Improving the Complexity of Index Calculus Algorithms in Elliptic Curves over Binary Fields

Jean-Charles Faugère[1] Ludovic Perret[1] Christophe Petit[2]
Guénaël Renault[1]

1: POLSYS project team, CNRS/INRIA/LIP6/UPMC, France
2: UCL Crypto Group, Belgium

Motivation

General Motivation

- Algebraic Cryptanalysis
- Identifying structures which help the solving step (computer algebra)

Elliptic Curve Discrete Logarithm (ECDLP)

$\Big\updownarrow$ Index Calculus (Semaev/Gaudry/Diem)

Polynomial System Solving (PoSSo)
with structures

Context: Solving the DLP

Discrete logarithm problem (DLP)

Given a finite cyclic group $\mathbb{G} = \langle g \rangle$ and $h \in \mathbb{G}$, find an integer k such that

$$h = [k]g = \underbrace{g + \ldots + g}_{k \text{ times}}$$

- **Generic algorithms** $O\left(\sqrt{\#\mathbb{G}}\right)$
  - Baby Step Giant Step, Pollard's rho, etc.
  - For any $\mathbb{G}$, **black box group**
- **index calculus** can be sub-exponential
  - sieving + linear algebra
  - $\mathbb{G} = (\mathbb{F}_q^{\times}, \times)$, $\mathbb{G} = (J_C(\mathbb{F}_q), +)$ with genus $g > 2$

☞ $\mathbb{G} = E(\mathbb{F}_q)$ **no sub-exponential** index calculus algorithm in general

Context: Index Calculus

Algorithm

Input : $P, Q \in \mathbb{G}$
Output : k such that $Q = [k]P$

- Factor basis: $\mathcal{F} = \{\pi_1, \ldots, \pi_s\}$, $s = \#\mathcal{F}$
- Sieving: decompose (if possible) $R = [a_j]P + [b_j]Q$ over $\mathcal{F}$ for many random (a_j, b_j)
- Linear Algebra: when at least $s+1$ relations are sieved, reduce them in order to find a (non trivial) relation between P and Q

$$\sum_j ([\lambda_j \cdot a_j]P + [\lambda_j \cdot b_j]Q) = 0$$

Complexity

- Balance between the sieving and linear algebra costs in function of s
- The existence of an efficient algorithm for decomposing over $\mathcal{F}$

Context: Diem's Variant of Index Calculus

- ☞ Semaev 04: introduce Summation Polynomials for decomposing points
- ☞ Gaudry 05: factor basis with a decomposition algo. (/PoSSo)
- ☞ Diem 05,11: generalization of Gaudry's approach

Algorithm (Diem's variant)

Input : $P, Q \in E(\mathbb{F}_{q^n})$, V a $\mathbb{F}_q$-vector space (dim $= n'$)
Output : x such that $Q = [x]P$

1. Factor basis: $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) \mid x \in V\}$
2. Sieving: $[a_j]P + [b_j]Q = P_1 + \cdots + P_m,\ P_i \in \mathcal{F}, \quad m \approx n/n'$
3. Linear algebra $\sum_j [\lambda_j \cdot a_j]P \oplus [\lambda_j \cdot b_j]Q = 0_{E(\mathbb{F}_{q^n})}$

Complexity

- SUBEXP in some cases (Diem 2011)
- When $q = 2$ the complexity is $\exp(O(n \log(n)^{1/2}))$

Point Decomposition Problem (PDP)

PDP(R)

Let be given

- $R \in E$
- $\mathcal{F}$ a factor basis of points in E

Find

- $P_1, \ldots, P_m \in \mathcal{F}$ such that $R = P_1 + \ldots + P_m$

☞ Modeling the problem as a polynomial system $\{\mathbf{g}_1, \ldots, \mathbf{g}_s\}$ and solve this system.

