightarrow$ better than generic methods for large q as soon as $n \geq 3$
- the hidden constant is huge and grows very fast with n
  $\longrightarrow$ not practically efficient

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2+t+1)$

- $E : y^2 = x^3 + (1+16t)x + (23+43t)$ s.t. $\#E = 10273$
- random points:
  $P = (71+85t, 82+47t)$, $Q = (81+77t, 61+71t)$
  $\rightarrow$ find x s.t. $Q = [x]P$

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2+t+1)$

- $E : y^2 = x^3 + (1+16t)x + (23+43t)$ s.t. $\#E = 10273$
- random points:
  $P = (71+85t, 82+47t)$, $Q = (81+77t, 61+71t)$
  $\rightarrow$ find x s.t. $Q = [x]P$
- random combination of P and Q:
  $R = [5962]P + [537]Q = (58+68t, 68+17t)$

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2+t+1)$

- $E : y^2 = x^3 + (1+16t)x + (23+43t)$ s.t. $\#E = 10273$
- random points:
  $P = (71+85t, 82+47t)$, $Q = (81+77t, 61+71t)$
  $\rightarrow$ find x s.t. $Q = [x]P$
- random combination of P and Q:
  $R = [5962]P + [537]Q = (58+68t, 68+17t)$
- use 3-rd "symmetrized" Semaev polynomial and Weil restriction:

$$(e_1^2 - 4e_2)x_R^2 - 2(e_1(e_2+a)+2b)x_R + (e_2-a)^2 - 4be_1 = 0$$

$$\Leftrightarrow \quad (32t+53)e_1^2 + (66t+86)e_1e_2 + (12t+49)e_1 + e_2^2 + (42t+89)e_2 + 88t + 45 = 0$$

$$\Leftrightarrow \quad \begin{cases} 53e_1^2 + 86e_1e_2 + 49e_1 + e_2^2 + 89e_2 + 45 = 0 \\ 32e_1^2 + 66e_1e_2 + 12e_1 + 42e_2 + 88 = 0 \end{cases}$$

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2 + t + 1)$

$$I = \langle 53e_1^2 + 86e_1e_2 + 49e_1 + e_2^2 + 89e_2 + 45, \\ 32e_1^2 + 66e_1e_2 + 12e_1 + 42e_2 + 88\rangle$$

- Gröbner basis of I for $lex_{e_1 > e_2}$:
  $G = \{e_1 + 86e_2^3 + 88e_2^2 + 58e_2 + 99, e_2^4 + 50e_2^3 + 85e_2^2 + 73e_2 + 17\}$

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2+t+1)$

$$I = \langle 53e_1^2 + 86e_1e_2 + 49e_1 + e_2^2 + 89e_2 + 45, \\ 32e_1^2 + 66e_1e_2 + 12e_1 + 42e_2 + 88\rangle$$

- Gröbner basis of I for $lex_{e_1>e_2}$:
  $G = \{e_1 + 86e_2^3 + 88e_2^2 + 58e_2 + 99, e_2^4 + 50e_2^3 + 85e_2^2 + 73e_2 + 17\}$
- $V(G) = \{(80,72),(97,68)\}$
  1. solution 1: $(e_1,e_2) = (80,72) \Rightarrow (x_{P_1}, x_{P_2}) = (5,75)$
     $\Rightarrow P_1 = (5, 89+71t); P_2 = (75, 57+74t)$ and $P_1 + P_2 = R$
  2. solution 2: $(e_1,e_2) = (97,68) \Rightarrow (x_{P_1}, x_{P_2}) = (19,78)$
     $\Rightarrow P_1 = (19, 35+9t); P_2 = (78, 75+4t)$ and $-P_1 + P_2 = R$

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2+t+1)$

$$I = \langle 53e_1^2 + 86e_1e_2 + 49e_1 + e_2^2 + 89e_2 + 45, \\ 32e_1^2 + 66e_1e_2 + 12e_1 + 42e_2 + 88\rangle$$

