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

Aurore Guillevic

INRIA Saclay / GRACE Team

École Polytechnique / LIX

ECC 2015, Sept. 28th

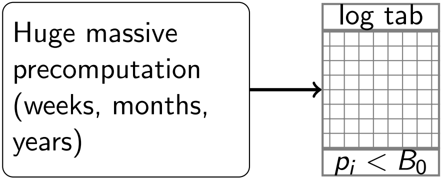
Link with Logjam attack (N. Heninger's talk)

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

Huge massive precomputation (weeks, months, years)

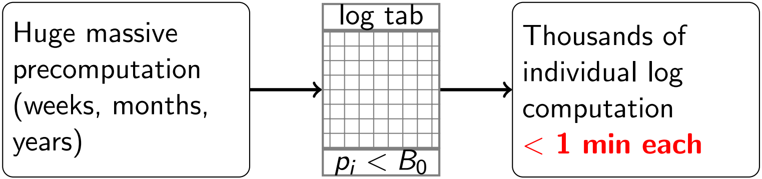
Link with Logjam attack (N. Heninger's talk)

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



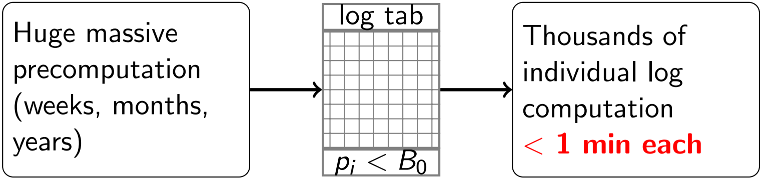
Link with Logjam attack (N. Heninger's talk)

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



Link with Logjam attack (N. Heninger's talk)

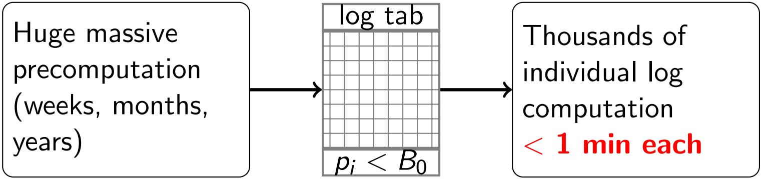
Solving actual practical problem:
Given a **fixed** finite field GF(q),



- Logjam: GF(q) = GF(p) (standardized) prime field

Link with Logjam attack (N. Heninger's talk)

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



- Logjam: GF(q) = GF(p) (standardized) prime field
- Pairing-based cryptosystems: GF(q) = GF(p^2), GF(p^6), GF(p^{12})

Link with Logjam attack (N. Heninger's talk)

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

Huge massive precomputation (weeks, months, years) → log tab ($p_i < B_0$) → Thousands of individual log computation **< 1 min each**

- Logjam: GF(q) = GF(p) (standardized) prime field
- Pairing-based cryptosystems: 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.

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

- 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:

- [Barbulescu Gaudry Joux Thomé 14] Quasi-Polynomial-time Algorithm (QPA)
- ...

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): $\mathbb{G}_T \subset \mathbb{F}_{p^3}$ ($\log p = 1024$), $\mathbb{F}_{p^4}$ ($\log p = 768$), $\mathbb{F}_{p^6}$ ($\log p = 512$)
- [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, $\log p = 512$, $k \log p = 9216$)

Theoretical attacks in non-small characteristic fields

Variants of NFS, generic fields

- MNFS [Coppersmith 89]: $\mathbb{F}_p$, [Barbulescu Pierrot 14], [Pierrot 15]: $\mathbb{F}_{p^k}$

Specific to pairing target groups, when $p = P(x_0)$, with $\deg P \geq 2$

- [Joux Pierrot 13]
- [Barbulescu Gaudry Kleinjung 15] Tower NFS

Theoretical attacks in non-small characteristic fields

Variants of NFS, generic fields

- MNFS [Coppersmith 89]: $\mathbb{F}_p$, [Barbulescu Pierrot 14], [Pierrot 15]: $\mathbb{F}_{p^k}$

Specific to pairing target groups, when $p = P(x_0)$, with $\deg P \geq 2$

- [Joux Pierrot 13]
- [Barbulescu Gaudry Kleinjung 15] Tower NFS

These attacks were not taken into account in the 3072-bit target field recommendation.

