

Improving NFS for the discrete logarithm problem in non-prime finite fields

Razvan Barbulescu, Pierrick Gaudry, *Aurore Guillevic*, François Morain

Institut national de recherche en informatique et en automatique (INRIA)

École Polytechnique/LIX

Centre national de la recherche scientifique (CNRS)

Université de Lorraine

Eurocrypt 2015, April 27th

Our Work

- \mathbf{F}_{p^2} : target group of pairing-based cryptosystems
 - Record computation of a Discrete Logarithm (DL) in \mathbf{F}_{p^2} of 600 bits ($\log_2 p = 300$ bits)
 - DL in \mathbf{F}_{p^2} is 260 times faster than DL in $\mathbf{F}_{p'}$ of same size
- *serious consequences for pairing-based crypto*
- source code: <http://cado-nfs.gforge.inria.fr/>

Context : Discrete logarithm problem (DLP) in $\mathbf{F}_{p^n}^*$

In a subgroup $\langle g \rangle$ of $\mathbf{F}_{p^n}^*$ of order ℓ ,

- $(g, x) \mapsto g^x$ is easy (polynomial time)
- $(g, g^x) \mapsto x$ is (in well-chosen subgroup) hard: DLP.

In our work:

- We attack DL in \mathbf{F}_{p^2} , starting point of $\mathbf{F}_{p^3}, \mathbf{F}_{p^4}, \dots, \mathbf{F}_{p^{12}}$
- p is large: quasi polynomial time algo. does NOT apply
- DLP in these \mathbf{F}_{p^n} still asymptotically as hard as in the 90's
- consequences for pairing-based crypto: \mathbf{F}_{p^2} target group

Practical improvements and new asymptotic complexities

L -notation: $Q = p^n$, $L_Q[1/3, c] = e^{(c+o(1))(\log Q)^{1/3} (\log \log Q)^{2/3}}$ for $c > 0$.

- DL in \mathbf{F}_{p^n} , small n , large p : complexity in $L_{p^n}[1/3, 1.92]$ (as for RSA modulus factorization) since the 90's
- $n \geq 2$: **two new polynomial selection methods**
- **great improvements in practice**
- **record of 600 bits**

Bonus: asymptotic complexity improvements in medium characteristic case

$\alpha = 1/3$	c , previous work	c , our work
DL in \mathbf{F}_{p^n} , $p = L_Q(2/3, c')$	1.92 < c < 2.42 ✗	1.74 ✓
DL in \mathbf{F}_{p^n} , medium p	2.42 ✗	2.20 ✓

MNFS variants: see [Pierrot15], Eurocrypt 2015.

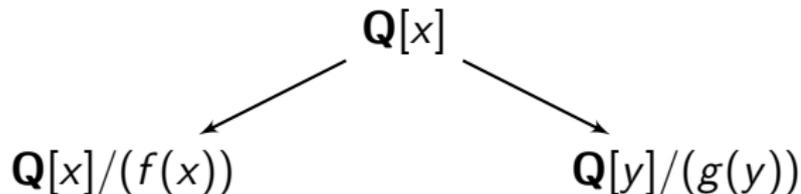
Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x)$ \rightarrow define number fields K_f, K_g .

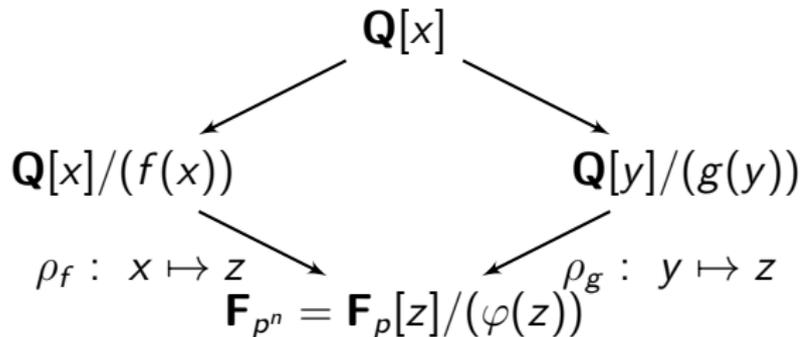
Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x) \rightarrow$ define number fields K_f, K_g .



