Individual Discrete Logarithm in GF(p^k)
(last step of the Number Field Sieve algorithm)

Aurore Guillevic

INRIA Saclay / GRACE Team

École Polytechnique / LIX

Asiacrypt 2015 Conference, Auckland, New Zealand, November 30

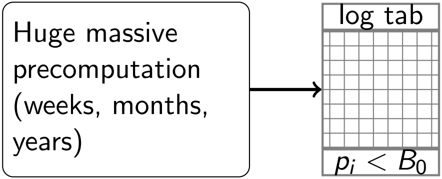
Logjam attack (weakdh.org)

Solving actual practical problem:
Given a **fixed** finite field GF(q),

Huge massive precomputation (weeks, months, years)

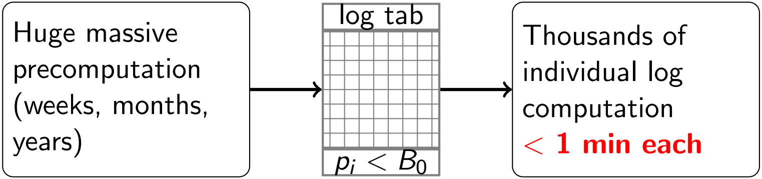
Logjam attack (weakdh.org)

Solving actual practical problem:
Given a **fixed** finite field GF(q),



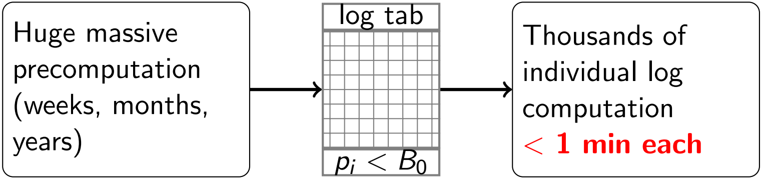
Logjam attack (weakdh.org)

Solving actual practical problem:
Given a **fixed** finite field GF(q),



Logjam attack (weakdh.org)

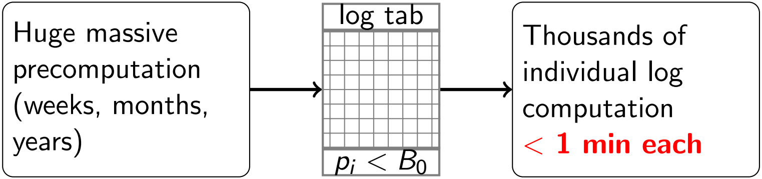
Solving actual practical problem:
Given a **fixed** finite field GF(q),



- Logjam: GF(q) = GF(p) (standardized) prime field of 512 bits
  real-time man-in-the-middle attack on Diffie-Hellman key exchange
  compute a discrete log in GF(p) in 70s in average

Logjam attack (weakdh.org)

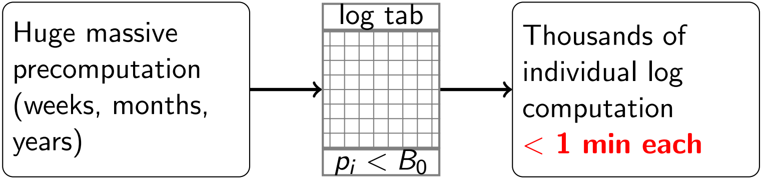
Solving actual practical problem:
Given a **fixed** finite field GF(q),



- Logjam: GF(q) = GF(p) (standardized) prime field of 512 bits real-time man-in-the-middle attack on Diffie-Hellman key exchange compute a discrete log in GF(p) in 70s in average
- Pairing-based cryptography: GF(q) = GF(p^2), GF(p^6), GF(p^{12})

Logjam attack (weakdh.org)

Solving actual practical problem:
Given a **fixed** finite field GF(q),



- Logjam: GF(q) = GF(p) (standardized) prime field of 512 bits real-time man-in-the-middle attack on Diffie-Hellman key exchange compute a discrete log in GF(p) in 70s in average
- Pairing-based cryptography: GF(q) = GF(p^2), GF(p^6), GF(p^{12})

Could we compute individual discrete logs in GF(p^2), GF(p^6), GF(p^{12}) in **less than 1 min**?

DLP in the target group of pairing-friendly curves

Why DLP in finite fields $\mathbb{F}_{p^2}, \mathbb{F}_{p^3}, \ldots$?

