Estimating size requirements for pairings: Simulating the Tower-NFS algorithm in GF(p^n)

Quentin Deschamps, *Aurore Guillevic*, Shashank Singh

ENS Lyon, Inria Nancy, Loria, CNRS, Université de Lorraine

November 15, 1027

Elliptic Curve Cryptography Conference
ECC17–Nijmegen, Netherlands

Cryptographic pairing: black-box properties

$(\mathbf{G}_1, +), (\mathbf{G}_2, +), (\mathbf{G}_T, \cdot)$ three cyclic groups of large prime order ℓ
Bilinear Pairing: map $e : \mathbf{G}_1 \times \mathbf{G}_2 \to \mathbf{G}_T$

1. bilinear: $e(P_1 + P_2,\ Q) = e(P_1, Q) \cdot e(P_2, Q)$,
   $e(P, Q_1 + Q_2) = e(P, Q_1) \cdot e(P, Q_2)$
2. non-degenerate: $e(g_1, g_2) \neq 1$ for $\langle g_1 \rangle = \mathbf{G}_1$, $\langle g_2 \rangle = \mathbf{G}_2$
3. efficiently computable.

Mostly used in practice:

$$e([a]P, [b]Q) = e([b]P, [a]Q) = e(P, Q)^{ab} \ .$$

$\rightsquigarrow$ Many applications in asymmetric cryptography.

Examples of application

- 1984: idea of identity-based encryption formalized by Shamir
- 1999: first practical identity-based cryptosystem of Sakai-Ohgishi-Kasahara
- 2000: constructive pairings, Joux's tri-partite key-exchange (Triffie-Hellman)
- 2001: IBE of Boneh-Franklin, short signatures Boneh-Lynn-Shacham

Rely on

- Discrete Log Problem (DLP): given $g, y \in \mathbf{G}$, compute x s.t. $g^x = y$ Diffie-Hellman Problem (DHP)
- bilinear DLP and DHP
  Given $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_T, g_1, g_2, g_T$ and $y \in \mathbf{G}_T$, compute $P \in \mathbf{G}_1$ s.t. $e(P, g_2) = y$, or $Q \in \mathbf{G}_2$ s.t. $e(g_1, Q) = y$
  if $g_T^x = y$ then $e(g_1^x, g_2) = e(g_1, g_2^x) = g_T^x = y$
- pairing inversion problem

Pairing setting: elliptic curves

$$E/\mathbb{F}_p: \ y^2 = x^3 + ax + b, \ a, b \in \mathbb{F}_p, \ p \geq 5$$

- proposed in 1985 by Koblitz, Miller
- $E(\mathbb{F}_p)$ has an efficient group law (chord an tangent rule) $\rightarrow$ **G**
- $\#E(\mathbb{F}_p) = p + 1 - tr$, trace tr: $|tr| \leq 2\sqrt{p}$
- efficient group order computation (*point counting*)
- large subgroup of prime order ℓ s.t. $\ell \mid p + 1 - tr$ and ℓ coprime to p
- $E[\ell] \simeq \mathbb{Z}/\ell\mathbb{Z} \oplus \mathbb{Z}/\ell\mathbb{Z}$ (for crypto)
- only generic attacks against DLP on well-chosen genus 1 and genus 2 curves
- optimal parameter sizes ($\log_2 \ell = \log_2 p$)

Pairings

1948 Weil pairing (accouplement)
1958 Tate pairing
1985 Miller, Koblitz: use Elliptic Curves in crypto
1986 Miller's algorithm to compute pairings
1988 Kaliski's implementation $E/\mathbb{F}_{11} : y^2 = x^3 - x$ (PhD at MIT)
At that time:

- easy to use supersingular curves for ECC: group order known

Supersingular elliptic curves

Example over $\mathbb{F}_p$, $p \geq 5$

$$E : y^2 = x^3 + x \ / \ \mathbb{F}_p, \quad p = 3 \bmod 4$$

s.t. $t = 0$, $\#E(\mathbb{F}_p) = p + 1$.
take p s.t. $p + 1 = 4 \cdot \ell$ where ℓ is prime.

1993: Menezes-Okamoto-Vanstone and Frey-Rück attacks
$\exists$ pairing $e : E(\mathbb{F}_p)$ into $\mathbb{F}_{p^2}$ where **DLP is much easier**.
Do not use supersingular curves (1993–1999)
But computing a pairing is **very slow**:
[Harasawa Shikata Suzuki Imai 99]: 161467s (112 days) on a 163-bit supersingular curve, where $\mathbf{G}_T \subset \mathbb{F}_{p^2}$ of 326 bits.