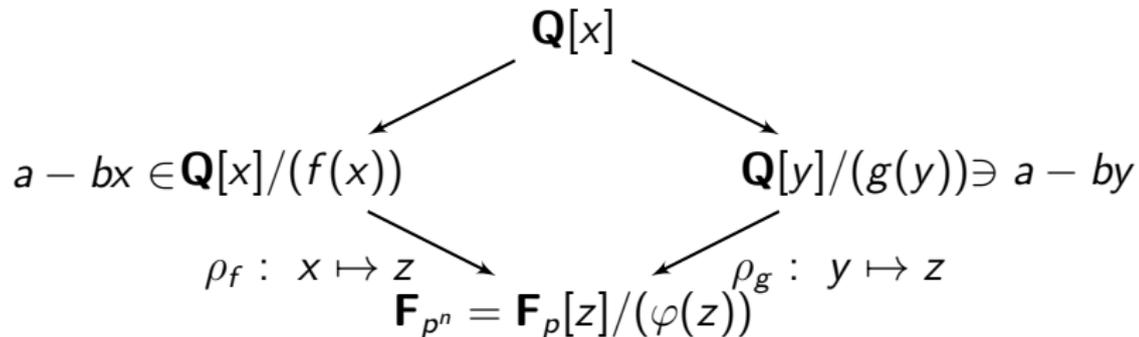
Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x) \rightarrow$ define number fields K_f, K_g .



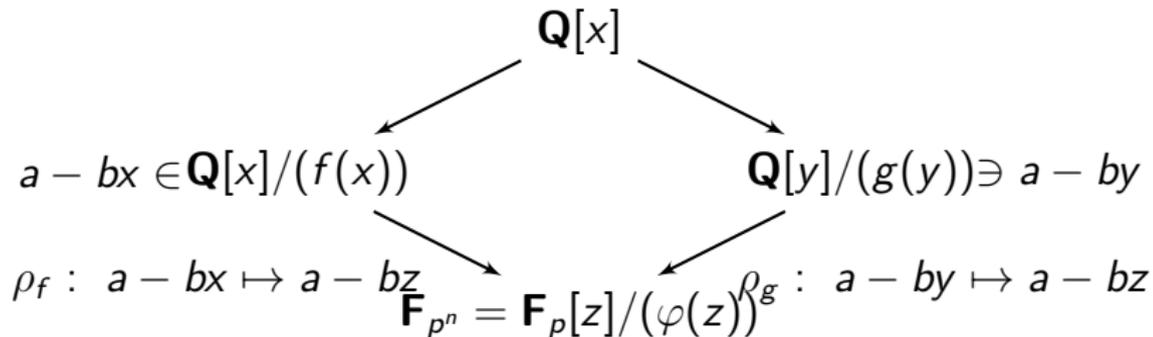
Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x) \rightarrow$ define number fields K_f, K_g .
2. *Relation collection* between ideals of each number field.



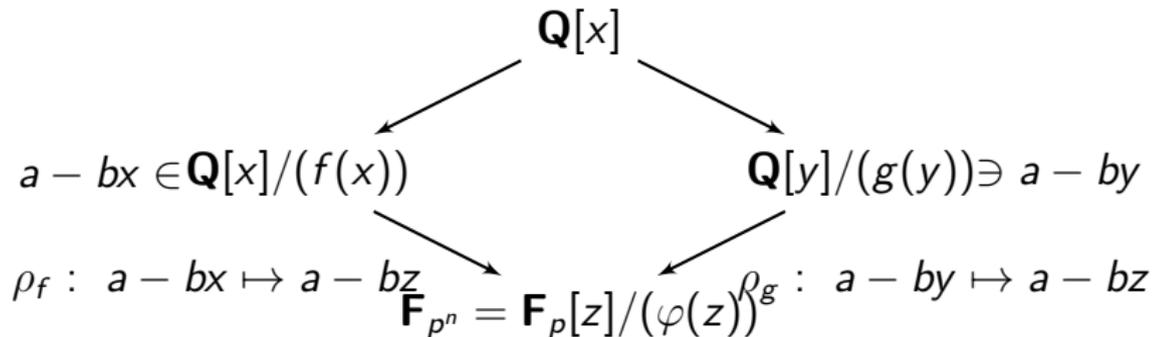
Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x) \rightarrow$ define number fields K_f, K_g .
2. *Relation collection* between ideals of each number field.



Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x)$ \rightarrow define number fields K_f, K_g .
2. *Relation collection* between ideals of each number field.