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	530-bit q: 0.2y + 1.25 GPU d	
[BGIJT14] 596-bit p: 131y	598-bit q: 0.75y + 18 GPU-d	**175× faster**
Individual Discrete Log		
512-bit p: 70s median ✓	530-bit q : few d	**slow**
768-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 talk:

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

NFS – Number Field Sieve algorithm

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
   $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/(\varphi(x))$

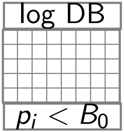
Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

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.

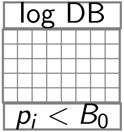


Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

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.

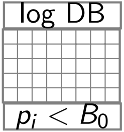


Number Field Sieve algorithm for DL in $\mathbb{F}_{p^k}$

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.



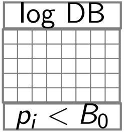
1. *Individual target discrete logarithm*

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: pratical improvements very efficient for small k

Example: [MNT01] parameters (explicitly advised to NOT use them)

Polynomial selection: Conjugation method [BGGM15]

- $k = 3$, $p = 12y_0^2 + 1$, $t = -6y_0 - 1$, $\ell \mid p + 1 - t = 12y_0^2 + 6y_0 + 2$, with $y_0 = -870230335309004989 8316902$
- $f = 12x^6 - 24x^5 - 85x^4 + 70x^3 + 215x^2 + 96x + 12$
- $\varphi_y = g = x^3 - yx^2 - (y+3)x - 1$, where $y = y_0 + 1$ (φ_{y_0} not irr.) $= x^3 + 8702303353090049898316901x^2 + 8702303353090049898316898x - 1$
- $f \pmod{p} = 12\varphi_y\varphi_{-y} = \mathrm{Res}_y(\varphi_y, 12y^2 + 1)$

$$G = X + 6 \in \mathbb{F}_{p^3}^* = \mathbb{F}_p[X]/(\varphi(X))$$
$$\text{randomized target } T = t_0 + t_1X + t_2X^2 \in \mathbb{F}_{p^3}^*$$

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

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

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

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

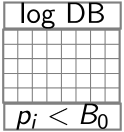
randomized target $T = t_0 + t_1 X + t_2 X^2 \in \mathbb{F}_{p^3}^* = \mathbb{F}_p[X]/(\varphi(X))$
Most simple preimage $\mathbf{T}$ choice:
$\mathbf{T} = \mathbf{t_0} + \mathbf{t_1} x + \mathbf{t_2} x^2 \in \mathbb{Z}[x]/(f(x))$, with $\mathbf{t_i} \equiv t_i \pmod p$.
We can always choose $\mathbf{T}$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

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

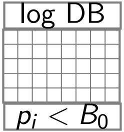
We need $\rho(\mathbf{T}) = T$ (where ρ is simply a reduction modulo (φ, p)) when f (resp. g) is monic

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

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

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

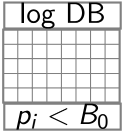
Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

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

1. booting 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 $\text{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } B_0 < q_i \leq B_1} \times$(elements in DL database),

   UNTIL $q_i \leq B_1$

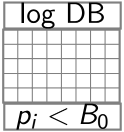
Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

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

1. booting 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. dedicated recursive procedure for each new q_i:
   $q_i = r_1 \cdots r_j \times$ (elements in the DL database) with
   $r_1, \ldots, r_j < B_j < q_i < B_i$.