In a subgroup $\mathbb{G} = \langle g \rangle$ of order ℓ,

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

$$\textbf{pairing:} \quad \begin{array}{ccccc} \mathbb{G}_1 & \times & \mathbb{G}_2 & \to & \mathbb{G}_T \\ \cap & & \cap & & \cap \\ E(\mathbb{F}_p) & & E(\mathbb{F}_{p^k}) & & \mathbb{F}_{p^k}^* \end{array}$$

- where $E/\mathbb{F}_p$ is a *pairing-friendly* curve
- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of large prime order ℓ (generic attacks in $O(\sqrt{\ell})$: take e.g. 256-bit ℓ)
- $1 \leq k \leq 12$ embedding degree: very specific property (specific attacks (NFS): take 3072-bit p^k)

DL records in small characteristic

✗ Small characteristic:

- supersingular curves $E/\mathbb{F}_{2^n}$: $\mathbb{G}_T \subset \mathbb{F}_{2^{4n}}$, $E/\mathbb{F}_{3^m}$: $\mathbb{G}_T \subset \mathbb{F}_{3^{6m}}$

Practical attacks (first one and most recent):

- Hayashi, Shimoyama, Shinohara, Takagi: GF($3^{6\cdot 97}$) (923 bit field) (2012)
- Granger, Kleinjung, Zumbragel: GF(2^{9234}), GF(2^{4404}) (2014)
- Adj, Menezes, Oliveira, Rodríguez-Henríquez: GF(3^{822}), GF(3^{978}) (2014)
- Joux: GF(3^{2395}) (with Pierrot, 2014), GF(2^{6168}) (2013)

Theoretical attacks: Quasi-Polynomial-time Algorithm (QPA)

- [Barbulescu Gaudry Joux Thomé 14]
- [Granger Kleinjung Zumbragel 14]

Common used pairing-friendly curves

✓ Curves over prime fields $E/\mathbb{F}_p$ where QPA does NOT apply (with $\log p \geq \log \ell \approx 256$ bits, s.t. $k \log p \geq 3072$)

- supersingular: $\mathbb{G}_T \subset \mathbb{F}_{p^2}$ ($\log p = 1536$)
- [Miyaji Nakabayashi Takano 01] (MNT): $\mathbb{G}_T \subset \mathbb{F}_{p^3}$ ($\log p = 1024$), $\mathbb{F}_{p^4}$ ($\log p = 768$), $\mathbb{F}_{p^6}$ ($\log p = 512$)
- [Freeman 06] $\mathbb{G}_T \subset \mathbb{F}_{p^{10}}$
- [Barreto Naehrig 05] (BN): $\mathbb{G}_T \subset \mathbb{F}_{p^{12}}$ ($\log p = 256$, optimal)
- [Kachisa Schaefer Scott 08] (KSS): $\mathbb{G}_T \subset \mathbb{F}_{p^{18}}$ (used for 192-bit security level: 384-bit ℓ, $\log p = 512$, $k \log p = 9216$)

Last DL records, with the NFS-DL algorithm

GF(p)	GF(p'^2), $p'^2 = q$ [BGGM15]	
Massive precomputation (d=core-day, y=core-year)		
[Logjam] 512-bit p: 10y		
[BGIJT14] 596-bit p: 131y	598-bit q: 0.75y + 18 GPU-d	**175× faster**
Individual Discrete Log		
512-bit p: 70s median ✓		
596-bit p: 2d	600-bit q : few d	**slow**

[Logjam]: see `weakdh.org`

[BGGM15]: Barbulescu, Gaudry, G., Morain

[BGIJT14]: Bouvier, Gaudry, Imbert, Jeljeli, Thomé

This work:

- Faster **individual** discrete logarithm in $\mathbb{F}_{p^k}$, especially $k = 2, 3, 4, 6$
- Apply to pairing target group $\mathbb{G}_T$
  - large characteristic $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^3}$
  - medium characteristic $\mathbb{F}_{p^4}$, $\mathbb{F}_{p^6}$, ...
- source code: written in Magma
  + part of `http://cado-nfs.gforge.inria.fr/`

Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

1. *Polynomial selection:*
   compute $f(x)$, $g(x)$ with
   $\varphi = \gcd(f,g) \pmod{p}$ and
   $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$

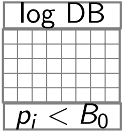
Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

1. *Polynomial selection:* compute $f(x),\ g(x)$ with $\varphi = \gcd(f, g) \pmod p$ and $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$
2. *Relation collection*

Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

1. *Polynomial selection:*
   compute $f(x),\ g(x)$ with
   $\varphi = \gcd(f, g) \pmod p$ and
   $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$
2. *Relation collection*
3. *Linear algebra modulo $\ell \mid p^k - 1$.*

➙ here we know the discrete log of a subset of elements.