- Gröbner basis of I for $lex_{e_1>e_2}$:
  $G = \{e_1 + 86e_2^3 + 88e_2^2 + 58e_2 + 99, e_2^4 + 50e_2^3 + 85e_2^2 + 73e_2 + 17\}$
- $V(G) = \{(80,72),(97,68)\}$
  1. solution 1: $(e_1,e_2) = (80,72) \Rightarrow (x_{P_1},x_{P_2}) = (5,75)$
     $\Rightarrow P_1 = (5, 89+71t); P_2 = (75, 57+74t)$ and $P_1 + P_2 = R$
  2. solution 2: $(e_1,e_2) = (97,68) \Rightarrow (x_{P_1},x_{P_2}) = (19,78)$
     $\Rightarrow P_1 = (19, 35+9t); P_2 = (78, 75+4t)$ and $-P_1 + P_2 = R$
- How many relations ?
  $\#\mathcal{F} = 104 \Rightarrow 105$ relations needed

A toy example over $\mathbb{F}_{101^2} \simeq \mathbb{F}_{101}[t]/(t^2+t+1)$

$$I = \langle 53e_1^2 + 86e_1e_2 + 49e_1 + e_2^2 + 89e_2 + 45, \quad 32e_1^2 + 66e_1e_2 + 12e_1 + 42e_2 + 88\rangle$$

- Gröbner basis of I for $lex_{e_1>e_2}$:
  $G = \{e_1 + 86e_2^3 + 88e_2^2 + 58e_2 + 99, e_2^4 + 50e_2^3 + 85e_2^2 + 73e_2 + 17\}$
- $V(G) = \{(80,72),(97,68)\}$
  1. solution 1: $(e_1,e_2) = (80,72) \Rightarrow (x_{P_1},x_{P_2}) = (5,75)$
     $\Rightarrow P_1 = (5, 89+71t); P_2 = (75, 57+74t)$ and $P_1 + P_2 = R$
  2. solution 2: $(e_1,e_2) = (97,68) \Rightarrow (x_{P_1},x_{P_2}) = (19,78)$
     $\Rightarrow P_1 = (19, 35+9t); P_2 = (78, 75+4t)$ and $-P_1 + P_2 = R$
- How many relations ?
  $\#\mathcal{F} = 104 \Rightarrow 105$ relations needed
- Linear algebra $\rightarrow x = 85$

Drawbacks of the original algorithm

Complexity of the system resolution

$c(n,q)$ = cost of the resolution of a multivariate polynomial system of n equations of total degree 2^{n-1} in n variables over $\mathbb{F}_q$

1. Diem's analysis: ideal generically of dimension 0 and of degree $2^{n(n-1)}$
2. Resolution of with resultants:

$$c(n,q) \leq Poly(n!2^{n(n-1)} \log q)$$

3. Resolution with Gröbner basis and Faugère's algorithms (F4, F5):
   - can only marginally improve this upper-bound because of the degree of the ideal (cf FGLM complexity)
     $\rightarrow$ for $n = 5$, $\deg I = 2^{20}$ meaning we need to compute the roots of an univariate polynomial of degree 1048576
   - adding the field equations $x^q - x = 0$ is not practical for large q.

$\rightarrow$ **huge constant because of the resolution of the polynomial system**

Our variant

Choose $m = n - 1$

- compute the n-th summation polynomial instead of the $(n+1)$-th
- solve system of n equations in $(n-1)$ unknowns
- $(n-1)!q$ expected numbers of trials to get one relation

Computation speed-up

1. The system to be solved is generically **overdetermined**:
   - in general there is no solution over $\overline{\mathbb{F}_q}$: $I = \langle 1 \rangle$
   - exceptionally: very few solutions (almost always one) $\rightarrow$ the Gröbner basis of the ideal is composed of univariate polynomials of degree 1
2. Adapted techniques to solve the system:
   - once the Gröbner basis is computed for *degrevlex* the resolution of the system is immediate (FGLM not needed)
   - "F4-like" algorithm more convenient than F5