$$\begin{cases} (x_1, y_1) \in E, \ldots (x_m, y_m) \in E \\ (x_1, y_1) \oplus (x_2, y_2) = (r_1, t_1) \\ (r_1, t_1) \oplus (x_3, y_3) = (r_2, t_2) \\ \quad \vdots \\ (r_{n-2}, t_{n-2}) \oplus (x_n, y_n) = (R_x, R_y) \end{cases}$$

Recent Related Works

- **Elliptic curve discrete logarithm problem over small degree extension fields.** Joux, Vitse (*To appear in Journ. of Crypto.*)
  ☞Instantiation approach in PDP step

- **Using Symmetries in the Index Calculus for Elliptic Curves Discrete Logarithm.** Faugère, Gaudry, Huot, R. (*ePrint 2012/199*)
  ☞Specific structures identified + used ⇒ save an exp. factor

$\boxed{\text{Rely on Gaudry's variant}}$

Our Contribution

What about Diem's variant
in the extremal case of an ECDLP over $\mathbb{F}_{2^p}$ with p prime?

- Identify specific structures in this case
- Provide an ad-hoc algorithm
- Investigate complexity
  $\hookrightarrow$ Obtain a better one (heuristic)

Outline

Algebraic modelling of PDP: Summation polynomials

Semaev, Technical report 2004

☞Projection of the PDP(R=0) on the $\{x_1, \ldots, x_m\}$

$$\text{PDP: } \langle \mathbf{g}_1(x_1, \ldots, x_m, y_1, \ldots, y_m), \ldots, \mathbf{g}_s(x_1, \ldots, x_m, y_1, \ldots, y_m) \rangle$$

$$\Big\downarrow \text{ Elimination (Resultant, Gröbner basis)}$$

$$\text{Summation: } \langle \mathbf{f}_m(x_1, \ldots, x_m) \rangle = \langle \mathbf{g}_1, \ldots, \mathbf{g}_s \rangle \cap \mathbb{F}_{q^n}[x_1, \ldots, x_m]$$

$$\deg_{x_i}(\mathbf{f}_m) = 2^{m-2}$$

Characterization

$$\mathbf{f}_m(x_1, ..., x_m) = 0$$

$$\Updownarrow$$

$$\exists (P_1, ..., P_m) \in E(\overline{\mathbb{K}})^m \text{ s.t. } \forall i, (P_i)_x = x_i \text{ and } P_1 + \cdots + P_m = 0$$

Algebraic modelling of PDP: Summation polynomials

Semaev, Technical report 2004

☞Projection of the PDP(R=0) on the $\{x_1, \ldots, x_m\}$

$$\text{PDP: } \langle \mathbf{g}_1(x_1, \ldots, x_m, y_1, \ldots, y_m), \ldots, \mathbf{g}_s(x_1, \ldots, x_m, y_1, \ldots, y_m) \rangle$$

$$\Big\downarrow \text{ Elimination (Resultant, Gröbner basis)}$$

$$\text{Summation: } \langle \mathbf{f}_m(x_1, \ldots, x_m) \rangle = \langle \mathbf{g}_1, \ldots, \mathbf{g}_s \rangle \cap \mathbb{F}_{q^n}[x_1, \ldots, x_m]$$

$$\deg_{x_i}(\mathbf{f}_m) = 2^{m-2}$$

Application in Index Calculus

Solving PDP(R) with factor basis $\mathcal{F} = \{(x, y) \in E(\mathbb{F}_{q^n}) \mid x \in V\}$.

$$\Updownarrow$$

Finding $(x_1, \ldots, x_m) \in V^m$ s.t. $\mathbf{f}_{m+1}(x_1, ..., x_m, R_x) = 0$

Solving equations with vectorial constraint

General problem

Let $\mathbf{f}(x_1,\ldots,x_m) \in \mathbb{F}_{2^n}[x_1,\ldots,x_m]$ and $V = \langle \nu_1,\ldots,\nu_{n'} \rangle \subset \mathbb{F}_{2^n}$ an n'-dim $\mathbb{F}_2$-vect with $mn' \approx n$. Find the solutions of $\mathbf{f}$ in V^m.

