Index calculus methods over $E(\mathbb{F}_{q^n})$
Application to the static Diffie-Hellman problem

Vanessa VITSE - Antoine JOUX

Université de Versailles Saint-Quentin, Laboratoire PRISM

March 26, 2010

Hardness of DLP

Discrete logarithm problem (DLP)

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

$$h = g^x$$

Difficulty is related to the group:

1. Generic attack: complexity in $\Omega(\max(\alpha_i \sqrt{p_i}))$ if $\#G = \prod_i p_i^{\alpha_i}$
2. $G \subset (\mathbb{F}_q^*, \times)$: index calculus method with complexity in $L_q(1/3)$
3. $G \subset (J_{\mathcal{C}}(\mathbb{F}_q), +)$: index calculus method with sub-exponential complexity (depending of the genus $g > 1$)

Hardness of ECDLP

ECDLP

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

Specific attacks on few families of curves:

Transfer methods

- lift to characteristic zero fields: anomalous curves
- transfer to $\mathbb{F}_{q^k}^*$ via pairings: curves with small embedding degree
- Weil descent: transfer from $E(\mathbb{F}_{p^n})$ to $J_{\mathcal{C}}(\mathbb{F}_p)$ where $\mathcal{C}$ is a genus $g \geq n$ curve

Otherwise, only generic attacks

Trying an index calculus approach over $E(\mathbb{F}_{q^n})$

Basic outline

1. Choice of a factor base: $\mathcal{F} = \{P_1, \ldots, P_N\} \subset G$
2. Relation search: decompose $[a_i]P + [b_i]Q$ (a_i, b_i random) into $\mathcal{F}$

$$[a_i]P + [b_i]Q = \sum_{j=1}^{N} [c_{i,j}]P_j$$

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 $v = (v_1, \ldots, v_k) \in \ker({}^t M)$ such that $vA \neq 0\ [r]$
   - compute the solution of DLP: $x = -\left(\sum_i a_i v_i\right) / \left(\sum_i b_i v_i\right) \bmod r$

Results

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 $\log q > C.n$
- sub-exponential complexity when $n = \Theta(\sqrt{\log q})$
- impracticable as soon as $n > 4$

Results

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 $\log q > C.n$
- sub-exponential complexity when $n = \Theta(\sqrt{\log q})$
- impracticable as soon as $n > 4$

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 $\log q < c'.n^3 \log n$
- works for $n = 5$

Ingredients (1)

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
- convert the decomposition into a multivariate polynomial, but get rid of the variables y_{P_i} by using Semaev's summation polynomials

Ingredients (1)

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
- convert the decomposition into a multivariate polynomial, but get rid of the variables y_{P_i} by using Semaev's summation polynomials

Semaev's summation polynomials

Let E be an elliptic curve defined over K.
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

$E : y^2 = x^3 + ax + b$

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

Ingredients (2)

Weil restriction

- write $\mathbb{F}_{q^n}$ as $\mathbb{F}_q[t]/(f(t))$ where f irreducible of degree n
- convenient choice of $\mathcal{F} = \{P = (x, y) \in E(\mathbb{F}_{q^n}) : x \in \mathbb{F}_q, y \in \mathbb{F}_{q^n}\}$
  $\rightsquigarrow$ R given, find $x_{P_1}, \ldots, x_{P_m} \in \mathbb{F}_q,\ f_{m+1}(x_{P_1}, \ldots, x_{P_m}, x_R) = 0$

Ingredients (2)

Weil restriction

- write $\mathbb{F}_{q^n}$ as $\mathbb{F}_q[t]/(f(t))$ where f irreducible of degree n
- convenient choice of $\mathcal{F} = \{P = (x, y) \in E(\mathbb{F}_{q^n}) : x \in \mathbb{F}_q, y \in \mathbb{F}_{q^n}\}$
  $\rightsquigarrow$ R given, find $x_{P_1}, \ldots, x_{P_m} \in \mathbb{F}_q$, $f_{m+1}(x_{P_1}, \ldots, x_{P_m}, x_R) = 0$