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

- $E : y^2 = x^3 + (44+52t+60t^2)x + (58+87t+74t^2)$, $\#E = 1029583$
- random points:

$P = (75+24t+84t^2, 61+18t+92t^2)$, $Q = (28+97t+35t^2, 48+64t+7t^2)$

$\rightarrow$ find x s.t. $Q = [x]P$

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

- $E : y^2 = x^3 + (44+52t+60t^2)x + (58+87t+74t^2)$, $\#E = 1029583$
- random points:
$P = (75{+}24t{+}84t^2, 61{+}18t{+}92t^2)$, $Q = (28{+}97t{+}35t^2, 48{+}64t{+}7t^2)$
  $\to$ find x s.t. $Q = [x]P$
- random combination of P and Q:
  $R = [236141]P + [381053]Q = (21+94t+16t^2, 41+34t+80t^2)$

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

- $E : y^2 = x^3 + (44+52t+60t^2)x + (58+87t+74t^2)$, $\#E = 1029583$
- random points:

$P = (75+24t+84t^2, 61+18t+92t^2)$, $Q = (28+97t+35t^2, 48+64t+7t^2)$
$\rightarrow$ find x s.t. $Q = [x]P$

- random combination of P and Q:
  $R = [236141]P + [381053]Q = (21+94t+16t^2, 41+34t+80t^2)$
- use 3-rd "symmetrized" Semaev polynomial and Weil restriction:

$$(e_1^2 - 4e_2)x_R^2 - 2(e_1(e_2+a)+2b)x_R + (e_2-a)^2 - 4be_1 = 0$$

$$\Leftrightarrow \quad (61t^2+78t+59)e_1^2 + (69t^2+14t+59)e_1e_2 + (40t^2+20t+57)e_1$$
$$+e_2^2 + (40t^2+89t+80)e_2 + 12t^2+11t+77 = 0$$

$$\Leftrightarrow \quad \begin{cases} 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77 = 0 \\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11 = 0 \\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12 = 0 \end{cases}$$

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

$$I = \langle 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77,\\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11,\\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12\rangle$$

- Gröbner basis of I for $\mathit{degrevlex}_{e_1>e_2}$:
  $G = \{e_1 + 32, e_2 + 26\}$

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

$$I = \langle 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77,\\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11,\\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12\rangle$$

- Gröbner basis of I for *degrevlex*$_{e_1>e_2}$:
  $G = \{e_1 + 32, e_2 + 26\}$
- $V(G) = \{(69, 75)\}$
  $(e_1, e_2) = (69, 75) \Rightarrow (x_{P_1}, x_{P_2}) = (6, 63)$
  $\Rightarrow P_1 = (6, 35 + 93t + 77t^2)$; $P_2 = (63, 2 + 66t + t^2)$ and
  $P_1 + P_2 = R$

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

$$I = \langle 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77,\\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11,\\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12\rangle$$

- Gröbner basis of I for $degrevlex_{e_1>e_2}$:
  $G = \{e_1 + 32, e_2 + 26\}$
- $V(G) = \{(69, 75)\}$
  $(e_1, e_2) = (69, 75) \Rightarrow (x_{P_1}, x_{P_2}) = (6, 63)$
  $\Rightarrow P_1 = (6, 35 + 93t + 77t^2)$; $P_2 = (63, 2 + 66t + t^2)$ and
  $P_1 + P_2 = R$
- How many relations ?
  $\#\mathcal{F} = 108 \Rightarrow 109$ relations needed

A toy example over $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$

$$I = \langle 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77,\\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11,\\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12\rangle$$

- Gröbner basis of I for *degrevlex*$_{e_1>e_2}$:
  $G = \{e_1 + 32, e_2 + 26\}$
- $V(G) = \{(69, 75)\}$
  $(e_1, e_2) = (69, 75) \Rightarrow (x_{P_1}, x_{P_2}) = (6, 63)$
  $\Rightarrow P_1 = (6, 35 + 93t + 77t^2)$; $P_2 = (63, 2 + 66t + t^2)$ and
  $P_1 + P_2 = R$
- How many relations ?
  $\#\mathcal{F} = 108 \Rightarrow 109$ relations needed
- Linear algebra $\rightarrow x = 370556$