Pairing-based cryptography

1999: Frey–Muller–Rück: actually, Miller Algorithm can be **much faster**.

2000: [Joux ANTS] Computing a pairing can be done efficiently (1s on a supersingular 528-bit curve, $\mathbf{G}_T \subset \mathbb{F}_{p^2}$ of 1055 bits).

Weil or Tate pairing on an elliptic curve

Discrete logarithm problem with one more dimension.

$$e \ : \ E(\mathbb{F}_{p^n})[\ell] \times E(\mathbb{F}_{p^n})[\ell] \longrightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab}$$

Pairing-based cryptography

1999: Frey–Muller–Rück: actually, Miller Algorithm can be **much faster**.

2000: [Joux ANTS] Computing a pairing can be done efficiently (1s on a supersingular 528-bit curve, $\mathbf{G}_T \subset \mathbb{F}_{p^2}$ of 1055 bits).

Weil or Tate pairing on an elliptic curve

Discrete logarithm problem with one more dimension.

$$e \; : \; E(\mathbb{F}_{p^n})[\ell] \times E(\mathbb{F}_{p^n})[\ell] \longrightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab}$$

Attacks

Pairing-based cryptography

1999: Frey–Muller–Rück: actually, Miller Algorithm can be **much faster**.

2000: [Joux ANTS] Computing a pairing can be done efficiently (1s on a supersingular 528-bit curve, $\mathbf{G}_T \subset \mathbb{F}_{p^2}$ of 1055 bits).

Weil or Tate pairing on an elliptic curve

Discrete logarithm problem with one more dimension.

$$e \ : \ E(\mathbb{F}_{p^n})[\ell] \times E(\mathbb{F}_{p^n})[\ell] \longrightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab}$$

Attacks

- inversion of e : hard problem (exponential)

Pairing-based cryptography

1999: Frey–Muller–Rück: actually, Miller Algorithm can be **much faster**.

2000: [Joux ANTS] Computing a pairing can be done efficiently (1s on a supersingular 528-bit curve, $\mathbf{G}_T \subset \mathbb{F}_{p^2}$ of 1055 bits).

Weil or Tate pairing on an elliptic curve

Discrete logarithm problem with one more dimension.

$$e \ : \ E(\mathbb{F}_{p^n})[\ell] \times E(\mathbb{F}_{p^n})[\ell] \longrightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab}$$

Attacks

- inversion of e : hard problem (exponential)
- discrete logarithm computation in $E(\mathbb{F}_p)$: hard problem (exponential, in $O(\sqrt{\ell})$)

Pairing-based cryptography

1999: Frey–Muller–Rück: actually, Miller Algorithm can be **much faster**.

2000: [Joux ANTS] Computing a pairing can be done efficiently (1s on a supersingular 528-bit curve, $\mathbf{G}_T \subset \mathbb{F}_{p^2}$ of 1055 bits).

Weil or Tate pairing on an elliptic curve

Discrete logarithm problem with one more dimension.

$$e \ : \ E(\mathbb{F}_{p^n})[\ell] \times E(\mathbb{F}_{p^n})[\ell] \longrightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab}$$

Attacks

- inversion of e : hard problem (exponential)
- discrete logarithm computation in $E(\mathbb{F}_p)$: hard problem (exponential, in $O(\sqrt{\ell})$)
- discrete logarithm computation in $\mathbb{F}_{p^n}^*$: **easier, subexponential** $\to$ take a large enough field

Pairing-friendly curves

$\ell \mid p^n - 1$, $E[\ell] \subset E(\mathbb{F}_{p^n})$, n **embedding degree**
Tate Pairing: $e: \; E(\mathbb{F}_{p^n})[\ell] \times E(\mathbb{F}_{p^n})/\ell E(\mathbb{F}_{p^n}) \to \mathbb{F}_{p^n}^*/(\mathbb{F}_{p^n}^*)^\ell$
When n is small i.e. $1 \leqslant n \leqslant 24$, the curve is *pairing-friendly*.
This is very rare: For a given curve, $\log n \sim \log \ell$
([Balasubramanian Koblitz]).

p^n	p^2, p^6	p^3, p^4, p^6	p^{12}	p^{16}	p^{18}
Curve	supersingular	MNT	BN, BLS12	KSS16	KSS18