Method

1. express the equation in terms of the elementary symmetric polynomials $e_1, \ldots, e_m$ of the variables $x_{P_1}, \ldots, x_{P_m}$
2. Weil restriction: sort according to the powers of t

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

3. solve the obtained 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

Complexity of the relation step

- Probability of decomposition as a sum of n points:

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

  $\rightsquigarrow$ about $n!$ trials give one relation

- each trial implies to solve over $\mathbb{F}_q$ a system of n polynomial equations in n variables, total degree 2^{n-1}, generically of dimension 0
  $\rightsquigarrow$ complexity is polynomial in $\log q$ but over-exponential in n

$\Rightarrow$ total complexity of the relation search step (n fixed): $\tilde{O}(q)$

Gaudry's original algorithm

First look at the total complexity

1. Relation step: $\tilde{O}(q)$ with constant exponential in n
2. Linear algebra step: find a vector in the kernel of a very sparse matrix $\rightsquigarrow$ complexity in $\tilde{O}(q^2)$ using Lanczos algorithm

$\Rightarrow$ Total complexity in $\tilde{O}(q^2)$

Gaudry's original algorithm

First look at the total complexity

1. Relation step: $\tilde{O}(q)$ with constant exponential in n
2. Linear algebra step: find a vector in the kernel of a very sparse matrix
$\rightsquigarrow$ complexity in $\tilde{O}(q^2)$ using Lanczos algorithm

$\Rightarrow$ Total complexity in $\tilde{O}(q^2)$

Improvement of the complexity

- rebalance the complexity of the two steps ("double large prime" technique)
- final complexity in $\tilde{O}(q^{2-2/n})$
$\rightarrow$ better than generic methods for large q as soon as $n \geq 3$

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

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

- 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

Analysis of the system resolution

$c(n, q)$ = cost of resolution over $\mathbb{F}_q$ of a system in n eq, n var, deg 2^{n-1}

Diem's analysis:

- ideal generically of dimension 0 and of degree $2^{n(n-1)}$
- resolution of with resultants: $c(n, q) \leq Poly(n!2^{n(n-1)} \log q)$

Drawbacks of the original algorithm

Analysis of the system resolution

$c(n,q) =$ cost of resolution over $\mathbb{F}_q$ of a system in n eq, n var, deg 2^{n-1}

Diem's analysis:

- ideal generically of dimension 0 and of degree $2^{n(n-1)}$
- resolution of with resultants: $c(n,q) \leq Poly(n! 2^{n(n-1)} \log q)$

Complexity of the system resolution with Gröbner basis

- compute a degrevlex Gröbner basis and use FGLM for ordering change

$$\underbrace{\tilde{O}\left(\left(2^{n(n-1)} e^n n^{-1/2}\right)^{\omega}\right)}_{\text{F5 algorithm}} + \underbrace{\tilde{O}\left(\left(2^{n(n-1)}\right)^3\right)}_{\text{FGLM}}$$

- adding the field equations $x^q - x = 0$ is not practical for large q.

Drawbacks of the original algorithm

Analysis of the system resolution

$c(n,q)$ = cost of resolution over $\mathbb{F}_q$ of a system in n eq, n var, deg 2^{n-1}

Diem's analysis:

- ideal generically of dimension 0 and of degree $2^{n(n-1)}$
- resolution of with resultants: $c(n,q) \leq Poly(n!2^{n(n-1)} \log q)$

Complexity of the system resolution with Gröbner basis

- compute a degrevlex Gröbner basis and use FGLM for ordering change

$$\underbrace{\tilde{O}\left(\left(2^{n(n-1)}e^n n^{-1/2}\right)^{\omega}\right)}_{\text{F5 algorithm}} + \underbrace{\tilde{O}\left(\left(2^{n(n-1)}\right)^3\right)}_{\text{FGLM}}$$

- adding the field equations $x^q - x = 0$ is not practical for large q.