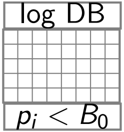
Individual DL of random target $T_0 \in \mathbb{F}_{p^k}^*$

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

1. booting 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. dedicated recursive procedure for each new q_i:
   $q_i = r_1 \cdots r_j \times$ (elements in the DL database) with
   $r_1, \ldots, r_j < B_j < q_i < B_i$.
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 $p_i < B$, $\log p_i \in$ 

1. booting 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 $\underbrace{\mathrm{Norm}(\mathbf{T})}_{\text{reduce this}} = \underbrace{q_1 \cdots q_i}_{\text{too large: } B_0 < q_i \leq B_1} \times$(elements in DL database),

   UNTIL $q_i \leq B_1$
2. dedicated recursive procedure for each new q_i:
   $q_i = r_1 \cdots r_j \times$ (elements in the DL database) with
   $r_1, \ldots, r_j < B_j < q_i < B_i$.
3. log combination to find the individual target DL

Booting Step

Norm computation

f monic,
$\mathbf{T} = t_0 + t_1 x + \ldots + t_d x^d \in \mathbb{Z}[x]/(f(x))$, $d < \deg f$:

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

with $||f||_\infty = \max_{1 \leq i \leq \deg f} |f_i|$

Example: [MNT01], $k = 3$, $\deg g = 3$, $||g||_\infty = O(p^{1/2})$

$$\begin{aligned}
p &= 9087610037904279080775489557583803566758290265 31247 \\
\mathbf{T} &= 314159265358979323846264338327950288419716939937510 + \\
&\quad 582097494459230781640628620899862803482534211706798 x + \\
&\quad 214808651328230664709384460955058223172535940812829 x^2 \\
f &= 12x^6 - 24x^5 - 85x^4 + 70x^3 + 215x^2 + 96x + 12 \\
g &= x^3 + 870230335309004989831690 1 x^2 + 8702303353090049898316898 x - 1
\end{aligned}$$

$\mathrm{Norm}_f(\mathbf{T})(\approx ||\mathbf{T}||_\infty^6 ||f||_\infty^2) =$ **1017**bits $\sim p^6$
$\mathrm{Norm}_g(\mathbf{T})(\approx ||\mathbf{T}||_\infty^3 ||g||_\infty^2) =$ **665**bits $\sim p^4$

Booting 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: $\mathrm{Norm}(\mathbf{T}) = \prod_{q_i \leq B_1} q_i \prod_{p_i \leq B_0} p_i$

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 (Booting 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).*

Booting step complexity

- $\mathbb{F}_p$: Norm(preimage) $\leq p = Q$, running-time: $L_Q[1/3, \mathbf{1.44}]$ with $B_1 = L_Q[\mathbf{2/3}, 0.69]$ [Commeine Semaev 06, Barbulescu 13]
- med. char. $\mathbb{F}_{p^k}$, JLSV1 poly. select.: $\deg f = \deg g = k$, $||f||_\infty = ||g||_\infty = O(p^{1/2})$, Norm(preimage) $\leq Q^{3/2}$, running-time: $L_Q[1/3, \mathbf{1.65}]$, with $B_1 = L_Q[\mathbf{2/3}, 0.91]$ [Joux Lercier Naccache Thomé 09, Barbulescu Pierrot 14]

field	$\mathbb{F}_p$	$\mathbb{F}_{p^k}$		
polynomial selec.		gJL	JLSV$_1$	Conj
NFS dominating, c $L_Q[\frac{1}{3}, c]$, 512-bit Q	1.92 2^{64}	1.92 2^{64}	2.42 2^{81}	2.20 2^{73}
Norm($\mathbf{T}$) $< Q^e =$ time $L_Q[1/3, c]$, c nb of operations, 512-bit Q	Q 1.44 2^{48}	Q 1.44 2^{48}	$Q^{3/2}$ 1.65 2^{55}	Q 1.44 2^{48}
q_i bound B_1	2^{90}	2^{90}	2^{118}	2^{90}