MNT, $n = 6$:
$p(x) = 4x^2 + 1, \; t(x) = 1 \pm 2x, \; \#E(\mathbb{F}_p)x^2 \mp 2x + 1$
BN, $n = 12$:
$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1, \; t(x) = 6x^2 + 1,$
$r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$
More in Aranha's talk.

security estimates

[Lenstra-Verheul'01] estimates RSA key-sizes
The usual security estimates use

- the asymptotic complexity of the best known algorithm (here NFS)
- the latest record computations (now 768-bit)
- extrapolation

Number Field Sieve Algorithm

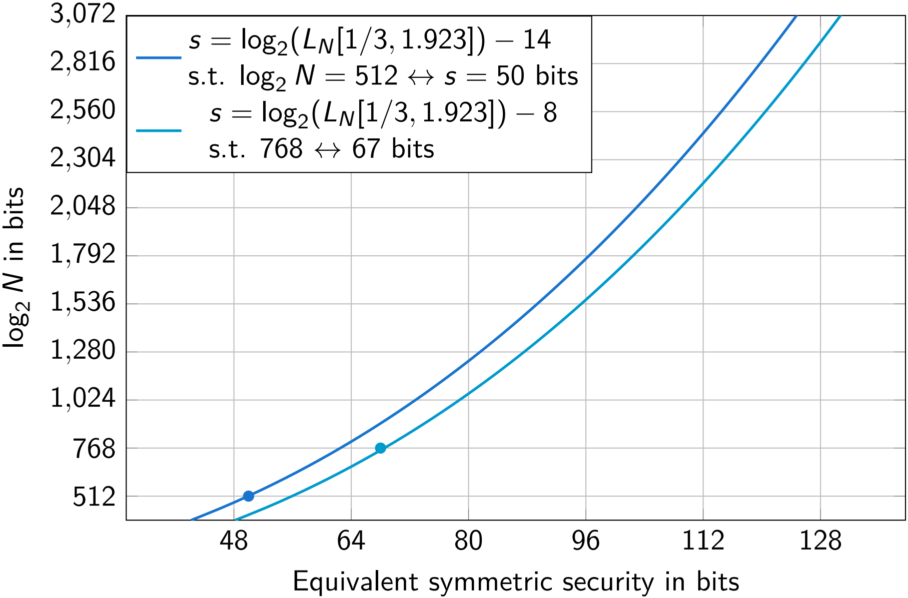
Subexponential asymptotic complexity:

$$L_{p^n}[\alpha, c] = e^{(c+o(1))(\log p^n)^\alpha (\log\log p^n)^{1-\alpha}}$$

- $\alpha = 1$: exponential
- $\alpha = 0$: polynomial
- $0 < \alpha < 1$: sub-exponential (including NFS)

1. polynomial selection (less than 10% of total time)
2. relation collection $L_{p^n}[1/3, c]$
3. linear algebra $L_{p^n}[1/3, c]$
4. individual discrete log computation $L_{p^n}[1/3, c' < c]$

Example for RSA key sizes



Pairing key-sizes in the 2000's

Assumed: DLP in prime fields $\mathbb{F}_p$ as hard as in medium and large characteristic fields $\mathbb{F}_Q$
$\rightarrow$ take the same size as for prime fields.

Security level	$\log_2 \ell$	finite field	n	$\log_2 p$	$\deg P$ $p = P(u)$	ρ	curve
128	256	3072		3072	(prime field)		
128	256	3072	2	1536	no poly	6	supersingular
	256	3072	3	1024	no poly	4	supersingular
	256	3072	12	256	4	1	Barreto-Naehrig
192	640	7680	12	640	4	$1 \rightarrow 5/3$	BN
	427	7680	12	640	6	$3/2$	BLS12
	384	9216	18	512	8	$4/3$	KSS18
	384	7680	16	480	10	$5/4$	KSS16
	384	11520	24	480	10	$5/4$	BLS24

Small, medium, large characteristic

$Q = p^n$, the characteristic p is

- small: $p = L_Q[\alpha, c]$ where $\alpha < 1/3$
- medium: $p = L_Q[\alpha, c]$ where $1/3 < \alpha < 2/3$
- large: $p = L_Q[\alpha, c]$ where $\alpha > 2/3$
- boundary cases: $p = L_Q[1/3, c]$ and $p = L_Q[2/3, c]$

Estimating key sizes for DL in GF(p^n)

GF(p^n) much less studied than GF(p) or integer factorization.