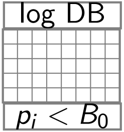


Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

1. *Polynomial selection:* compute $f(x),\ g(x)$ with $\varphi = \gcd(f, g) \pmod p$ and $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$
2. **Relation collection**
3. **Linear algebra modulo $\ell \mid p^k - 1$**

massive precomputation

➙ here we know the discrete log of a subset of elements.

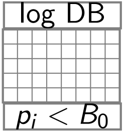


Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

1. *Polynomial selection:* compute $f(x),\ g(x)$ with $\varphi = \gcd(f, g) \pmod{p}$ and $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$
2. **Relation collection**
3. **Linear algebra modulo $\ell \mid p^k - 1$**

massive precomputation

➙ here we know the discrete log of a subset of elements.



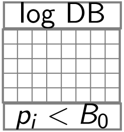
1. *Individual target discrete logarithm*

Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

1. *Polynomial selection:* compute $f(x),\ g(x)$ with $\varphi = \gcd(f, g) \pmod{p}$ and $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$
2. **Relation collection**
3. **Linear algebra modulo $\ell \mid p^k - 1$**

massive precomputation

➙ here we know the discrete log of a subset of elements.



1. **Individual target discrete logarithm for each given DLP instance**

- not so trivial
- this talk: practical improvements very efficient for small k or even k

Polynomial Selection for DL in $\mathbb{F}_{p^k}$, and norm

- f, g irreducible over $\mathbb{Q}$, $f \neq g$ (define $\neq$ number fields)
- $\gcd(f \bmod p, g \bmod p) = \varphi$ irreducible of degree k
- $\|f\|_\infty$, $\|g\|_\infty$, $\deg f$, $\deg g$ small enough s.t. $\mathrm{Norm}_f(\cdot)$, $\mathrm{Norm}_g(\cdot)$ are as small as possible

Norm of degree 1 element $a - bx \in \mathbb{Z}[x]/(f(x))$:

- $Norm_f(a - bx) = \sum_{i=0}^{\deg f} a^i b^{\deg f - i} f_i$

More generally, when f is monic:

- $Norm_f(T) = \mathrm{Res}(T, f) \leq A(\deg f, \deg T) \|\mathbf{T}\|_\infty^{\deg f} \|f\|_\infty^d$

where $\|f\|_\infty = \max_{0 \leq i \leq \deg f} |f_i|$

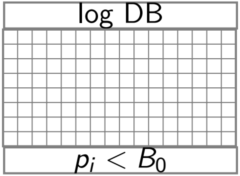
Polynomial Selection for $\mathbb{F}_{p^4}$

Both polynomials have large coefficients. $\mathbb{F}_{p^4}$ record of 392 bits (120 dd):

- $p = 314159265358979323846270891033$ of 98 bits (30 decimal digits dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- let $\mathbf{y} = 560499121640472$ and compute $u/v \equiv \mathbf{y} \pmod{p}$
- $g = v \cdot f_{\mathbf{y} \leftarrow u/v}(x)$
  $g = 560499121639105x^4 + 4898685125033473x^3 - 3362994729834630x^2 - 4898685125033473x + 560499121639105$
- $Norm_{\mathbb{Q}[x]/(f(x))}(a - bx) =$
  $a^4 - 560499121640472a^3b - 6a^2b^2 + 560499121640472ab^3 + b^4$
  $\approx \max(|a|, |b|)^4 \|f\|_\infty$

Relation collection and Linear algebra

2. Relation collection (cado-nfs: Pierrick Gaudry and Laurent Grémy)
3. Linear algebra (cado-nfs: Emmanuel Thomé and Cyril Bouvier)



- We know the log of *small* elements in $\mathbb{Z}[x]/(f(x))$ and $\mathbb{Z}[x]/(g(x))$
- *small* elements are of the form $a_i - b_i x = \in \mathbb{Z}[x]/(f(x))$, s.t. $|\mathrm{Norm}(a_i - b_i x)| = q_i \leq B_0$