Optimizing the Preimage Computation

Preimage optimization

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

To reduce the norm,

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

Previous work

- $\mathbb{F}_p$: Rational Reconstruction. $T \in \mathbb{Z}/p\mathbb{Z}$, $\mathbf{T}$ is an integer $< p$. Rational Reconstruction gives $\mathbf{T} = u/v \pmod p$ with $u, v < \sqrt{p}$
  - booting step: we want u,v to be B_1-smooth at the same time, instead of $\mathbf{T}$ to be B_1-smooth. $\mathbf{T}$ is already split in two integers of half size each.
- [Blake Mullin Vanstone 84] Waterloo algorithm in $\mathbb{F}_2[x]$: $\mathbf{T} = U/V = \frac{u_0+\ldots+u_{\lfloor d/2 \rfloor}x^{\lfloor d/2 \rfloor}}{v_0+\ldots+v_{\lfloor d/2 \rfloor}x^{\lfloor d/2 \rfloor}}$ **reduce degree**
- [Joux Lercier Smart Vercauteren 06] in $\mathbb{F}_{p^k}$: $\mathbf{T} = U/V = \frac{u_0+\ldots+u_d x^d}{v_0+\ldots+v_d x^d}$, where $|u_i|, |v_i| \sim p^{1/2}$ **reduce coefficient size**

Previous work

- $\mathbb{F}_p$: Rational Reconstruction. $T \in \mathbb{Z}/p\mathbb{Z}$, $\mathbf{T}$ is an integer $< p$. Rational Reconstruction gives $\mathbf{T} = u/v \pmod p$ with $u, v < \sqrt{p}$
  - booting step: we want u,v to be B_1-smooth at the same time, instead of $\mathbf{T}$ to be B_1-smooth. $\mathbf{T}$ is already split in two integers of half size each.
- [Blake Mullin Vanstone 84] Waterloo algorithm in $\mathbb{F}_2[x]$: $\mathbf{T} = U/V = \frac{u_0+\ldots+u_{\lfloor d/2 \rfloor}x^{\lfloor d/2 \rfloor}}{v_0+\ldots+v_{\lfloor d/2 \rfloor}x^{\lfloor d/2 \rfloor}}$ **reduce degree**
- [Joux Lercier Smart Vercauteren 06] in $\mathbb{F}_{p^k}$: $\mathbf{T} = U/V = \frac{u_0+\ldots+u_d x^d}{v_0+\ldots+v_d x^d}$, where $|u_i|, |v_i| \sim p^{1/2}$ **reduce coefficient size**

How much is the booting step improved?

Booting step: First experiments

Commonly assumed: launch at morning coffee ... finished for afternoon tea.

- $\mathbb{F}_{p^2}$ 600 bits was easy (BGGM15 record), as fast as for $\mathbb{F}_{p'}$ ($<$ one day)
- $\mathbb{F}_{p^3}$ 400 bits and MNT 508 bits were much slower (days, week)
- $\mathbb{F}_{p^4}$ 400 bits was even worse ($>$ one week)

What happened?

- $\mathbb{F}_{p^3}$: $||\mathbf{T}||_\infty = p$, $\deg f = 6$, [JLSV06] method: $\mathrm{Norm}(\mathbf{T}) \leq Q \rightarrow c = 1.44$ (but still much slower)
- $\mathbb{F}_{p^4}$: $||f||_\infty = O(p^{1/2})$, $\mathrm{Norm}(\mathbf{T}) \leq Q^{3/2} \rightarrow c = 1.65$

Booting step: First experiments

Commonly assumed: launch at morning coffee ... finished for afternoon tea.

- $\mathbb{F}_{p^2}$ 600 bits was easy (BGGM15 record), as fast as for $\mathbb{F}_{p'}$ ($<$ one day)
- $\mathbb{F}_{p^3}$ 400 bits and MNT 508 bits were much slower (days, week)
- $\mathbb{F}_{p^4}$ 400 bits was even worse ($>$ one week)

What happened?