- 2000 LUC, XTR cryptosystems: multiplicative subgroup of prime order $\mid \Phi_n(p)$ (cyclotomic subgroup) of GF(p^2), GF(p^6)
- what is the hardness of computing DL in GF(p^n), $n = 2, 6$?
- 2005 [Granger Vercauteren] $L_Q[1/2]$
- 2006 Joux–Lercier–Smart–Vercauteren $L_Q[1/3, 2.423]$ (NFS-HD)
- rising of pairings: what is the security of DL in GF(2^n),GF(3^m),GF(p^{12})?

Asymptotic complexities

Needed:

- asymptotic complexity (constants α, c)
- record computations to scale the shape (guess the $o(1)$)

Asymptotic complexities now:

- For tiny characteristic: quasi-polynomial
- For small characteristic: $L(\alpha)$ for $\alpha < 1/3$
- For medium and large characteristic: $L(1/3, c + o(1))$

Asymptotic complexities

Needed:

- asymptotic complexity (constants α, c)
- record computations to scale the shape (guess the $o(1)$)

Asymptotic complexities now:

- For tiny characteristic: quasi-polynomial
- For small characteristic: $L(\alpha)$ for $\alpha < 1/3$
- For medium and large characteristic: $L(1/3, c + o(1))$

What is c for medium and large characteristic?

Theoretical improvements and records

	theoretical improvements	record computations
2013	Joux–Pierrot (SNFS for pairings)	
2014	MNFS, Conjugation	GF(p^2)
2015	TNFS	GF(p^2), GF(p^3), GF(p^4)
2016	Sarkar–Singh, exTNFS	GF(p^3)
2017	more exTNFS	NFS-HD: GF(p^5), GF(p^6)

Estimating key sizes for DL in GF(p^n)

- Latest variants of TNFS (Kim–Barbulescu, Kim–Jeong) seems most promising for GF(p^n) where n is composite
- We need record computations if we want to extrapolate from asymptotic complexities
- The asymptotic complexities do not correspond to a fixed n, but to a ratio between n and p in $Q = p^n$

Complexities

large characteristic $p = L_Q[\alpha],\ \alpha > 2/3$:	
$(64/9)^{1/3} \simeq 1.923$	NFS
special p:	
$(32/9)^{1/3} \simeq 1.526$	SNFS (e.g. Thomé's talk)

medium characteristic $p = L_Q[\alpha],\ 1/3 < \alpha < 2/3$:	
$(96/9)^{1/3} \simeq 1.201$	prime n NFS-HD (Conjugation)
$(48/9)^{1/3} \simeq 1.747$	composite n,
	best case of TNFS: when parameters fit perfectly
special p:	
$(64/9)^{1/3} \simeq 1.923$	NFS-HD+Joux–Pierrot'13
$(32/9)^{1/3} \simeq 1.526$	composite n, best case of STNFS

The NFS diagram for DLP in $\mathbb{F}_{p^n}^*$

Let f, g be two polynomials defining two number fields and such that in $\mathbb{F}_p[z]$, f and g have a common irreducible factor $\varphi(z) \in \mathbb{F}_p[z]$ of degree n, s.t. one can define the extension $\mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z))$
Diagram:

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

The NFS diagram for DLP in $\mathbb{F}_{p^n}^*$

Let f, g be two polynomials defining two number fields
and such that in $\mathbb{F}_p[z]$, f and g have a common irreducible factor
$\varphi(z) \in \mathbb{F}_p[z]$ of degree n, s.t. one can define the extension
$\mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z))$
Diagram: Large p:

$$
\begin{tikzcd}
 & a_0 - a_1 x \in \mathbb{Z}[x] \arrow[dl, "x \mapsto \alpha_f"'] \arrow[dr, "x \mapsto \alpha_g"] & \\
\begin{array}{c}(a_0 - a_1\alpha_f)\\ \text{smooth?}\end{array} \quad \mathbb{Z}[x]/(f(x)) \arrow[dr, "\alpha_f \mapsto z"'] & & \mathbb{Z}[x]/(g(x)) \arrow[dl, "\alpha_g \mapsto z"] \quad \begin{array}{c}(a_0 - a_1\alpha_g)\\ \text{smooth?}\end{array} \\
 & \mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z)) &
\end{tikzcd}
$$