Individual Discrete Logarithm

Preimage in $\mathbb{Z}[x]/(f(x))$ and ρ map

$$\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
\mathbb{Z}[x]/(f(x)) \arrow[dr, "\rho_f \,:\, x \mapsto z"'] & & \mathbb{Z}[x']/(g(x')) \arrow[dl, "\rho_g \,:\, x' \mapsto z"] \\
 & \mathbb{F}_{p^k} = \mathbb{F}_p[z]/(\varphi(z)) &
\end{tikzcd}$$

Randomized target $T = t_0 + t_1 X + t_2 X^2 + t_3 X^3 \in \mathbb{F}_{p^4}^* = \mathbb{F}_p[X]/(\varphi(X))$
Simplest choice of preimage $\mathbf{T}$: since $f = \varphi$,
$\mathbf{T} = \mathbf{t_0} + \mathbf{t_1} x + \mathbf{t_2} x^2 + \mathbf{t_3} x^3 \in \mathbb{Z}[x]/(f(x))$, with $\mathbf{t_i} \equiv t_i \pmod{p}$.
We can always choose $\mathbf{T}$ s.t.

- $|\mathbf{t_i}| < p$
- $\deg \mathbf{T} < \deg \varphi$

$$\text{We need } \rho(\mathbf{T}) = T$$

(where ρ is simply a reduction modulo (φ, p) when f (resp. g) is monic)

Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

Given G and a log database s.t. for all $p_i < B_0$, $\log p_i \in$ log DB $p_i < B_0$

Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

Given G and a log database s.t. for all $p_i < B_0$, $\log p_i \in$ log DB $p_i < B_0$

1. boot step (a.k.a. smoothing step):
   DO
   1.1 take t at random in $\{1, \ldots, \ell - 1\}$ and set $T = G^t T_0$
   (hence $\log_G(T_0) = \log_G(T) - t$)
   1.2 factorize $\mathrm{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } B_0 < q_i \leq B_1} \times$(elements in DL database),

   UNTIL $q_i \leq B_1$

Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

Given G and a log database s.t. for all $p_i < B_0$, $\log p_i \in$ log DB $p_i < B_0$

1. boot step (a.k.a. smoothing step):
   DO
   1.1 take t at random in $\{1, \ldots, \ell - 1\}$ and set $T = G^t T_0$
   (hence $\log_G(T_0) = \log_G(T) - t$)
   1.2 factorize $\mathrm{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } B_0 < q_i \leq B_1} \times$(elements in DL database),

   UNTIL $q_i \leq B_1$
2. Descent strategy: set $\mathcal{S} = \{q_i : \ B_0 < q_i \leq B_1\}$
   while $\mathcal{S} \neq \emptyset$ **do**
   - set $B_j < B_i$
   - find a relation $q_i = \prod_{B_0 < q_j < B_j} q_j \times$ (elements in log DB)
   - $\mathcal{S} \leftarrow \mathcal{S} \setminus \{q_i\} \cup \{q_j\}_{j \in J}$

   end while

Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

Given G and a log database s.t. for all $p_i < B_0$, $\log p_i \in$ log DB $p_i < B_0$

1. boot step (a.k.a. smoothing step):
   DO
   1.1 take t at random in $\{1, \ldots, \ell - 1\}$ and set $T = G^t T_0$
   (hence $\log_G(T_0) = \log_G(T) - t$)
   1.2 factorize $\mathrm{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } B_0 < q_i \leq B_1} \times$(elements in DL database),

   UNTIL $q_i \leq B_1$
2. Descent strategy: set $\mathcal{S} = \{q_i : \ B_0 < q_i \leq B_1\}$
   while $\mathcal{S} \neq \emptyset$ **do**
   - set $B_j < B_i$
   - find a relation $q_i = \prod_{B_0 < q_j < B_j} q_j \times$ (elements in log DB)
   - $\mathcal{S} \leftarrow \mathcal{S} \setminus \{q_i\} \cup \{q_j\}_{j \in J}$

   end while
3. log combination to find the individual target DL

Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

Given G and a log database s.t. for all $p_i < B_0$, $\log p_i \in$ $\boxed{\text{log DB} \; | \; p_i < B_0}$