Complexity of Gröbner basis computation

An available estimate of the complexity (Bardet, Faugère, Salvy)

Let $I = \langle f_1, \ldots, f_m \rangle \subset K[X_1, \ldots, X_n]$ be a zero-dimensional and semi-regular ideal, with $\mathbf{m} > \mathbf{n}$. Then the total number of field arithmetic operations performed by F5 is bounded by

$$O\left(\binom{n + d_{reg}}{n}^{\omega}\right)$$

where

- $\omega < 2.4$ (exponent in the complexity of matrix multiplication)
- degree of regularity d_{reg} smaller than the Macaulay bound $\displaystyle\sum_{i=1}^{m}(\deg f_i - 1) + 1$.

Analysis of the variant

Complexity of our variant

- Cost of the resolution with Bardet et al. estimate:

$$\tilde{O}\left(\binom{n2^{n-2}}{n-1}^{\omega}\right) = \tilde{O}\left(\left(2^{(n-1)(n-2)}e^n n^{-1/2}\right)^{\omega}\right)$$

- $(n-1)!q$ trials to get one relation and q relations needed

$$\Rightarrow \tilde{O}\left((n-1)!q^2\left(2^{(n-1)(n-2)}e^n n^{-1/2}\right)^{\omega}\right)$$

- Linear algebra step: $n-1$ non-zero entries per row $\Rightarrow \tilde{O}(nq^2)$ complexity $\rightsquigarrow$ negligible compared to the relation search

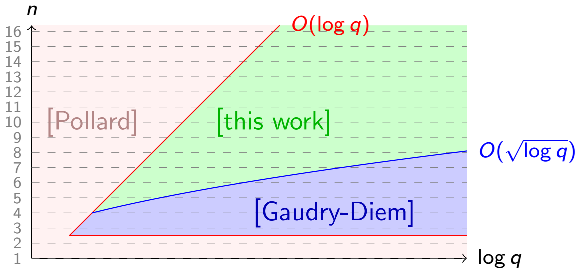
Complexity of our variant

Main result

Let E be an elliptic curve defined over $\mathbb{F}_{q^n}$, there exists an algorithm to solve the DLP in E with asymptotic complexity

$$\tilde{O}\left((n-1)!\left(2^{(n-1)(n-2)}e^n\,n^{-1/2}\right)^{\omega}q^2\right)$$

where $\omega \leq 2.4$ is the exponent in the complexity of matrix multiplication.



Main improvement

Reminder of Faugère's algorithms

- F4: complete reduction of the polynomials but many critical pairs reduced to zero $\Rightarrow$ computational waste
- F5: no reduction to zero (semi-regular system) but tails of polynomials not reduced $\Rightarrow$ number of critical pairs still not optimal

An "F4-like" algorithm without reduction to zero

- incremental nature of F5 less relevant for overdetermined systems
- key observation: all systems considered during the relation step have the same shape
- possible to remove all reductions to zero in latter F4 computations by observing the course of the first execution
- this approach gives better results than F5

Main improvement

Quick outline of the "F4-like" algorithm

1. Run a standard F4 algorithm on the first system, but:
   - at each iteration, store the list of selected critical pairs.
   - if there is a reduction to zero, remove the corresponding critical pair from the list
2. For each subsequent system, run a F4 computation but:
   - do not maintain nor update a queue of untreated pairs.
   - at each iteration, pick directly from the previously stored list the relevant pairs.

Second improvement

Symmetrized summation polynomials

- Semaev's summation polynomials are huge: $\deg_{X_i} f_m = 2^{m-2} \rightsquigarrow$ difficult to compute (even for $m = 5$, f_5 has 54777 monomials)
- rewriting $f_m(x_1, \ldots, x_m)$ in terms of the elementary symmetric polynomials is time-consuming
- faster and less memory-consuming to symmetrize between each resultant computation

Static Diffie Hellman problem

SDHP

G finite group, $P, Q \in G$ s.t. $Q = [d]P$ where d secret.