The NFS diagram for DLP in $\mathbb{F}_{p^n}^*$

Let f, g be two polynomials defining two number fields and such that in $\mathbb{F}_p[z]$, f and g have a common irreducible factor $\varphi(z) \in \mathbb{F}_p[z]$ of degree n, s.t. one can define the extension $\mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z))$

Diagram: Large p:

$$\begin{tikzcd}
 & & a_0 - a_1 x \in \mathbb{Z}[x] \arrow[dl, "x \mapsto \alpha_f"'] \arrow[dr, "x \mapsto \alpha_g"] & & \\
\begin{array}{c}(a_0 - a_1\alpha_f) \\ \text{smooth?} \\ = \prod \mathfrak{q}_i^{e_i}\end{array} & \mathbb{Z}[x]/(f(x)) \arrow[dr, "\alpha_f \mapsto z"'] & & \mathbb{Z}[x]/(g(x)) \arrow[dl, "\alpha_g \mapsto z"] & \begin{array}{c}(a_0 - a_1\alpha_g) \\ \text{smooth?} \\ = \prod \mathfrak{r}_j^{e_i'}\end{array} \\
 & & \mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z)) & &
\end{tikzcd}$$

relation: "$\sum e_i \text{ vlog } \mathfrak{q}_i = \sum e_j' \text{ vlog } \mathfrak{r}_j$"

The NFS diagram for DLP in $\mathbb{F}_{p^n}^*$

Let f, g be two polynomials defining two number fields and such that in $\mathbb{F}_p[z]$, f and g have a common irreducible factor $\varphi(z) \in \mathbb{F}_p[z]$ of degree n, s.t. one can define the extension $\mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z))$

Diagram: Medium p: [Joux Lercier Smart Vercauteren 06]

$$
\begin{tikzcd}
 & & a_0 - a_1x + a_2x^2 \in \mathbb{Z}[x] \arrow[dl, "x \mapsto \alpha_f"'] \arrow[dr, "x \mapsto \alpha_g"] & & \\
\begin{array}{c}(a_0 + a_1\alpha_f + a_2\alpha_f^2)\\ \text{smooth?}\end{array} & \mathbb{Z}[x]/(f(x)) \arrow[dr, "\alpha_f \mapsto z"'] & & \mathbb{Z}[x]/(g(x)) \arrow[dl, "\alpha_g \mapsto z"] & \begin{array}{c}(a_0 + a_1\alpha_g + a_2\alpha_g^2)\\ \text{smooth?}\end{array} \\
 & & \mathbb{F}_{p^n} = \mathbb{F}_p[z]/(\varphi(z)) & &
\end{tikzcd}
$$

NFS parameters

- factor base $=$
  $\{$prime ideals $\mathfrak{p}_i, \ |\text{Norm}(\mathfrak{p}_i)| \leq B\}$
  $\cup\{$prime ideals $\mathfrak{r}_j, \ |\text{Norm}(\mathfrak{r}_i)| \leq B\}$
- we need as many relations as prime ideals $\mathfrak{p}_i, \mathfrak{r}_j$ to get a square matrix
- balance the relation collection time with the linear algebra time

Algebraic Norms

The asymptotic complexity is determined by the *size of norms* of the elements $\sum_{0\leq i<t} a_i\alpha^i$ in the relation collection step.
We want both sides *smooth* to get a relation.

"An ideal is B-smooth" approximated by
"its norm is B-smooth".

Smoothness bound: $B = L_{p^n}[1/3, \beta]$
Size of norms: $L_{p^n}[2/3, c_N]$
Complexity: minimize c_N in the formulas.
To reduce NFS complexity, reduce size of norms *asymptotically*.
$\rightarrow$ very hard task.

Extended TNFS [Kim Barbulescu 16]

- Tower NFS (TNFS): Barbulescu Gaudry Kleinjung
- Extended TNFS: Kim–Barbulescu, Kim–Jeong, Sarkar–Singh
- Tower of number fields
- $\deg(h)$ will play the role of t, where $a_0 + a_1\alpha + \ldots + a_{t-1}\alpha^{t-1}$
- $a_0 - a_1\alpha$ becomes $(a_{00} + a_{01}\tau) - (a_{10} + a_{11}\tau)\alpha$

$(a_{00} + a_{01}\tau) - (a_{10} + a_{11}\tau)\alpha_f$ smooth?