1. boot step (a.k.a. smoothing step):
   DO
   1.1 take t at random in $\{1, \ldots, \ell - 1\}$ and set $T = G^t T_0$
   (hence $\log_G(T_0) = \log_G(T) - t$)
   1.2 factorize $\underbrace{\text{Norm}(\mathbf{T})}_{\text{reduce this}} = \underbrace{q_1 \cdots q_i}_{\text{too large: } B_0 < q_i \leq B_1} \times$(elements in DL database),

   UNTIL $q_i \leq B_1$
2. Descent strategy: set $\mathcal{S} = \{q_i : \; B_0 < q_i \leq B_1\}$
   while $\mathcal{S} \neq \emptyset$ **do**
   - set $B_j < B_i$
   - find a relation $q_i = \prod_{B_0 < q_j < B_j} q_j \times$ (elements in log DB)
   - $\mathcal{S} \leftarrow \mathcal{S} \setminus \{q_i\} \cup \{q_j\}_{j \in J}$

   end while
3. log combination to find the individual target DL

Boot step complexity

Given random target $T_0 \in \mathbb{F}_{p^k}^*$, and G a generator of $\mathbb{F}_{p^k}^*$
repeat

1. take t at random in $\{1, \ldots, \ell - 1\}$ and set $T = G^t T_0$
2. factorize $\mathrm{Norm}(\mathbf{T})$

until it is B_1-smooth: $\mathrm{Norm}(\mathbf{T}) = \prod_{q_i \leq B_1} q_i \times$(elts in log DB)

L-notation: $Q = p^k$, $L_Q[1/3, \mathbf{c}] = e^{(\mathbf{c}+o(1))(\log Q)^{1/3} \ (\log \log Q)^{2/3}}$ for $\mathbf{c} > 0$.
Norm factorization done with ECM method, in time $L_{B_1}[1/2, \sqrt{2}]$

Lemma (Boot step running-time)

If $\mathrm{Norm}(\mathbf{T}) \leq Q^e$, *take* $B_1 = L_Q[2/3, (e^2/3)^{1/3}]$, *then the running-time is* $L_Q[1/3, (3e)^{1/3}]$ *(and this is optimal).*

Preimage optimization

f, $\deg f$, $\|f\|_\infty$, g, $\deg g$, $\|g\|_\infty$ are given by the polynomial selection step (NFS-DL step 1)

$$\mathrm{Norm}_f(\mathbf{T}) = Res(f, \mathbf{T}) \leq A \|\mathbf{T}\|_\infty^{\deg f} \|f\|_\infty^d$$

To reduce the norm,

- reduce $\|\mathbf{T}\|_\infty$
- and/or reduce $d = \deg \mathbf{T}$

Boot step: First experiments

Commonly assumed to be very easy and very fast. This is not always so easy!

- $\mathbb{F}_{p_{90}^2}$ 600 bits (BGGM15 record) was easy, as fast as for $\mathbb{F}_{p_{180}}$ ($<$ one day) with [JLSV06] improvement technique
- $\mathbb{F}_{p^3}$ MNT 508 bits was much slower (days, week)
- $\mathbb{F}_{p^4}$ 392 bits was even worse ($>$ one week)

What happened?

- $\mathbb{F}_{p^3}$: asymptotically the same as $\mathbb{F}_{p^2}$: $L_Q[1/3, c = 1.44]$ but still much slower, **Because of the constant hidden in the $O()$?**
- $\mathbb{F}_{p^4}$: [JLSV06] not suited, $\|f\|_\infty = O(p^{1/2})$, $\mathrm{Norm}(\mathbf{T}) \approx Q^{3/2} \to L_Q[1/3, c = 1.65]$

Our solution

Lemma

Let $T \in \mathbb{F}_{p^k}$.
Then $\log(T) = \log(u \cdot T) \pmod{\ell}$ *for any* u *in a proper subfield of* $\mathbb{F}_{p^k}$.

Our solution

Lemma

Let $T \in \mathbb{F}_{p^k}$.
Then $\log(T) = \log(u \cdot T) \pmod{\ell}$ *for any* u *in a proper subfield of* $\mathbb{F}_{p^k}$.

- $\mathbb{F}_p$ is a proper subfield of $\mathbb{F}_{p^k}$
- target $T = t_0 + t_1 x + \ldots + t_d x^d$
- we divide the target by its leading term:

$$\log(T) = \log(T/t_d) \pmod{\ell}$$

From now on we assume that the target is monic.