1. SDHP-solving algorithm $\mathcal{A}$:
   given P, Q and a challenge $X \in G \rightarrow$ outputs $[d]X$

Static Diffie Hellman problem

SDHP

G finite group, $P, Q \in G$ s.t. $Q = [d]P$ where d secret.

1. SDHP-solving algorithm $\mathcal{A}$:
   given P, Q and a challenge $X \in G \rightarrow$ outputs $[d]X$
2. "oracle-assisted" SDHP-solving algorithm $\mathcal{A}$:
   - learning phase:
     any number of queries $X_1, \ldots, X_l$ to an oracle $\rightarrow [d]X_1, \ldots, [d]X_l$
   - given a previously unseen challenge $X \rightarrow$ outputs $[d]X$

Static Diffie Hellman problem

SDHP

G finite group, $P, Q \in G$ s.t. $Q = [d]P$ where d secret.

1. SDHP-solving algorithm $\mathcal{A}$:
   given P, Q and a challenge $X \in G \rightarrow$ outputs $[d]X$
2. "oracle-assisted" SDHP-solving algorithm $\mathcal{A}$:
   - learning phase:
     any number of queries $X_1, \ldots, X_l$ to an oracle $\rightarrow [d]X_1, \ldots, [d]X_l$
   - given a previously unseen challenge $X \rightarrow$ outputs $[d]X$

From decomposition into $\mathcal{F}$ to oracle-assisted SDHP-solving algorithm

$\mathcal{F} = \{P_1, \ldots, P_l\}$

- learning phase: ask $Q_i = [d]P_i$ for $i = 1, \ldots, l$
- decompose the challenge X into the factor base: $X = \sum_i [c_i]P_i$
- answer $Y = \sum_i [c_i]Q_i$

Solving SDHP over $G = E(\mathbb{F}_{q^n})$

$\mathcal{F} = \{P \in E(\mathbb{F}_{q^n}) : P = (x_p, y_p), x_p \in \mathbb{F}_q\}$

An oracle-assisted SDHP-solving algorithm

1. learning phase: ask the oracle to compute $Q = [d]P$ for each $P \in \mathcal{F}$
2. self-randomization: given a challenge X, pick a random integer r coprime to the order of G and compute $X_r = [r]X$
3. check if X_r can be written as a sum of m points of $\mathcal{F}$: $X_r = \sum_{i=1}^m P_i$
4. if X_r is not decomposable, go back to step 2; else output $Y = [s]\left(\sum_{i=1}^m Q_i\right)$ where $s = r^{-1} \bmod |G|$.

Solving SDHP over $G = E(\mathbb{F}_{q^n})$

$\mathcal{F} = \{P \in E(\mathbb{F}_{q^n}) : P = (x_p, y_p), x_p \in \mathbb{F}_q\}$

An oracle-assisted SDHP-solving algorithm

1. learning phase: ask the oracle to compute $Q = [d]P$ for each $P \in \mathcal{F}$
2. self-randomization: given a challenge X, pick a random integer r coprime to the order of G and compute $X_r = [r]X$
3. check if X_r can be written as a sum of m points of $\mathcal{F}$: $X_r = \sum_{i=1}^{m} P_i$
4. if X_r is not decomposable, go back to step 2; else output $Y = [s]\left(\sum_{i=1}^{m} Q_i\right)$ where $s = r^{-1} \bmod |G|$.

Some complexities over $\mathbb{F}_{q^n}$

Degree of the extension field $\mathbb{F}_{q^n}$	$4\|n$	$5\|n$
Oracle calls	$O(q^{n/4})$	$O(q^{n/5})$
Decomposition cost	$Poly(\log q)$	$\tilde{O}(q^{n/5})$
Overall complexity	$O(q^{n/4})$	$\tilde{O}(q^{n/5})$