- $\mathbb{F}_{p^3}$: $||\mathbf{T}||_\infty = p$, $\deg f = 6$, [JLSV06] method: $\mathrm{Norm}(\mathbf{T}) \leq Q \to c = 1.44$ (but still much slower)
- $\mathbb{F}_{p^4}$: $||f||_\infty = O(p^{1/2})$, $\mathrm{Norm}(\mathbf{T}) \leq Q^{3/2} \to c = 1.65$

Because of the constant hidden in the $O()$?

Our solution

Lemma

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

Our solution

Lemma

Let $T \in \mathbb{F}_{p^k}$.
$\log(T) = \log(u \cdot T) \pmod{\ell}$ *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 we assume that the target is monic.

Our solution

Lemma

Let $T \in \mathbb{F}_{p^k}$*.*
$\log(T) = \log(u \cdot T) \pmod{\ell}$ *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 we assume that the target is monic.
Similar technique in pairing computation: Miller loop denominator elimination [Boneh Kim Lynn Scott 02]

Subfield Simplification + LLL

We want to reduce $||\mathbf{T}||_\infty$. Example with $\mathbb{F}_{p^3}$:

- $f = x^6 + 19x^5 + 90x^4 + 95x^3 + 10x^2 - 13x + 1$
- $\varphi = x^3 - yx^2 - (y+3)x - 1 \; y \in \mathbb{Z}$
- $\mathbf{T} = t_0 + t_1 x + x^2$
- define $L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 & 0 \\ 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 \\ 0 & 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 \end{bmatrix}$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 T + \lambda_3 \varphi + \lambda_4 x\varphi + \lambda_5 x^2 \varphi$.
  $r = r_0 + \ldots + r_5 x^5$, $||r_i||_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$

Subfield Simplification + LLL

We want to reduce $||\mathbf{T}||_\infty$. Example with $\mathbb{F}_{p^3}$:

- $f = x^6 + 19x^5 + 90x^4 + 95x^3 + 10x^2 - 13x + 1$
- $\varphi = x^3 - yx^2 - (y+3)x - 1 \ y \in \mathbb{Z}$
- $\mathbf{T} = t_0 + t_1 x + x^2$
- define $L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 & 0 \\ 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 \\ 0 & 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 \end{bmatrix}$
- LLL(L) outputs a short vector r, linear combination of L's rows. $r = \lambda_0 p + \lambda_1 px + \lambda_2 T + \lambda_3 \varphi + \lambda_4 x\varphi + \lambda_5 x^2\varphi$. $r = r_0 + \ldots + r_5 x^5, \ ||r_i||_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$
- $\log \rho(r) = \log(T) \pmod{\ell}$

Subfield Simplification + LLL

We want to reduce $||\mathbf{T}||_\infty$. Example with $\mathbb{F}_{p^3}$:

- $f = x^6 + 19x^5 + 90x^4 + 95x^3 + 10x^2 - 13x + 1$
- $\varphi = x^3 - yx^2 - (y+3)x - 1 \ y \in \mathbb{Z}$
- $\mathbf{T} = t_0 + t_1 x + x^2$
- define $L = \begin{array}{c} \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 & 0 \\ 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 \\ 0 & 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 \end{bmatrix} \end{array} \begin{array}{l} \rho(p) = 0 \in \mathbb{F}_{p^k} \\ \\ T \\ \rho(\varphi) = 0 \in \mathbb{F}_{p^k} \\ \\ \\ \end{array}$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 T + \lambda_3 \varphi + \lambda_4 x\varphi + \lambda_5 x^2 \varphi$.
  $r = r_0 + \ldots + r_5 x^5, \ ||r_i||_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$
- $\log \rho(r) = \log(T) \pmod{\ell}$

Subfield Simplification + LLL

We want to reduce $||\mathbf{T}||_\infty$. Example with $\mathbb{F}_{p^3}$:

- $f = x^6 + 19x^5 + 90x^4 + 95x^3 + 10x^2 - 13x + 1$
- $\varphi = x^3 - yx^2 - (y+3)x - 1 \; y \in \mathbb{Z}$
- $\mathbf{T} = t_0 + t_1 x + x^2$
- define $L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 & 0 \\ 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 & 0 \\ 0 & 0 & \varphi_0 & \varphi_1 & \varphi_2 & 1 \end{bmatrix} \begin{array}{l} \rho(p) = 0 \in \mathbb{F}_{p^k} \\ \\ T \\ \rho(\varphi) = 0 \in \mathbb{F}_{p^k} \\ \\ \\ \end{array}$
- LLL(L) outputs a short vector r, linear combination of L's rows.
  $r = \lambda_0 p + \lambda_1 px + \lambda_2 T + \lambda_3 \varphi + \lambda_4 x\varphi + \lambda_5 x^2\varphi$.
  $r = r_0 + \ldots + r_5 x^5, \; ||r_i||_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$
- $\log \rho(r) = \log(T) \pmod{\ell}$ because $\rho(r) = \lambda_2 T$ with $\lambda_2 \in \mathbb{F}_p$

Subfield Simplification + LLL

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

- $\mathrm{Norm}_f(r) \leq ||r||_\infty^6 ||f||_\infty^5 = O(p^2) = O(Q^{2/3}) < O(Q)$

MNT example: $\log Q = 508$ bits

	$\mathrm{Norm}_f(\mathbf{T})$		$\mathrm{Norm}_g(\mathbf{T})$		$L_Q[1/3, c]$		$q_i \leq B_1 =$
	Q^e	bits	Q^e	bits	c	time	$L_Q[\frac{2}{3}, c]$
Nothing	Q^2	1010	$Q^{4/3}$	667	1.58	2^{53}	2^{109}
[JLSV06]	Q	508	$Q^{5/3}$	847	1.44	2^{48}	2^{90}
This work	$\mathbf{Q^{2/3}}$	**340**	$\mathbf{Q}$	**508**	**1.26**	$\mathbf{2^{42}}$	$\mathbf{2^{69}}$

$\mathbb{F}_{p^4}$: JLSV$_1$ polynomial selection and booting step improvement

$\mathbb{F}_{p^4}$ of 400 bits

[JLSV06] first method: choose f of degree 4 and very small coefficients, and set $g = f + p$. Booting step on f side, with the $\mathbf{T} = U/V$ method.

$\mathbb{F}_{p^4}$ of 400 bits

[JLSV06] first method: choose f of degree 4 and very small coefficients, and set $g = f + p$. Booting step on f side, with the $\mathbf{T} = U/V$ method.

Relation collection and Linear algebra do not scale well for large p

$\mathbb{F}_{p^4}$ of 400 bits

[JLSV06] first method: choose f of degree 4 and very small coefficients, and set $g = f + p$. Booting step on f side, with the $\mathbf{T} = U/V$ method.

Relation collection and Linear algebra do not scale well for large p

We use JLSV06 other method: $\deg f = \deg g = k$, $||f||_\infty = ||g||_\infty = p^{1/2}$

$$\begin{aligned} p &= 314159265358979323846270891033 \text{ of 98 bits (30 dd)} \\ \ell &= 986960440108935861883490271847705742814406423277877598070 9 \text{ of 192 bits} \\ f &= x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1 \\ g &= 560499121639105x^4 + 4898685125033473x^3 - 3362994729834630x^2 \\ &\quad -4898685125033473x + 560499121639105 \\ \varphi &= g \end{aligned}$$

Terribly slow booting step (more than one week)

Terribly slow booting step

- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- 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 $\max(\deg f, \deg g) = 4$
- compute LLL(L), get r, $||r||_\infty \approx p^{3/4}$, $\mathrm{Norm}_f(r) \approx ||r||_\infty^4 ||f||_\infty^3 \approx p^{9/2} = Q^{9/8}$ of 450 bits!
- Booting step, nb of operations: 2^{44}
- Large prime bound B_1 of 82 bits