Our solution

Lemma

Let $T \in \mathbb{F}_{p^k}$*.*
Then $\log(T) = \log(u \cdot T) \pmod{\ell}$ *for any* u *in a proper subfield of* $\mathbb{F}_{p^k}$*.*

- $\mathbb{F}_p$ is a proper subfield of $\mathbb{F}_{p^k}$
- target $T = t_0 + t_1 x + \ldots + t_d x^d$
- we divide the target by its leading term:

$$\log(T) = \log(T/t_d) \pmod{\ell}$$

From now on we assume that the target is monic.
Similar technique in pairing computation:
Miller loop denominator elimination [Boneh Kim Lynn Scott 02]

$\mathbb{F}_{p^4}$ of 392 bits: Terribly slow booting step

- $p = 314159265358979323846270891033$ of 98 bits (30 dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- we want to reduce $\|\mathbf{T}\|_\infty$. Define $L = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix}$
- dim 4 because $\max(\deg f, \deg g) = 4$
- LLL(L) outputs a short vector r, linear combination of L's rows. $r = \lambda_0 p + \lambda_1 p x + \lambda_2 p x^2 + \lambda_3 T$,

$\mathbb{F}_{p^4}$ of 392 bits: Terribly slow booting step

- $p = 314159265358979323846270891033$ of 98 bits (30 dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- we want to reduce $\|\mathbf{T}\|_\infty$. Define $L =$
$$\begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \begin{array}{l} p \mapsto 0 \text{ in } \mathbb{F}_{p^4} \\ px \mapsto 0 \\ px^2 \mapsto 0 \\ \mathbf{T} \mapsto T \end{array}$$
- dim 4 because $\max(\deg f, \deg g) = 4$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 px^2 + \lambda_3 T$,

$\mathbb{F}_{p^4}$ of 392 bits: Terribly slow booting step

- $p = 314159265358979323846270891033$ of 98 bits (30 dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- we want to reduce $\|\mathbf{T}\|_\infty$. Define $L =$
  $$\begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \begin{array}{l} p \mapsto 0 \text{ in } \mathbb{F}_{p^4} \\ px \mapsto 0 \\ px^2 \mapsto 0 \\ \mathbf{T} \mapsto T \end{array}$$
- dim 4 because $\max(\deg f, \deg g) = 4$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 px^2 + \lambda_3 T,$

$\mathbb{F}_{p^4}$ of 392 bits: Terribly slow booting step

- $p = 314159265358979323846270891033$ of 98 bits (30 dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- we want to reduce $\|\mathbf{T}\|_\infty$. Define $L =$
$$\begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \quad \begin{array}{l} p \mapsto 0 \text{ in } \mathbb{F}_{p^4} \\ px \mapsto 0 \\ px^2 \mapsto 0 \\ \mathbf{T} \mapsto T \end{array}$$
- dim 4 because $\max(\deg f, \deg g) = 4$
- LLL(L) outputs a short vector r, linear combination of L's rows.
$r = \lambda_0 p + \lambda_1 px + \lambda_2 px^2 + \lambda_3 T$, $\mathbf{\log \rho(r) = \log(T) \pmod{\ell}}$

$\mathbb{F}_{p^4}$ of 392 bits: Terribly slow booting step

- $p = 314159265358979323846270891033$ of 98 bits (30 dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- we want to reduce $\|\mathbf{T}\|_\infty$. Define $L =$
  $$\begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \begin{array}{l} p \mapsto 0 \text{ in } \mathbb{F}_{p^4} \\ px \mapsto 0 \\ px^2 \mapsto 0 \\ \mathbf{T} \mapsto T \end{array}$$
- dim 4 because $\max(\deg f, \deg g) = 4$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 px^2 + \lambda_3 T$, $\mathbf{\log \rho(r) = \log(T) \pmod{\ell}}$
- $r = r_0 + \ldots + r_3 x^3$, $\|r_i\|_\infty \leq C \det(L)^{1/4} = O(p^{3/4})$
- $\mathrm{Norm}_f(r) \approx \|r\|_\infty^4 \|f\|_\infty^3 \approx p^{9/2} = Q^{9/8}$ of 450 bits instead of 588 b
- Booting step, number of operations: 2^{44}
- Large prime bound B_1 of 81 bits