☞Weill restriction of scalars in two steps!

1. Change variables: $x_i = \nu_1 t_{i,1} + \ldots + \nu_{n'} t_{i,n'}$.

   $\mathbf{f}_V(t_{1,1},\ldots,t_{m,n'}) = 0$ with $\mathbf{f}_V \in \mathbb{F}_{2^n}[t_{1,1},\ldots,t_{m,n'}]/\langle t_{i,j}^2 - t_{i,j} \rangle$
2. Usual scalar restriction: $\{\omega_1,\ldots,\omega_n\}$ be a $\mathbb{F}_2$-basis of $\mathbb{F}_{2^n}$

   $$\mathbf{f}_V = \varphi_1(\mathbf{f}_V)\omega_1 + \cdots + \varphi_n(\mathbf{f}_V)\omega_n,\ \varphi_i(\mathbf{f}_V) \in \mathbb{F}_2[t_{i,j}]/\langle t_{i,j}^2 - t_{i,j} \rangle$$

General problem equivalent to solve

$$\varphi_1(\mathbf{f}_V) = \cdots = \varphi_n(\mathbf{f}_V) = 0 \text{ over } \mathbb{F}_2$$

Solving equations with vectorial constraint by linearization

Polynomial system model

Rational solutions of $\mathcal{S}_{alg}$: $\{\varphi_1(\mathbf{f}_V), \ldots, \varphi_n(\mathbf{f}_V)\} \subset \mathbb{F}_2[t_{1,1}, \ldots, t_{m,n'}]$

☞Naive method: linearization

- We consider many $\mathbf{mf}$ with $\mathbf{m} = \prod_{i=1}^{m} x_i^{e_i}$
- We add $\varphi_1((\mathbf{mf})_V), \ldots, \varphi_n((\mathbf{mf})_V)$ in $\mathcal{S}_{alg}$
- We construct a linear system $\mathcal{S}_{lin}$ from $\mathcal{S}_{alg}$ (Macaulay matrix)

$$\begin{array}{cc} & \text{monomials in } \mathcal{S}_{alg} \\ \begin{array}{c} \vdots \\ \cdots + c_m^i \mathbf{t}_i + c_m^j \mathbf{t}_j + \cdots = \varphi_k(\mathbf{mf}) \end{array} & \begin{pmatrix} \ldots & c_m^i & \ldots & c_m^j & \ldots \end{pmatrix} \end{array}$$

$\boxed{\text{Is there any linear dependencies between the } \mathbf{m_i f}\text{'s ?}}$

Linear Dependencies

☞Frobenius transform is linear

- $\varphi_i(\mathbf{g}_V^2) = \sum_j \alpha_j \varphi_i(\mathbf{g}_V)$
  $\hookrightarrow$ avoid $\mathbf{m}$ such that $\mathbf{m} = LT(\mathbf{m}'\mathbf{f})$ for a preceding $\mathbf{m}'$.
- $\{x_1^{2^1}, x_1^{2^2}, \ldots, x_1^{2^{n'+1}}\} \subset$ vect. space of dim. n'
  $\hookrightarrow$ consider monomials $\mathbf{m} = \prod_{i=1}^m x_i^{e_i}$ with $e_i \leq 2^{n'}$

Assumption

If we choose the monomials outside the set we identified here all the algebraic equations in $\mathcal{S}_{alg}$ are linearly independant.

Under this assumption, it is now possible to evaluate the number of columns/rows of the smallest square Macaulay matrix.