$(a_{00} + a_{01}\tau) - (a_{10} + a_{11}\tau)\alpha_g$ smooth?

$$\begin{array}{ccc} K_h[x]/(f(x)) & & K_h[x]/(g(x)) \\ & \diagdown \quad \diagup & \\ & K_h = \mathbb{Q}[\tau]/(h(\tau)) & \\ & \mid & \\ & \mathbb{Q} & \end{array}$$

Complexities

large characteristic $p = L_Q[\alpha],\ \alpha > 2/3$:	
$(64/9)^{1/3} \simeq 1.923$	NFS
special p:	
$(32/9)^{1/3} \simeq 1.526$	SNFS (e.g. Thomé's talk)

medium characteristic $p = L_Q[\alpha],\ 1/3 < \alpha < 2/3$:	
$(96/9)^{1/3} \simeq 1.201$	prime n NFS-HD (Conjugation)
$(48/9)^{1/3} \simeq 1.747$	composite n,
	best case of TNFS: when parameters fit perfectly
special p:	
$(64/9)^{1/3} \simeq 1.923$	NFS-HD+Joux–Pierrot'13
$(32/9)^{1/3} \simeq 1.526$	composite n, best case of STNFS

Largest record computations in GF(p^n) with NFS[1]

Finite field	Size of p^n	Cost: CPU days	Authors	sieving dim
GF(p^{12})	203	11	[HAKT13]	7
GF(p^6)	422	9,520	[GGMT17]	3
GF(p^5)	324	386	[GGM17]	3
GF(p^4)	392	510	[BGGM15b]	2
GF(p^3)	593	8,400	[GGM16]	2
GF(p^2)	595	175	[BGGM15a]	2
GF(p)	768	1,935,825	[KDLPS17]	2

None used TNFS, only NFS and NFS-HD were implemented.

[1] Data extracted from DiscreteLogDB

Limitations of asymptotic complexity

use: $\mathrm{Norm}_{K_f}(a(\alpha)) = \mathrm{Res}(a(x), f(x))$ (for monic f)

$$|\,\mathrm{Res}(a, f)| \leq (d_a + 1)^{d_f/2}(d_f + 1)^{d_a/2}\|a\|_\infty^{d_f}\|f\|_\infty^{d_a}$$

- based on bounds on coefficient size of polynomials, bounds on algebraic norms
- Kalkbrener, Bistritz–Lifshitz bounds are not satisfying enough
- no record computation available to re-scale the asymptotic formulas

Finding a better estimation and designing an implementation at the same time

Menezes–Sarkar–Singh Estimations

curve	$\log_2 p^n$	$\log_2 p$	variant	deg h	cost
BN	3072	256	TNFS with constants	4	2^{136}
BN	3732	311	TNFS without constants	4	2^{128}
BN	3072	256	STNFS with constants	6	2^{150}
BN	4596	383	STNFS without constants	6	2^{128}
BLS	4608	384	TNFS with constants	4	2^{156}
BLS	4608	384	TNFS without constants	4	2^{140}
BLS	4608	384	STNFS with constants	6	2^{189}
BLS	4608	384	STNFS without constants	6	2^{132}

Simulation

- compute record-looking polynomials
- simulate relation collection $\rightarrow$ extrapolate the number of relations
- estimate linear algebra
- neglect individual log

Questions:

- how to simulate well without being too slow?
- how to model the filtering step (packing the matrix)?
- by how much balancing relation collection and linear algebra?

Barbulescu-Duquesne simulation

Estimation of cost:

$$\frac{2B}{\mathcal{A}\log B}\rho\left(\frac{\log_2 N_f}{\log_2 B}\right)^{-1}\rho\left(\frac{\log_2 N_g}{\log_2 B}\right)^{-1}+2^7\frac{B^2}{\mathcal{A}(\log B)^2(\log_2 B)^2}$$

where $\mathcal{A} \leq n/\gcd(\deg h, n/\deg h)$,
ρ is the Dickman-ρ function

- takes into account Galois automorphisms
- takes into account filtering (reduced matrix)
- assume the coefficients of h, f are minimal
- assume $\alpha(f), \alpha(g) = 0$
- balance cost of sieving $\approx$ cost of linear algebra

Barbulescu-Duquesne estimates

curve	$\log_2 p^n$	$\log_2 p$	$\deg h$	cost
BN	3072	256	6	$2^{99,69}$
BN	5534	462	6	2^{128}
BLS	5530	461	6	2^{128}