$\mathbb{F}_{p^4}$ of 392 bits: Terribly slow booting step

- $p = 314159265358979323846270891033$ of 98 bits (30 dd)
- $f = x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1$
- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- we want to reduce $\|\mathbf{T}\|_\infty$. Define $L =$

$$\begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \begin{array}{l} p \mapsto 0 \text{ in } \mathbb{F}_{p^4} \\ px \mapsto 0 \\ px^2 \mapsto 0 \quad \leftarrow \text{ could we find something else, } \textit{monic}? \\ \mathbf{T} \mapsto T \end{array}$$

- dim 4 because $\max(\deg f, \deg g) = 4$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 px^2 + \lambda_3 T$, $\mathbf{\log \rho(r) = \log(T) \pmod{\ell}}$
- $r = r_0 + \ldots + r_3 x^3$, $\|r_i\|_\infty \leq C \det(L)^{1/4} = O(p^{3/4})$
- $\mathrm{Norm}_f(r) \approx \|r\|_\infty^4 \|f\|_\infty^3 \approx p^{9/2} = Q^{9/8}$ of 450 bits instead of 588 b
- Booting step, number of operations: 2^{44}
- Large prime bound B_1 of 81 bits

Our solution: quadratic subfield cofactor simplification

Lemma

Let $T \in \mathbb{F}_{p^k}$, k even. We can always find $u \in \mathbb{F}_{p^2}$ and $T' \in \mathbb{F}_{p^k}$ such that $T' = u \cdot T$ and T' is represented by a polynomial of degree $k-2$ instead of $k-1$.

Our solution: quadratic subfield cofactor simplification

Lemma

Let $T \in \mathbb{F}_{p^k}$, k even. We can always find $u \in \mathbb{F}_{p^2}$ and $T' \in \mathbb{F}_{p^k}$ such that $T' = u \cdot T$ and T' is represented by a polynomial of degree $k-2$ instead of $k-1$.

- define $L = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ t'_0 & t'_1 & 1 & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix}$
- LLL$(L) \rightarrow$ short vector r linear combination of L's rows
  $r = r_0 + \ldots + r_3 x^3,\ \|r_i\|_\infty \leq C \det(L)^{1/4} = O(p^{1/2})$

Our solution: quadratic subfield cofactor simplification

Lemma

Let $T \in \mathbb{F}_{p^k}$, k even. We can always find $u \in \mathbb{F}_{p^2}$ and $T' \in \mathbb{F}_{p^k}$ such that $T' = u \cdot T$ and T' is represented by a polynomial of degree $k-2$ instead of $k-1$.

- define $L = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ t'_0 & t'_1 & 1 & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \begin{array}{l} \rho(p) = 0 \in \mathbb{F}_{p^k} \\ \rho(px) = 0 \in \mathbb{F}_{p^k} \\ T' \\ T \end{array}$
- LLL(L) $\rightarrow$ short vector r linear combination of L's rows
  $r = r_0 + \ldots + r_3 x^3,\ \|r_i\|_\infty \leq C \det(L)^{1/4} = O(p^{1/2})$
- $\rho(r) = \lambda_2 T' + \lambda_3 T = \underbrace{(\lambda_2 u + \lambda_3)}_{\in \text{ subfield } \mathbb{F}_{p^{k/2}}} T$

Our solution: quadratic subfield cofactor simplification

Lemma

Let $T \in \mathbb{F}_{p^k}$, k even. We can always find $u \in \mathbb{F}_{p^2}$ and $T' \in \mathbb{F}_{p^k}$ such that $T' = u \cdot T$ and T' is represented by a polynomial of degree $k-2$ instead of $k-1$.

- define $L = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ t'_0 & t'_1 & 1 & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \begin{array}{l} \rho(p) = 0 \in \mathbb{F}_{p^k} \\ \rho(px) = 0 \in \mathbb{F}_{p^k} \\ T' \\ T \end{array}$
- LLL(L) $\rightarrow$ short vector r linear combination of L's rows
  $r = r_0 + \ldots + r_3 x^3, \ \|r_i\|_\infty \leq C \det(L)^{1/4} = O(p^{1/2})$
- $\rho(r) = \lambda_2 T' + \lambda_3 T = \underbrace{(\lambda_2 u + \lambda_3)}_{\in \text{ subfield } \mathbb{F}_{p^{k/2}}} T$
- $\log \rho(r) = \log(T) \pmod{\ell}$
- $\mathrm{Norm}_f(r) = \|r\|_\infty^4 \|f\|_\infty^3 = p^{7/2} = Q^{7/8} < Q$