Intrinsic Structures

$$\mathbf{g} = \mathbf{mf}, \quad \mathbf{g}(\boxed{x_1}, \ldots, \boxed{x_j}, \ldots)$$

$$\varphi_i(\mathbf{g}_V)(\boxed{t_{1,1}, \ldots, t_{1,n'}}, \ldots, \boxed{t_{j,1}, \ldots, t_{j,n'}}, \ldots) \qquad \mod \langle t_{i,j}^2 + t_{i,j} \rangle$$

Block Affine Multilinear

Let $k = \mathrm{Max}_i(\log_2(\deg_{x_i}(\mathbf{f})))$ and $\mathbf{m} = \prod_{i=1}^{r} x_i^{e_i}$

☞Due to field equations, $\varphi_i(\mathbf{mf})$ are affine multilinear

☞Deg of $\varphi_i(\mathbf{mf})$ w.r.t $X_i = \{t_{i,1}, \ldots, t_{i,n'}\}$ is $\leq \max_{0 \leq e_i' \leq 2^k} \mathrm{HW}(e_i + e_i')$

$\hookrightarrow$ MonLinB(d) = { multilinears monomials of degree $\leq d$ in each X_i}

$\hookrightarrow$ control the number $E(d)$ of monomials

affine multinear: $t_1 t_2 t_4 t_5 + t_1 t_7 + 1$

Intrinsic Structures

Block Affine Multilinear

Let $k = \mathrm{Max}_i(\log_2(\deg_{x_i}(\mathbf{f})))$ and $\mathbf{m} = \prod_{i=1}^{r} x_i^{e_i}$

☞Due to field equations, $\varphi_i(\mathbf{mf})$ are affine multilinear

☞Deg of $\varphi_i(\mathbf{mf})$ w.r.t $X_i = \{t_{i,1}, \ldots, t_{i,n'}\}$ is $\leq \max_{0 \leq e'_i \leq 2^k} \mathrm{HW}(e_i + e'_i)$

$\hookrightarrow$ MonLinB(d) = { multilinears monomials of degree $\leq d$ in each X_i}

$\hookrightarrow$ control the number $E(d)$ of monomials

☞ MonLinB(d) $\subset\subset$ monomials of total degree d^m.

$M(d) = \#$MonLinB(d)

$n \cdot E(d) \quad \begin{pmatrix} c_m^i \ldots c_m^j \end{pmatrix}$

Macaulay

Intrinsic Structures

Block Affine Multilinear

Let $k = \mathrm{Max}_i(\log_2(\deg_{x_i}(\mathbf{f})))$ and $\mathbf{m} = \prod_{i=1}^{r} x_i^{e_i}$

☞Due to field equations, $\varphi_i(\mathbf{mf})$ are affine multilinear

☞Deg of $\varphi_i(\mathbf{mf})$ w.r.t $X_i = \{t_{i,1}, \ldots, t_{i,n'}\}$ is $\leq \max_{0 \leq e_i' \leq 2^k} \mathrm{HW}(e_i + e_i')$

$\hookrightarrow$ MonLinB(d) = { multilinears monomials of degree $\leq d$ in each X_i}

$\hookrightarrow$ control the number $E(d)$ of monomials

☞ MonLinB(d) $\subset\subset$ monomials of total degree d^m.

$\boxed{n \cdot E(d) \geq M(d)?}$

$$\left(\begin{array}{cc} & M(d) = \#\text{MonLinB}(d) \\ n \cdot E(d) & \begin{pmatrix} c_m^i \ldots c_m^j \end{pmatrix} \\ & \text{Macaulay} \end{array} \right)$$

Complexity results

Solving equations with linear constraints

$M(d) = \left(\sum_{d'=0}^{d} \binom{n'}{d'}\right)^m$ and $E(d) = 2^{tm}\left(\sum_{d'=t}^{d} \binom{n'-t}{d'-t}\right)^m$, thus

$$n \cdot E(d) \geq M(d) \text{ as soon as } d \approx \frac{n'}{2}$$

assumption of linear independency $\Rightarrow O(2^{\omega\, n/2})$

☞In the application to ECDLP here, the sieving step is dominant

Solving the ECDLP over $\mathbb{F}_{2^n}$ with index calculus

Under assumption of linear independency the complexity is bounded by

$$O(2^{\omega\, n/2})$$

Outline

1. Main Result

2. Experimental results and Conclusion

Experiments: Validating the assumption

Fact

A random Boolean matrix of size $(M+5) \times M$ has rank M or $M-1$ or $M-2$ with proba $\approx 99.9\%$.