Terribly slow booting step

- $T = t_0 + t_1 x + t_2 x^2 + x^3$
- define
$$L = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ t_0 & t_1 & t_2 & 1 \end{bmatrix} \leftarrow$$ could we find something else, *monic*?
- dim 4 because $\max(\deg f, \deg g) = 4$
- compute LLL(L), get r, $||r||_\infty \approx p^{3/4}$,
$\mathrm{Norm}_f(r) \approx ||r||_\infty^4 ||f||_\infty^3 \approx p^{9/2} = Q^{9/8}$ of 450 bits!
- Booting step, nb of operations: 2^{44}
- Large prime bound B_1 of 82 bits

Our solution: quadratic subfield simplification

Lemma

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

Our solution: quadratic subfield simplification

Lemma

Let $T \in \mathbb{F}_{p^k}$, k even. We can always find $u \in \mathbb{F}_{p^{k/2}}$ and $T' \in \mathbb{F}_{p^k}$, such that $T' = u \cdot T$ and T' is 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 rows
  $r = r_0 + \ldots + r_3 x^3, \ ||r_i||_\infty \le C \det(L)^{1/4} = O(p^{1/2})$

Our solution: quadratic subfield simplification

Lemma

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

- define $L = \begin{array}{cl} \left[\begin{matrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ t'_0 & t'_1 & 1 & 0 \\ t_0 & t_1 & t_2 & 1 \end{matrix}\right] & \begin{matrix} \rho(p) = 0 \in \mathbb{F}_{p^k} \\ \\ T' \\ T \end{matrix} \end{array}$
- LLL(L) $\rightarrow$ short vector r linear combination of L 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 simplification

Lemma

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

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

$\mathrm{Norm}_f(r) = ||r||_\infty^4 ||f||_\infty^3 = p^{7/2} = Q^{7/8} < Q$

Summary of results

$\mathbb{G}_T \subset$	$\mathbb{F}_{p^2}$	$\mathbb{F}_{p^3}$	$\mathbb{F}_{p^4}$	$\mathbb{F}_{p^6}$
Norm bound				
prev.	Q [JLSV06]		$Q^{3/2}$ (nothing)	
this work	$Q^{1/2}$	$Q^{2/3}$	$Q^{7/8}$	$Q^{11/12}$
Booting step running time in $L_Q[1/3, c]$				
prev. c (*)	1.44		1.65	
new c	**1.14**	**1.26**	**1.38**	**1.40****
numerical values for a 512-bit Q				
prev. nb of operations	2^{48}		2^{55}	
new nb of operations	$\mathbf{2^{38}}$	$\mathbf{2^{42}}$	$\mathbf{2^{46}}$	$\mathbf{2^{47}}$
q_i bound $B_1 = L_Q[2/3, c']$				
previous B_1	2^{90}		2^{118}	
new $\mathbf{B_1}$	$\mathbf{2^{57}}$	$\mathbf{2^{69}}$	$\mathbf{2^{83}}$	$\mathbf{2^{85}}$

* [CommeineSemaev06, JouxLercierNaccacheThomé09, Barbulescu13, Bar.Pierrot14]

** with cubic subfield simplification

Summary of results

- Accepted paper at Asiacrypt 2015, Auckland, New Zealand
- online version HAL 01157378
- `guillevic@lix.polytechnique.fr`

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

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

$$\begin{aligned}
p &= 314159265358979323846270891033 \text{ of 98 bits (30 dd)}\\
\ell &= 986960440108935861883490271847705742814406423277877598 0709 \text{ of 192 bits}\\
f &= x^4 - 560499121640472x^3 - 6x^2 + 560499121640472x + 1\\
g &= 560499121639105x^4 + 4898685125033473x^3 - 3362994729834630x^2\\
&\quad -4898685125033473x + 560499121639105\\
\varphi &= g\\
G &= x + 3 \in \mathbb{F}_{p^4}\\
T_0 &= 31415926535897x^3 + 93238462643383x^2 + 27950288419716x + 93993751058209
\end{aligned}$$

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

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