Simulation without sieving

space: $\mathcal{S} = \{\sum_{0 \leq i < d_h} a_i y^i + (\sum_{0 \leq i < d_h} b_i y^i) x,\ |a_i|, |b_i| < A\}$
volume: $Vol = 2^{2d_h - 1} A^{2d_h}$
algebraic norm:
$N = \mathrm{Norm}_{K_f}(a(\alpha_h, \alpha_f)) = \mathrm{Res}_y(\mathrm{Res}_x(a(x,y), f(x)), h(y))$
(monic h, f)
N is B-smooth ($N = \prod_{p_i < B} p_i^{e_i}$) with probability

$$u = \frac{\log N + \alpha}{\log B},\ \Pr = \rho(u) + (1-\gamma)\frac{\rho(u-1)}{\log N}$$

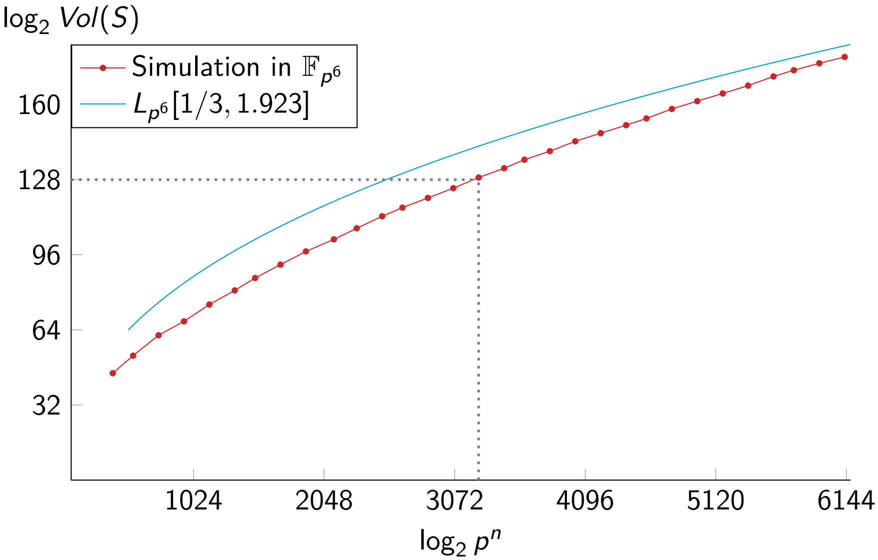
where $\gamma \approx 0.577$ is Euler γ constant,
ρ is Dickman-ρ function

Simulation without sieving

Implementation of Barbulescu–Duquesne technique
Variants:

- compute $\alpha(f), \alpha(g)$ (w.r.t. subfield)
- select h, f, g with good low $\alpha(f) < -3, \alpha(g) < -4$
- Monte-Carlo simulation with 10^6 to 10^9 points in $\mathcal{S}$ taken at random. For each point:
  1. compute its algebraic norm N_f, N_g in each number field
  2. smoothness probability with Dickman-ρ
- Average smoothness probability over the subset of points $\rightarrow$ estimation of the total number of possible relations in $\mathcal{S}$
- dichotomy to approach the best balanced parameters: smoothness bound B, coefficient bound A.

MNT curves, $\mathbf{G}_T \subset \mathbb{F}_{p^6}$



Observations

$(a) = (\sum_{i=0}^{d_h-1} a_i\tau)$, $(b) = (\sum_{i=0}^{d_h-1} b_i\tau)$ randomly chosen are coprime with probability $1/\zeta_{K_h}(2)$
Much different than for integers: $1/\zeta(2) = 6/\pi^2 \approx 0.6$

$$\zeta_{K_h}(s) = \sum_{n\in\mathbb{N}} \frac{1}{n^s}(\#\text{ideals of norm } n \text{ in } K_h)$$

$h = x^2 + 1$: $1/\zeta_{K_h}(2) \approx 0.6$
$h = x^2 - x + 4$: $1/\zeta_{K_h}(2) \approx 0.469$
$h = x^2 + x - 1$: $1/\zeta_{K_h}(2) \approx 0.861$
Experimentally: a good α comes with a low coprime probability

Future work

- How to rank polynomials according to their smoothness properties? α function (S. Singh) **faster**, generalized Murphy's E function
- How to build the factor basis?
- How to deal with generalized bad ideals?
- **How to sieve very efficiently in even dimension 4 to 24?**

Thank you for your attention.