Results (binary fields $< 2^{40}$)

- For random polynomials f with degree $< 2^{m-1}$ in each of its $m < 5$ variables.
- Semaev's summation polynomials (evaluate) $m = 2, \ldots, 4$.

The test was repeated 100 times for each examples. The proba. is always $\approx 100\%$.

$$\begin{array}{cc} & \overset{M(d)}{\longleftrightarrow} \\ n \cdot E(d) \; \updownarrow & \begin{pmatrix} ; \\ c_m^i \cdots c_m^j \\ ; \end{pmatrix} \end{array}$$

$$\begin{pmatrix} (M(d)+5) \times M(d) \end{pmatrix}$$

Conclusion

☞Structures are identified!

↪ Ad-hoc linearization algorithm

↪ Better complexity result!

☞Linearization: first step in a PoSSo study

↪ Preleminary experiments with Gröbner show better performances.

n	m	Number of Operations (GB)	Theoretical bound
41	2	$2^{23.5}$	$M(d)^2 \approx 2^{60}$
67	2	$2^{37.1}$	$M(d)^2 \approx 2^{90}$
97	2	$2^{51.1}$	$M(d)^2 \approx 2^{125}$
131	2	$2^{74.5}$	$M(d)^2 \approx 2^{160}$

Conclusion

☞Structures are identified!

$\hookrightarrow$ Ad-hoc linearization algorithm

$\hookrightarrow$ Better complexity result!

☞Linearization: first step in a PoSSo study

$\hookrightarrow$ Preleminary experiments with Gröbner show better performances.

We obtain a better complexity result but still worst than exhaustive search... Nonetheless, we give some indication that these polynomial systems are easier than one might expect at first!

Conclusion

☞Structures are identified!

$\hookrightarrow$ Ad-hoc linearization algorithm

$\hookrightarrow$ Better complexity result!

☞Linearization: first step in a PoSSo study

$\hookrightarrow$ Preleminary experiments with Gröbner show better performances.

We obtain a better complexity result but still worst than exhaustive search...
Nonetheless, we give some indication that these polynomial systems are easier than one might expect at first!

Is ECDLP($\mathbb{F}_{2^n}$) SUBEXP?

Conclusion

☞Structures are identified!

$\hookrightarrow$ Ad-hoc linearization algorithm

$\hookrightarrow$ Better complexity result!

☞Linearization: first step in a PoSSo study

$\hookrightarrow$ Preleminary experiments with Gröbner show better performances.

Is ECDLP($\mathbb{F}_{2^n}$) SUBEXP?

Conclusion

☞Structures are identified!

$\hookrightarrow$ Ad-hoc linearization algorithm

$\hookrightarrow$ Better complexity result!

☞Linearization: first step in a PoSSo study

$\hookrightarrow$ Preleminary experiments with Gröbner show better performances.

Semaev summation polynomials are very particular!

$\hookrightarrow$ Can not apply usual theoretical/heuristical results in a generic way

$\hookrightarrow$ Pitfall of linear dependency!

$\hookrightarrow$ Too small experiments for interpolating a better complexity!

Conclusion, future works

Semaev summation polynomials are very particular!

$\hookrightarrow$ Can not apply usual theoretical/heuristical results in a generic way

$\hookrightarrow$ Pitfall of linear dependency!

$\hookrightarrow$ Too small experiments for interpolling a better complexity!

☞Semaev summation polynomials contain many more structures!

Using these structures is the only way to progress

$\hookrightarrow$ To handle larger examples (at least $m = 5, 6$)

$\hookrightarrow$ To provide theoretical results about <u>degree of regularity</u>