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)
   - Gröbner basis computation with *degrevlex*, FGLM not needed
2. Adapted techniques to solve the system with an “F4-like” algorithm (more convenient than F4, F5 or hybrid approach)

Complexity of the Gröbner basis computation

Shape of the system

- system of n polynomials of degree 2^{n-2} in $n-1$ variables
- semi-regular with degree of regularity $d_{reg} \leq \sum_{i=1}^{m} (\deg f_i - 1) + 1$

Upper bound

- computation of the row echelon form of the d_{reg}-Macaulay matrix with at most $\binom{n-1+d_{reg}}{n-1}$ columns and smaller number of lines
- using fast reduction techniques, the complexity is at most

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

Total complexity of our variant

- Relation search step: $(n-1)!q$ 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
  $\Rightarrow$ complexity of $\tilde{O}(nq^2)$

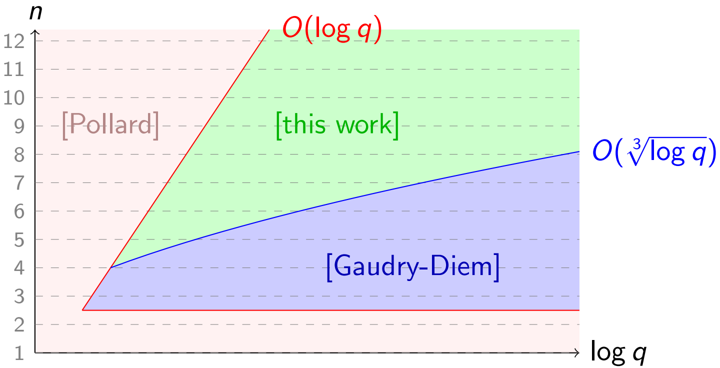
Main result

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

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

where ω is the exponent in the complexity of matrix multiplication.

Comparison of the three attacks of ECDLP over $\mathbb{F}_{q^n}$



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:

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

- random combination of P and Q:
  $R = [236141]P + [381053]Q = (21+94t+16t^2, 41+34t+80t^2)$

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:

$$
\begin{aligned}
& (e_1^2 - 4e_2)x_R^2 - 2(e_1(e_2+a)+2b)x_R + (e_2-a)^2 - 4be_1 = 0 \\
\Leftrightarrow\ & (61t^2+78t+59)e_1^2 + (69t^2+14t+59)e_1e_2 + (40t^2+20t+57)e_1 \\
& + e_2^2 + (40t^2+89t+80)e_2 + 12t^2 + 11t + 77 = 0 \\
\Leftrightarrow\ & \begin{cases} 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77 = 0 \\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11 = 0 \\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12 = 0 \end{cases}
\end{aligned}
$$

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 $\rightarrow x = 370556$

Comparison with hybrid approach

Applying hybrid approach

- trade-off between exhaustive search on some variables and Gröbner basis techniques
- one specialized variable $\rightsquigarrow$ compute q Gröbner bases of systems of n equations in $n-1$ variables
- but total degree of systems is 2^{n-1} vs 2^{n-2} in our approach

method	nb of systems	nb of eq	nb of var	total degree
Gaudry-Diem	$n!$	n	n	2^{n-1}
hybrid approach	$n!\,q$	n	$n-1$	2^{n-1}
this work	$(n-1)!\,q$	n	$n-1$	2^{n-2}

Adapted techniques to solve the system

Reminder of Faugère's algorithms

- F4: complete reduction of the polynomials but many critical pairs reduce to zero
- F5: no reduction to zero for semi-regular system but incomplete polynomial reductions may slow down future reductions

An "F4-like" algorithm without reduction to zero

- 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
- even if this algorithm is probabilist, it gives better results than F5 on the systems arising from index calculus methods