Subfield Cofactor Simplification + LLL results

		$\mathrm{Norm}_f(\mathbf{T})$		$L_Q[1/3, c]$		$q_i \leq B_1 =$
		Q^e	bits	c	time	$L_Q[\frac{2}{3}, c]$
$\mathbb{F}_{p^2}$ 600 bits	$T = U/V$	$Q^{1/2}Q^{1/2}$	600	1.44	2^{52}	2^{100}
	This work	$\mathbf{Q^{1/2}}$	**300**	**1.14**	$\mathbf{2^{41}}$	$\mathbf{2^{64}}$

$\mathbb{F}_{p^3}$ 508 bits	$T = U/V$	$Q^{1/2}Q^{1/2}$	508	1.44	2^{48}	2^{90}
	This work	$\mathbf{Q^{2/3}}$	**340**	**1.26**	$\mathbf{2^{42}}$	$\mathbf{2^{69}}$

$\mathbb{F}_{p^4}$ 392 bits	prev.	$Q^{3/2}$	588	1.65	2^{49}	2^{98}
	This work	$\mathbf{Q^{7/8}}$	**343**	**1.38**	$\mathbf{2^{41}}$	$\mathbf{2^{68}}$

Subfield Cofactor Simplification + LLL results

		$\mathrm{Norm}_f(\mathbf{T})$		$L_Q[1/3, c]$		$q_i \leq B_1 =$
		Q^e	bits	c	time	$L_Q[\frac{2}{3}, c]$
$\mathbb{F}_{p^2}$ 600 bits	$T = U/V$	$Q^{1/2}Q^{1/2}$	600	1.44	2^{52}	2^{100}
	This work	$\mathbf{Q^{1/2}}$	**300**	**1.14**	$\mathbf{2^{41}}$	$\mathbf{2^{64}}$

$\mathbb{F}_{p^3}$ 508 bits	$T = U/V$	$Q^{1/2}Q^{1/2}$	508	1.44	2^{48}	2^{90}
	This work	$\mathbf{Q^{2/3}}$	**340**	**1.26**	$\mathbf{2^{42}}$	$\mathbf{2^{69}}$

$\mathbb{F}_{p^4}$ 392 bits	prev.	$Q^{3/2}$	588	1.65	2^{49}	2^{98}
	This work	$\mathbf{Q^{7/8}}$	**343**	**1.38**	$\mathbf{2^{41}}$	$\mathbf{2^{68}}$

Faster descent

DL record computation in $\mathbb{F}_{p^4}$ of 392 bits (120dd)

Joint work with R. Barbulescu, P. Gaudry, F. Morain

$$
\begin{aligned}
p &= 314159265358979323846270891033 \text{ of 98 bits (30 dd)} \\
\ell &= 986960440108935861883490271847705742814406423277878775980709 \text{ of 192 bits} \\
f &= x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1 \\
g &= 560499121639105x^4 + 4898685125033473x^3 - 3362994729834630x^2 \\
&\quad -4898685125033473x + 560499121639105 \\
\varphi &= g \\
G &= x + 3 \in \mathbb{F}_{p^4} \\
T_0 &= 31415926535897x^3 + 93238462643383x^2 + 27950288419716x + 93993751058209
\end{aligned}
$$

$$\mathbf{\log_G(T_0) =}$$

$$\mathbf{1364394725868398385294409072195832018219505919841942570 22 \pmod{\ell}}$$

Summary of results

- better practical and asymptotic running-time of the boot step
- better when k is even

- online version HAL 01157378
- `guillevic@lix.polytechnique.fr`

Future work

- Degree-d subfield cofactor simplification thanks to an anonymous Asiacrypt 2015 reviewer remark, generalization in large characteristic, application to small characteristic
- look at Sarkar Singh (eprint 2015/944) polynomial selection
- optimize the descent
- add early abort strategy (Barbulescu improvement)
- $\mathbb{F}_{p^6}$, $\mathbb{F}_{p^{12}}$

Future work

- Degree-d subfield cofactor simplification thanks to an anonymous Asiacrypt 2015 reviewer remark, generalization in large characteristic, application to small characteristic
- look at Sarkar Singh (eprint 2015/944) polynomial selection
- optimize the descent
- add early abort strategy (Barbulescu improvement)
- $\mathbb{F}_{p^6}$, $\mathbb{F}_{p^{12}}$

Be careful with the hidden constant in the $O(\cdot)$