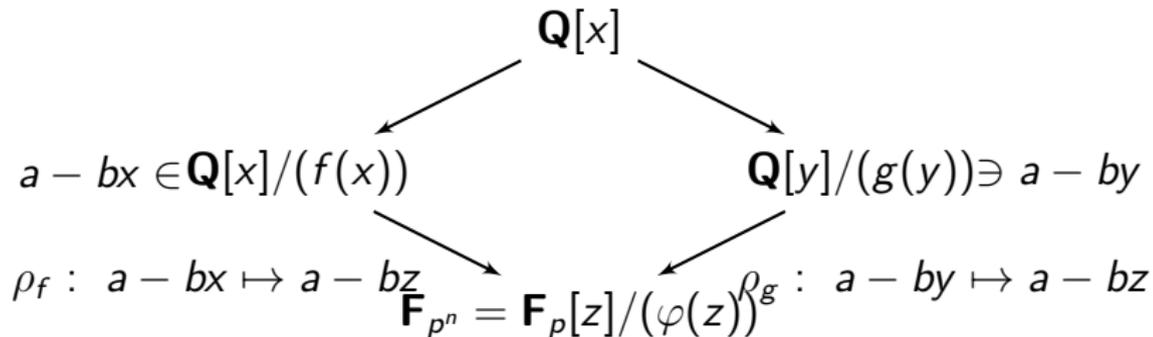


3. *Linear algebra modulo $\ell \mid p^n - 1$.*

\rightarrow here we know the discrete log of a subset of ideals of K_f, K_g .

Number Field Sieve algorithm for DL in \mathbf{F}_{p^n}

1. *Polynomial Selection*: compute $f(x)$, $g(x) \rightarrow$ define number fields K_f, K_g .
2. *Relation collection* between ideals of each number field.



3. *Linear algebra modulo $\ell \mid p^n - 1$.*
 \rightarrow here we know the discrete log of a subset of ideals of K_f, K_g .
4. *Individual Logarithm.*

Relation collection

We need a high smoothness probability of

- ideals $(a - bx) \in K_f$, $(a - by) \in K_g$, $|a|, |b| < E$
- integers $Norm_{K_f/\mathbb{Q}}(a - bx)$ and $Norm_{K_g/\mathbb{Q}}(a - by)$
- we approximate $|Norm_{K_f/\mathbb{Q}}(a - bx)| \leq E^{\deg f} \|f\|_\infty$ with $\|f\|_\infty = \max_{1 \leq i \leq \deg f} |f_i|$
- we want to minimize the product of norms:

$$E^{\deg f} \|f\|_\infty E^{\deg g} \|g\|_\infty$$

We need

- f, g of small degrees
- f, g of small coefficients

Relation collection

We need a high smoothness probability of

- ideals $(a - bx) \in K_f$, $(a - by) \in K_g$, $|a|, |b| < E$
- integers $Norm_{K_f/\mathbb{Q}}(a - bx)$ and $Norm_{K_g/\mathbb{Q}}(a - by)$
- we approximate $|Norm_{K_f/\mathbb{Q}}(a - bx)| \leq E^{\deg f} \|f\|_{\infty}$ with $\|f\|_{\infty} = \max_{1 \leq i \leq \deg f} |f_i|$
- we want to minimize the product of norms:

$$E^{\deg f} \|f\|_{\infty} E^{\deg g} \|g\|_{\infty}$$

We need

- f, g of small degrees
- f, g of small coefficients

We cannot have both, we need to balance degrees and coefficient sizes.

A. Generalized Joux-Lercier method

Simplified version: $\deg f = n + 1$, $\deg g = n$

1. choose f , $\deg f = n + 1$, s.t.
2. $f \equiv \tilde{f}\varphi \pmod{p}$, φ a monic irreducible factor of degree n modulo p

$$\varphi(x) = \varphi_0 + \varphi_1 x + \cdots + x^n$$

3. Reduce the following matrix using LLL

$$M = \left[\begin{array}{cccc} p & & & \\ & \ddots & & \\ & & p & \\ \varphi_0 & \varphi_1 & \cdots & 1 \end{array} \right] \left. \begin{array}{l} \deg \varphi = \\ n \text{ rows} \\ \\ \\ 1 \text{ row} \end{array} \right\} \rightarrow \text{LLL}(M) = \left[\begin{array}{cccc} g_0 & g_1 & \cdots & g_n \\ & & & \\ & & * & \\ & & & \end{array} \right]$$

4. $g = g_0 + g_1 x + \cdots + g_n x^n$, $\|g\|_\infty = O(p^{n/(n+1)})$

$$E^{\deg f + \deg g} \|f\|_\infty \|g\|_\infty = E^{2n+1} O(p^{n/(n+1)})$$

A. Generalized Joux-Lercier method: example

- $p = 10000000019$ and $n = 2$
- $f = x^3 + x + 1$
- $\varphi = x^2 + 3402015304x + 6660167027$
- $M = \begin{bmatrix} p & & \\ & p & \\ \varphi_0 & \varphi_1 & 1 \end{bmatrix} \xrightarrow{\text{LLL}} g = 746193x^2 + 914408x + 4935648$
- $\|f\|_\infty = O(1)$, $\|g\|_\infty = O(p^{2/3})$

Historical remark:

- this construction appears in Barbulescu PhD thesis (2013)
- In January we were told about Matyukhin's work [МАТЮХИН 2006]:
ЭФФЕКТИВНЫЙ ВАРИАНТ МЕТОДА РЕШЕТА ЧИСЛОВОГО ПОЛЯ ДЛЯ
ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ В ПОЛЕ $GF(p^k)$.