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 all polynomial multiples coming from the critical pairs
   - if there is a reduction to zero during the echelon computing phase, remove a well-chosen multiple from the stored list
2. For each subsequent system, run a F4 computation with the following modifications (`F4Remake`):
   - do not maintain nor update a queue of untreated pairs
   - at each iteration, pick directly from the previously stored list the relevant multiples

Practical results on $E(\mathbb{F}_{p^5})$

1. **Timings of F4/F4Remake**

$\lvert p\rvert_2$	estim. failure probability	F4Precomp	F4Remake	F4	Magma
8 bits	0.11	8.963	2.844	5.903	9.660
16 bits	4.4×10^{-4}	(19.07)	3.990	9.758	9.870
25 bits	2.4×10^{-6}	(32.98)	4.942	16.77	118.8
32 bits	5.8×10^{-9}	(44.33)	8.444	24.56	1046

2. **Comparison with F5**
   - F5 (homogenized system): computes 50% more labeled polynomials than F4
   - F5 (affine system): 600% more than F4!

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

An oracle-assisted SDHP-solving algorithm

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

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

An oracle-assisted SDHP-solving algorithm

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

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 $s = r^{-1} \bmod |G|$.

Remark

$P \in \mathcal{F} \Leftrightarrow -P \in \mathcal{F} \rightsquigarrow$ only $\#\mathcal{F}/2$ oracle calls are needed

Practical attacks of SDHP over $E(\mathbb{F}_{q^d})$

Extension degree 4 ($q^d = q'^4$) with Gaudry's approach

- $\simeq q'$ oracle calls needed
- self-randomization: average of 4! trials needed

Extension degree 5 ($q^d = q''^5$) with our approach

- $\simeq q''$ oracle calls needed
- self-randomization: average of $4!q''$ trials needed

Degree of the extension field $\mathbb{F}_{q^d}$	$4\|d$	$5\|d$
nb of oracle calls	$\simeq q^{d/4}$	$\simeq q^{d/5}$
decomposition cost	$\tilde{O}(1)$	$\tilde{O}(q^{d/5})$
overall complexity	$\tilde{O}(q^{d/4})$	$\tilde{O}(q^{d/5})$

Quid of $n > 5$?

Trade-off

1. decompose in a small number of points $R = P_1 + \ldots + P_m$
   - degree of $m+1$-Semaev in 2^{m-1}
2. enlarge the factor base $\mathcal{F}$
   - probability of decomposition not too small

Example for $n = 7$, $m = 3$, $\mathbb{F}_{q^7} = \mathbb{F}_q(t)$

$$\mathcal{F} = \{P \in E(\mathbb{F}_{q^7}) : x_P = x_{0,P} + x_{1,P}t, \quad x_{0,P}, x_{1,P} \in \mathbb{F}_q\}$$

Semaev + Weil descent $\rightsquigarrow$ 7 equations in 6 variables of degree 4 in each variables, total degree 12

Example for $n = 7$, $m = 3$, $\mathbb{F}_{q^7} = \mathbb{F}_q(t)$

Remarks

- polynomials no longer symmetric
- but invariant under the action of $\mathfrak{S}_3$

Example for $n = 7$, $m = 3$, $\mathbb{F}_{q^7} = \mathbb{F}_q(t)$

Remarks

- polynomials no longer symmetric
- but invariant under the action of $\mathfrak{S}_3$

How to take advantage of this invariance ?

- working in the invariant ring $\mathbb{F}_q[\underline{X}]^{\mathfrak{S}_3}$ is awkward
  - not a free algebra $\rightsquigarrow$ more variables and equations
  - in our example: 3 additional variables and 5 algebra relations
- SAGBI-Gröbner basis ?

Index calculus methods over $E(\mathbb{F}_{q^n})$ Application to the static Diffie-Hellman problem

Vanessa VITSE - Antoine JOUX

Université de Versailles Saint-Quentin, Laboratoire PRISM

March 26, 2010