B. The Conjugation Method for F_{p^2} : example

1. $p = 7 \bmod 8$
2. $f = x^4 + 1$ irreducible over \mathbf{Z} , small
3. $f = (x^2 + \sqrt{2}x + 1)(x^2 - \sqrt{2}x + 1)$ over $\mathbf{Q}(\sqrt{2})$
4. $x^2 - 2$ has two roots $\pm r \bmod p$
5. $\varphi = x^2 + rx + 1$ is irreducible over F_p since $p \equiv 7 \bmod 8$, and over \mathbf{Z}
6. compute (u, v) s.t. $u/v \equiv r \bmod p$, with $|u|, |v| \sim p^{1/2}$ with the *rational reconstruction* method
7. $g = vx^2 + ux + v \equiv v \cdot \varphi \bmod p$

Generalize to higher n :

- $\deg f = 2n$, $\deg g = n$, $\|f\|_\infty = O(1)$, $\|g\|_\infty = O(p^{1/2})$

$$E^{\deg f + \deg g} \|f\|_\infty \|g\|_\infty = E^{3n} O(p^{1/2})$$

Our Record: Discrete Logarithm in \mathbb{F}_{p^2} of 600 bits

$$\begin{aligned}
 p &= 314159265358979323846264338327950288419716939 \backslash \\
 (300 \text{ bits}) & \quad 937510582097494459230781640628620899877709223 \\
 p + 1 &= 8 \cdot \ell \\
 \ell &= 392699081698724154807830422909937860524646174 \backslash \\
 (295 \text{ bits}) & \quad 92188822762186807403847705078577612484713653 \\
 p - 1 &= 6 \cdot h_0 \text{ with } h_0 \text{ a 295 bit prime}
 \end{aligned}$$

- Cryptographic subgroup: G of order ℓ
- For our record: $Q = p^2$, $\log_2 Q = 600$, optimal value of E around $\log_2 E = 27$ bits.

Our Record: Discrete Logarithm in \mathbf{F}_{p^2} of 600 bits

Polynomial selection:

- Generalized Joux Lercier: $f = x^3 + x + 1$, $\|g\|_\infty = O(p^{2/3})$, Norms bounded by $E^5 p^{2/3}$ of 339 bits **X**
- Conjugation: $f = x^4 + 1$, $\|g\|_\infty = O(p^{1/2})$, Norms bounded by $E^6 p^{1/2}$ of 317 bits **→22 bits less ✓**

$$f = x^4 + 1$$

$$g = 448225077249286433565160965828828303618362474 x^2 \\ - 296061099084763680469275137306557962657824623 x \\ + 448225077249286433565160965828828303618362474 .$$

$$\|g\|_\infty = 150 \text{ bits}$$

$$\varphi = x^2 + yx + 1, \quad \log_2 y = \log_2 p$$

Target:

$$s = \lfloor (\pi(2^{298})/8) \rfloor x + \lfloor (\gamma \cdot 2^{298}) \rfloor \in \mathbf{F}_{p^2} = \mathbf{F}_p[x]/(\varphi(x))$$

$$gen = x + 2$$

Speed-up of Relation Collection and Linear Algebra

- Galois automorphism: $x \mapsto 1/x$ both for $f = x^4 + 1$ and $g = vx^2 + ux + v$
- $a - bx \mapsto -b + ax$: a second relation for free
- speed-up by a factor 2 for relation collection
- speed-up by a factor 4 for linear algebra
- others important algebraic simplification and speed-up

Finally,

$$\log_{gen} s \equiv 276214243617912804300337349268306605403758173 \backslash \\ 81941441861019832278568318885392430499058012 \pmod{\ell}.$$

Record running-time comparison in years for 600-bit inputs

Algorithm	relation collection	linear algebra	total	
NFS Integer Factorization	5y	0.5y	5.5y	×11
NFS DL in \mathbf{F}_p	50y	80y	130y	×260
This work: NFS DL in \mathbf{F}_{p^2}	0.4y	0.05y (GPU)	0.5y	×1

DL in \mathbf{F}_{p^2} < Integer Factorization < DL in \mathbf{F}_p

- Paper: <https://hal.inria.fr/hal-01112879>
- Algebraic secrets: <https://hal.inria.fr/hal-01052449>
- **Source code:** <http://cado-nfs.gforge.inria.fr/>
- ➔ **Download it and solve your own DL in \mathbf{F}_{p^2}**
- *Stay tuned for more records during summer.*